

ULTIMATE
SOLID

2017 R1

Руководство
администратора

ver. 6.0.0

www.ultimatesolid.ru

Оглавление

Введение	4
Архитектура Ultimate Solid	4
Логика композиции систем на платформе Ultimate Solid	5
Руководство администратора	7
Аппаратный комплекс	7
Сервер базы данных	7
Установка базы данных в Oracle	7
Установка базы данных в PostgreSQL.....	7
Обновление платформы.....	8
Лицензирование.....	10
Сервер приложений	13
Таймаут транзакций.....	19
Сервер печати	19
Сервер экспорта	22
Клиентское приложение	24
Обновление клиентского приложения.....	27
Журналирование	28
Serilog.....	30
NLog	32
Описание процесса печати	33
Функционал модуля администратора	35
Группа инструментов Access control	36
Пользователи.....	37
Роли	38
Форма редактирования ролей	41
Предикаты.....	46
Права	47
Права доступа к системным таблицам БД	48
Группа инструментов конфигурации системы	50
Константы.....	50
Кластеры и сервера.....	52
Теги версий.....	55
Группа инструментов Monitor	57
Сессии.....	57
Просмотр логов.....	58
Группа инструментов Fast access	60
Группа инструментов Printing	61
Принтеры.....	61
Очередь печати.....	62
Группа инструментов History	64
Поиск по истории.....	64
Статистика вызова команд.....	66
Статистика печати.....	67

Группа инструментов Open by ID.....	67
Группа инструментов Manage.....	68
Настройка журналирования объектов метаданных.....	68
Пользовательские настройки.....	69
Пересчет итогов.....	70
Сброс кеша справочников.....	71
Группа инструментов Webservice.....	71
Пользователи web сервисов.....	72
Роли web сервисов.....	73
Сессии web сервисов.....	74
Настройка пользовательского интерфейса.....	74
Интерфейсы пользователей.....	79

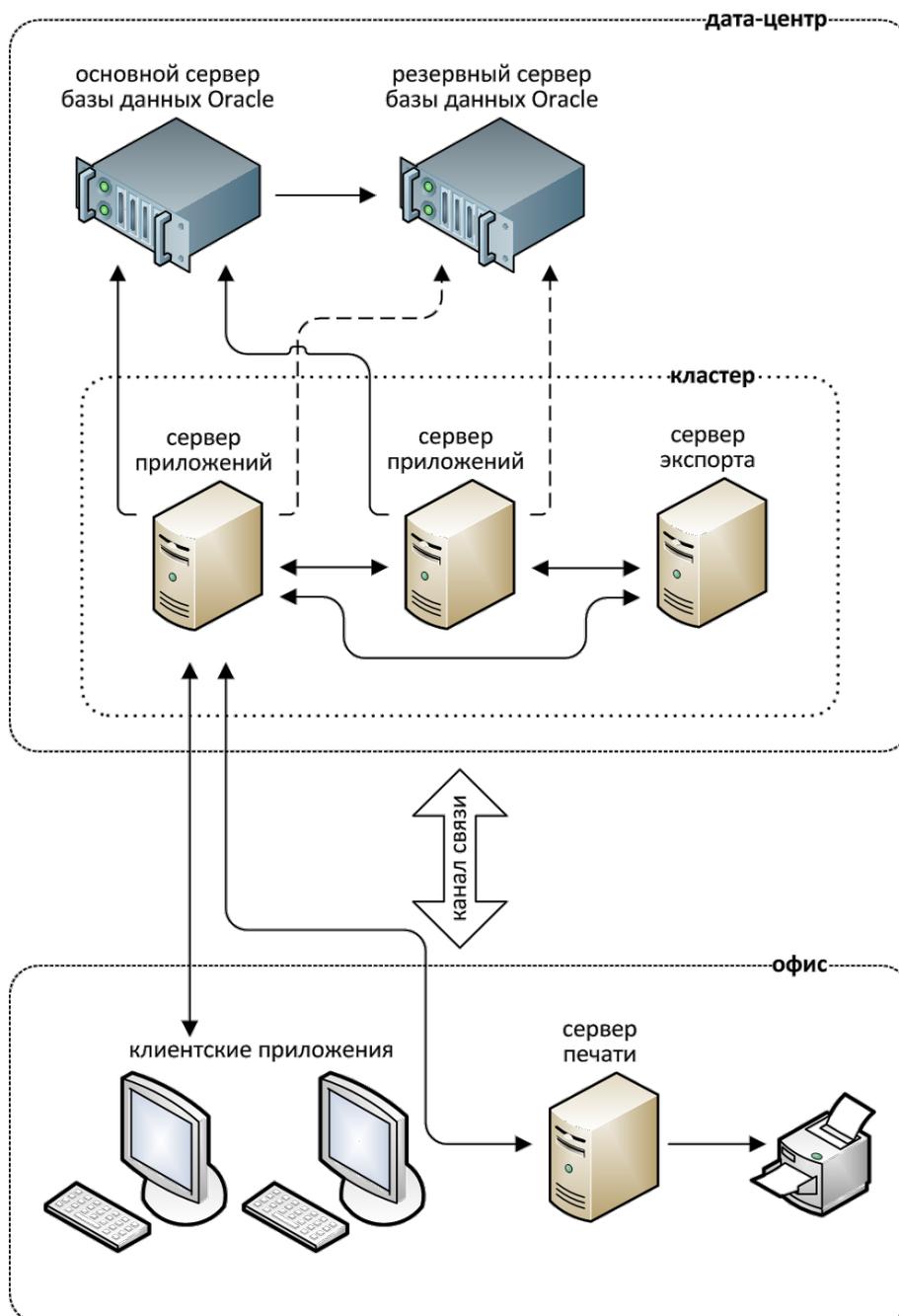
Введение

Архитектура Ultimate Solid

Платформа Ultimate Solid является программным комплексом трехуровневой архитектуры ([wikipedia](#)) и включает в себя:

- сервер базы данных (Oracle 11gR2 Enterprise Edition или Oracle 12c Enterprise Edition);
- сервер приложений;
- сервер печати;
- клиентские приложения.

Общая модель взаимодействия представлена на схеме:



Как правило, и это настоятельно рекомендуется, основной **сервер базы данных** располагается в дата-центре. Для повышения устойчивости к отказам оборудования, снижения риска потери

данных, а также для распределения нагрузки рекомендуется установить резервный (standby) сервер/кластер серверов.

Из аналогичных соображений отказоустойчивости и производительности **сервер приложений**, осуществляющий процессинг бизнес-логики и напрямую обменивающийся потоками данных с сервером БД, также может быть масштабирован в кластер. Вследствие чрезвычайно высокой интенсивности обмена данных в связке *сервер приложений - сервер БД*, они должны быть расположены в непосредственной близости друг от друга – как минимум, в рамках одной локальной сети.

В противоположность связке *сервер приложений - сервер БД*, **сервер печати** располагается в местах нахождения печатающих устройств и сотрудников, что позволяет существенно снизить нагрузку на каналы связи за счет передачи меньшего объема данных.

С помощью экранной логики **клиентских приложений** пользователь просматривает, вводит и редактирует данные.

Логика композиции систем на платформе Ultimate Solid

Эксплуатируемая заказчиком человекозамещающая управляющая система предприятия на платформе Ultimate Solid, состоит из двух логически различных сущностей.

Платформа Ultimate Solid – инвариантная часть системы (в дальнейшем термины *платформа* и *ядро* используются как взаимозаменяемые), изменяется только вендором (разработчиками Ultimate). Она (в частности) обеспечивает функционирование второй, изменяемой партнерами или независимыми разработчиками, части, называемой пространство бизнес-логики.

Пространство бизнес-логики содержит сценарии обработки бизнес-правил, сами бизнес правила, экранные формы и так далее и тому подобное. Конкретное наполнение пространства бизнес-логики, например, соответствующее конкретному клиенту или чуть более обще конкретному варианту бизнеса называется конфигурацией пространства бизнес-логики. Или коротко просто конфигурацией.

Развернутое ПО в виде платформы и заданной конфигурации у конкретного клиента называется инсталляцией. Готовый комплект платформы и конфигурации называется решением и обычно имеет тоже имя что и конфигурация. Например, для конфигурации e-Trade решение будет так же называться Ultimate e-Trade. В дальнейшем мы так же как синонимы будем использовать термины система, продукт и решение.

Платформа обеспечивает функционирование всего программного комплекса в целом и предоставляет следующие инструменты и механизмы:

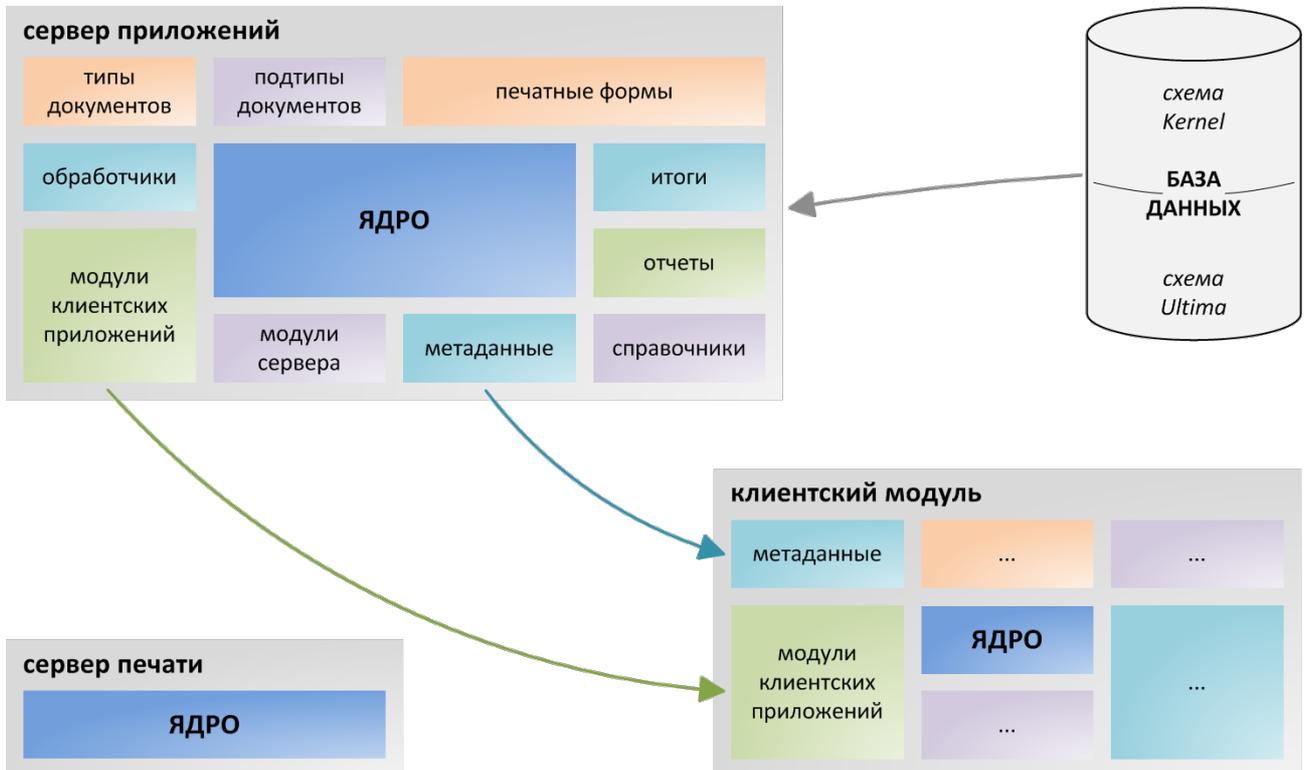
- хранения пользователей, их авторизации и проверки прав;
- коммуникации между приложениями;
- доступа к СУБД (сохранения объектов системы в базу данных);
- управления изменениями конфигураций (механизм версионирования);
- преобразования "сырых данных" из БД в объекты системы;
- интеграции с другими системами – SOAP, JSON, REST, XML и т.д.;
- среда разработки и изменения конфигурации.

Пространство бизнес-логики или конфигурация – это набор метаданных, описывающих бизнес-объекты и непосредственных данных:

- документы;
- справочники;

- связи между справочниками и документами;
- итоги;
- обработчики, описывающие логику взаимодействия бизнес-объектов;
- серверные модули;
- клиентские модули;
- права пользователей.

Общая модель представлена на схеме:



Конфигурация целиком хранится в базе данных. В качестве СУБД используется Oracle 12c Enterprise Edition или PostgreSQL 9.6. В базе данных имеются две схемы:

- *Kernel* – статичная, именно из этой схемы сервер приложений при запуске загружает конфигурацию;
- *Ultima* – прикладная, в ней создаются таблицы и хранятся данные прикладной области.

В базе PostgreSQL используется еще несколько других схем, также относящихся к ядру. Версия системы на базе PostgreSQL имеет некоторые ограничения по сравнению с версией на базе Oracle, но в остальном функционально эквивалентна. Ограничения версии PostgreSQL и особенности разработки для нее описаны в соответствующих разделах.

Ядро недоступно для изменения прикладным разработчиком, однако предоставляет интерфейс и инструменты для реализации бизнес-логики. Сама бизнес-логика, исполняемая на сервере приложений, реализуется в виде классов на C#, унаследованных от соответствующих классов платформы (в дальнейшем такие классы будут называться скриптами). Экранная логика выполняется на клиентском приложении и реализуется в виде модулей на любом .NET совместимом языке.

Руководство администратора

Аппаратный комплекс

Сервер базы данных

В Ultimate Solid используется Oracle 12c Enterprise Edition или PostgreSQL 9.6. Администратор должен обладать необходимыми навыками администрирования соответствующей СУБД. Для снижения риска потери данных и сокращения возможного времени простоя в случае аварии на основном сервере базы данных рекомендуется устанавливать хотя бы один резервный (StandBy) сервер. Если резервные сервера открыты для чтения, их можно использовать для снижения нагрузки на основном сервере. База данных хранит всю первичную информацию. Любые файлы на сервере приложений или клиентских устройствах либо относятся к платформе, либо являются конфигурационными файлами приложений платформы, либо являются кешем для данных в БД. С сервером базы данных общается только сервер приложений, остальные части системы не требуют и не должны иметь доступа к БД.

Установка базы данных в Oracle

База данных поставляется в виде готового образа pluggable database для Oracle Multitenant Architecture.

Для подключения базы сервер Oracle должен быть установлен с поддержкой Pluggable Databases и кодировкой UTF-8 (используется по умолчанию в Oracle 12cR2).

Выполните следующие шаги:

1. Разархивируйте дистрибутив с базой данных в папку где установлен Oracle. Для Windows обычно `c:\app\oracle`
2. Запустите скрипт с правами на запись в папку (правами администратора для Windows) инсталляции для соответствующей ОС (.bat или .sh)
3. Все действия будут выполняться автоматически.

Установка базы данных в PostgreSQL

База данных для PostgreSQL поставляется в виде полного архива базы данных. Установка осуществляется путем простого копирования файлов в папку баз данных PostgreSQL. Данная инструкция предполагает, что администратор обладает навыками администрирования ОС Linux.

Для установки выполните следующие действия:

1. Установите PostgreSQL 9.6 (за инструкциями обратитесь к оригинальной документации к этой СУБД)
2. Остановите сервер PostgreSQL
`/etc/init.d/postgresql-9.6 stop`
3. Разархивируйте архив с бд в `~postgres/`
`(cd ~postgres ; tar xzf /tmp/pgdb.tgz ; chown -R postgres.postgres ~postgres)`
4. Убедитесь, что PostgreSQL доступен снаружи сервера
`/sbin/iptables -I INPUT -p tcp --dport 1522 -j ACCEPT`

5. Запустите PostgreSQL
/etc/init.d/postgresql-9.6 start

Для различных вариантов дистрибутивов OS Linux пункты 2, 4 и 5 могут незначительно отличаться. Обратитесь к оригинальной документации к СУБД для соответствующего дистрибутива.

Обновление платформы

Процесс обновления платформы предполагает последовательное выполнение следующих шагов:

1. создание рабочей копии базы данных;
2. проведение на созданной копии процедуры обновления и всех дополнительных мероприятий, которые необходимы для преодоления изменений, нарушающих обратную совместимость (подробно описаны [далее](#) в этом же разделе);
3. проведение тестирования основных процессов компании в обновленной копии базы;
4. только в случае отсутствия проблем на предыдущих этапах допустим переход к обновлению основной базы данных;
5. остановка работы в базе (выключение всех серверов приложений, для надежности можно сменить пароли пользователей чтобы гарантировать монопольный доступ);
6. создание резервной копии базы данных (backup);
7. выполнение процедуры обновления и всех сопутствующих мероприятий, проведение тестирования критических функций;
8. при возникновении проблем следует восстановить базу данных из резервной копии, откатив изменения, и предоставить пользователям возможность работать дальше на старой версии базы. С возникшими проблемами следует разбираться на копии базы данных(!), повторив процедуру с пункта 1.

Процедура обновления начинается с обновления базы данных. Оно производится с помощью утилиты-консольного приложения **VersionUpgrade.exe**, которая входит в дистрибутив Ultimate Solid.

Настройки утилиты хранятся в XML-файле **VersionUpgrade.exe.config** (при первом запуске утилиты можно скопировать или переименовать файл *VersionUpgrade.exe.config.sample*). Настройки, которые может менять администратор, находятся в секции *userSettings*:

```
<userSettings>
  <Ultima.VersionUpgrade.Settings>
    <setting name="AppServerID" serializeAs="String">
      <value>1</value>
    </setting>
    <setting name="UserID" serializeAs="String">
      <value>1</value>
    </setting>
    <setting name="ConnectionString" serializeAs="String">
      <value>Data Source=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
        (HOST=192.168.0.24)(PORT=1521))
        (CONNECT_DATA=(SERVICE_NAME=UNEXT)));
        User ID=kernel;Password=*****;Enlist=True;
        Promotable Transaction=Local;</value>
    </setting>
    <setting name="SysConnectionString" serializeAs="String">
      <value>Data Source=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
        (HOST=192.168.0.24)(PORT=1521))
        (CONNECT_DATA=(SERVICE_NAME=UNEXT)));
        User ID=sys;Password=*****;Enlist=True;
        Promotable Transaction=Local;DBA PRIVILEGE=sysdba;</value>
    </setting>
  </Ultima.VersionUpgrade.Settings>
</userSettings>
```

```
</setting>
</Ultima.Server.Properties.Settings>
</userSettings>
```

AppServerID – идентификатор сервера приложений.

UserID – идентификатор пользователя, от имени которого производится изменение данных.



У пользователя *UserID* должны быть права разработчика (роль обладает правом *Developer*).

ConnectionString – строка, содержащая настройки доступа к серверу базы данных для пользователя *kernel* (приведен пример для СУБД Oracle).

SysConnectionString – строка соединения для пользователя с правами администратора, который может назначать пользователю *kernel* новые права (приведен пример для СУБД Oracle, укажите доступ для пользователя *postgres* для PostgreSQL).

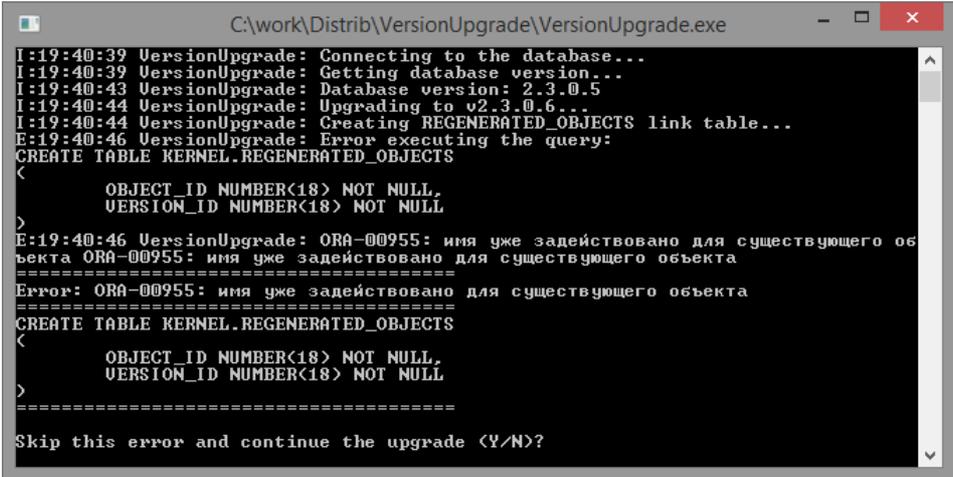


Для обновления базы данных необходимо:

- отредактировать конфигурационный файл *VersionUpgrade.exe.config*:
 - задать в секциях *ConnectionString* и *SysConnectionString* значения параметров *HOST*, *SERVICE_NAME* и *PASSWORD*;
 - убедиться, что указанный в секции *AppServerID* сервер приложений работает на ветке конфигурации, помеченной тегом *Default*. Запускать процедуру обновления базы на любом другом теге недопустимо (!);
- запустить утилиту *VersionUpgrade.exe*.

Обновление версии базы данных производится инкрементально. То есть, если производится обновлению с версии 2.3.0.6 до 2.3.0.9, процесс последовательно пройдет все стадии обновления: сначала до 2.3.0.7, затем до 2.3.0.8 и, наконец, до 2.3.0.9.

Журналирование процесса обновления настроено на вывод в консоль, в которой запущена утилита. Если база данных была клонирована из промежуточного релиза, при обновлении возможны некритические ошибки (например, попытка создать уже существующую таблицу). По каждой такой ошибке утилита обновления предоставляется пользователю возможность выбора – прервать или продолжить процесс обновления. Если по тексту запроса очевидно, что ошибку можно пропустить, следует продолжить процесс. При любых сомнениях следует проконсультироваться с системным отделом:



```
C:\work\Distriб\VersionUpgrade\VersionUpgrade.exe
I:19:40:39 VersionUpgrade: Connecting to the database...
I:19:40:39 VersionUpgrade: Getting database version...
I:19:40:43 VersionUpgrade: Database version: 2.3.0.5
I:19:40:44 VersionUpgrade: Upgrading to v2.3.0.6...
I:19:40:44 VersionUpgrade: Creating REGENERATED_OBJECTS link table...
E:19:40:46 VersionUpgrade: Error executing the query:
CREATE TABLE KERNEL.REGENERATED_OBJECTS
<
  OBJECT_ID NUMBER(18) NOT NULL,
  VERSION_ID NUMBER(18) NOT NULL
>
E:19:40:46 VersionUpgrade: ORA-00955: имя уже задействовано для существующего об
екта ORA-00955: имя уже задействовано для существующего объекта
=====
Error: ORA-00955: имя уже задействовано для существующего объекта
=====
CREATE TABLE KERNEL.REGENERATED_OBJECTS
<
  OBJECT_ID NUMBER(18) NOT NULL,
  VERSION_ID NUMBER(18) NOT NULL
>
=====
Skip this error and continue the upgrade (Y/N)?
```

После обновления базы данных можно приступить к обновлению платформы, для чего следует:

- сделать резервную копию текущих бинарных файлов платформы;

2. скопировать бинарные файлы поверх файлов платформы на серверах приложений. При необходимости, например, если изменилась первая (major) версия платформы, необходимо перекомпилировать клиентские модули и метаданные;
3. запустить новый сервер приложений и клиентское приложение. При запуске сервер или клиент могут сообщить о невозможности загрузить обработчики или клиентские модули, следует игнорировать эти сообщения;
4. на каждой используемой ветке выполнить Developer -> Compile -> Regenerate metadata classes. На обычных не branch-тегах это делать не следует (!);
5. на каждой используемой ветке выполнить Developer -> Compile -> Compile & Reload metadata;
6. перекомпилировать клиентские модули с новой версией метаданных;
7. перезапустить сервер и клиентское приложение;
8. обновление завершено.

По завершении обновления следует преодолеть изменения, нарушающие обратную совместимость (breaking changes). Подробно об этом написано в соответствующем разделе документации разработчика.

Лицензирование

Для работы с сервером приложений Ultimate Solid необходимо иметь лицензию. Лицензия представляет собой текстовый файл с именем и ключом компании-владельца и списком параметров лицензирования, заверенный цифровой подписью партнерского центра. Параметры лицензирования устанавливают набор ограничений на эксплуатационные характеристики системы:

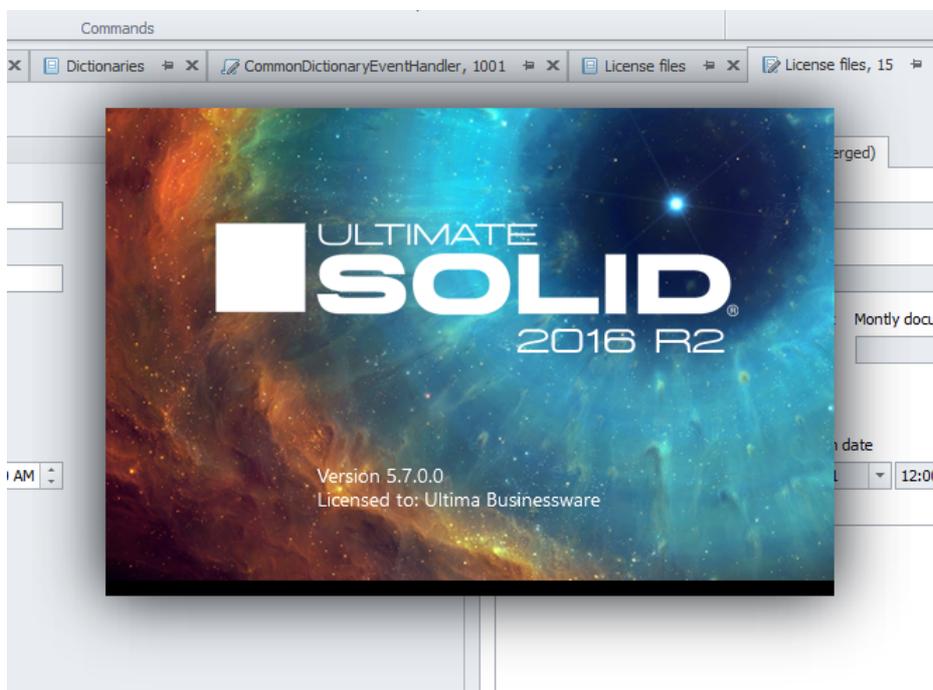
- Количество одновременных сеансов пользователей;
- Количество одновременных сеансов разработчиков;
- Лимит на количество создаваемых документов (в пересчете на одного пользователя за сутки);
- Список учетных записей веб-сервиса, которым разрешается иметь неограниченное число сеансов;
- Дата выдачи лицензии;
- Дата начала срока лицензирования (обычно совпадает с датой выдачи лицензии);
- Дата окончания срока лицензирования (только для временных лицензий).

После приобретения лицензии ее необходимо загрузить в справочник лицензий. Для этого достаточно подложить ее в рабочий каталог сервера приложений, чтобы при запуске сервера автоматически произошла загрузка. После загрузки в справочник лицензия будет доступна всем серверам приложений, работающих с этой же базой данных.

Компании, получившей лицензию на использование платформы, присваивается уникальный ключ-идентификатор, которым будут отмечены все файлы лицензий, выданные в дальнейшем этой компании. Ключ предприятия определяется лицензией, помеченной в справочнике как главная. В каждый момент времени может быть активно несколько файлов лицензий. Количество пользователей и разработчиков, указанных во всех действующих лицензиях, при этом суммируется, списки безлимитных веб-пользователей объединяются, лимит количества документов на пользователя — выбирается максимальный. Повторная загрузка одних и тех же файлов лицензий в справочник не приводит к увеличению числа разрешенных пользователей, так как дубликаты не учитываются.

Активными считаются все лицензии, ключ предприятия которых совпадает с ключом предприятия главной лицензии и период действия которых еще не истек. Название компании-лицензиата задается последней выданной лицензией, поэтому при смене названия достаточно получить одну лицензию с новым именем компании (ключ предприятия при этом остается прежним). Если

конфигурация требует увеличения характеристик за пределы заданных в лицензионном файле ограничений (например, при увеличении числа одновременно работающих пользователей), необходимо приобрести и загрузить новую лицензию. Управлять загруженными лицензиями можно в справочнике файлов лицензий. Название компании-владельца лицензии отображается в окне «О программе», в самом низу:



Контроль лимитов документов и сеансов подключения

В ходе работы система автоматически измеряет все ограничиваемые активной лицензией параметры: количество одновременных сеансов пользователей и созданных документов.

Сеансы пользователей бывают двух видов: нативные подключения и веб-подключения. Нативные подключения учитывают отдельно обычных пользователей и пользователей-разработчиков и всегда учитываются по количеству. Веб-подключения позволяют два варианта учета: посессионный (количественный) и неограниченный. По умолчанию используется посессионная схема. Неограниченные веб-подключения разрешаются только для тех пользователей, учетные записи которых явно перечислены в активной лицензии. Неограниченные веб-подключения обычно используются для интернет-магазина, для модуля интеграции с Bitrix и т. п.



Для включения неограниченного режима учета требуется, чтобы логин и пароль веб-пользователя полностью (с учетом регистра) совпадали с теми, которые были указаны при оформлении файла лицензии. При авторизации веб-пользователя пароль безлимитной учетной записи проверяется по файлу лицензий, а не по справочнику веб-пользователей.

Приложения, использующие неограниченную схему, должны скрывать логин и пароль доступа (например, шифрованием файлов конфигурации) и гарантировать невозможность его изменения.

Приложения, использующие посессионную схему, должны обращаться к серверу не реже, чем раз в 2 минуты. Подключение считается активным, если последний доступ через него был не более 2.5 минут назад.

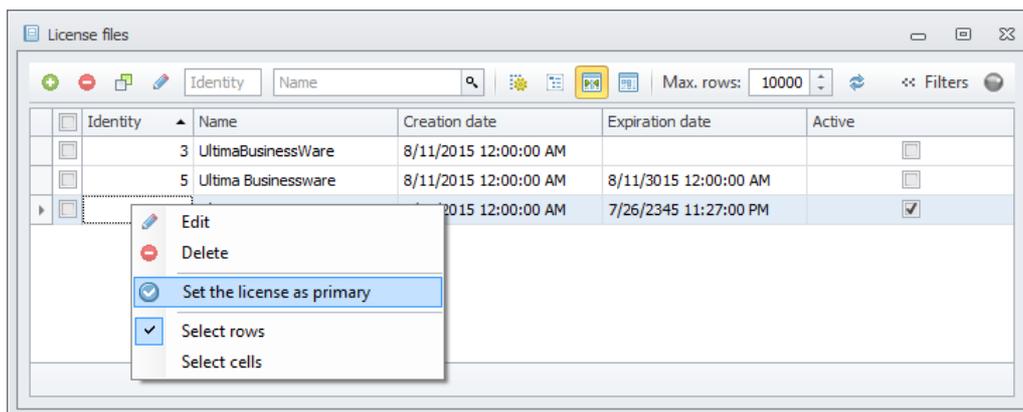
Контроль количества одновременных сеансов происходит при входе пользователя в систему или при восстановлении неактивного сеанса (например, при возвращении сотрудника к работе после обеденного перерыва). Если лимит превышен, вход в систему будет ограничен до тех пор, пока не освободится какой-нибудь сеанс.

Сеансы разработчиков учитываются отдельно от обычных пользователей (разработчиком считается любой пользователь, обладающий разрешением Developer). Если количество одновременно работающих разработчиков превысило лимит, установленный лицензией, разработчик может воспользоваться незанятым сеансом обычного пользователя. Обратное неверно.

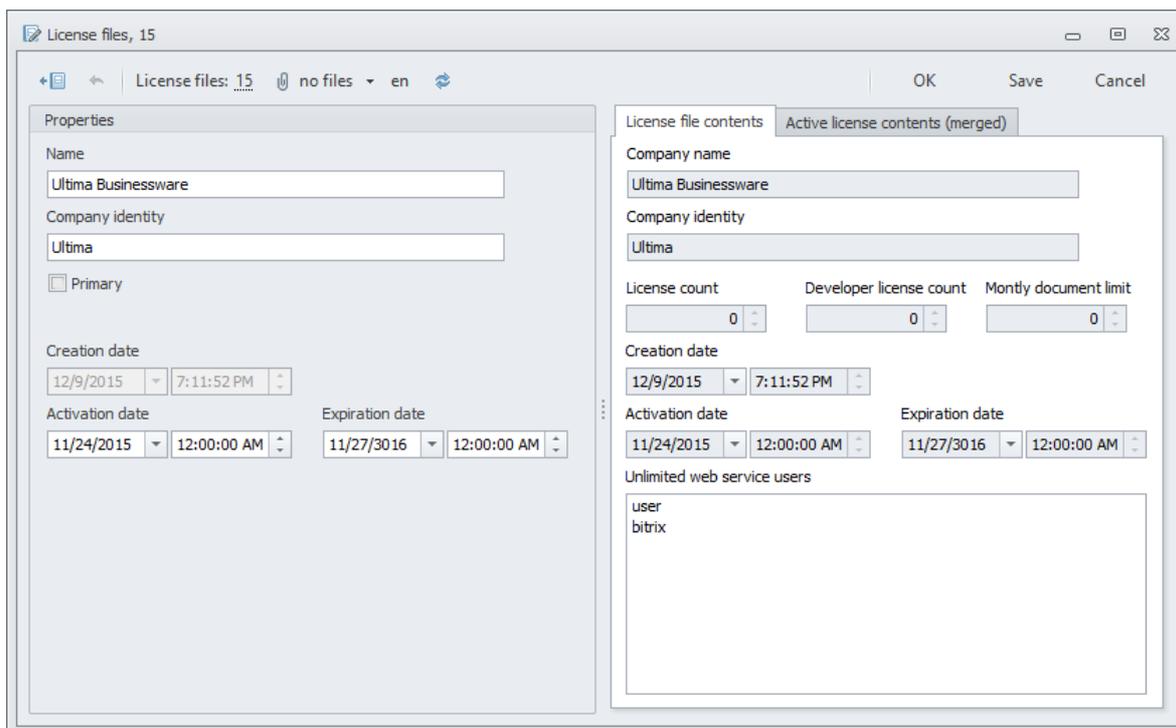
Учет количества создаваемых документов работает следующим образом. Учитываются последние 90 дней работы системы, разделенные на три периода по 30 дней. Подсчитывается количество созданных документов в каждом периоде. Если лимит на количество документов был превышен в первых двух периодах, система выдает предупреждение. Если лимит превышен во всех трех периодах, система блокирует создание новых документов. Пересчет количества созданных документов происходит при старте сервера и далее раз в сутки.

Справочник файлов лицензий

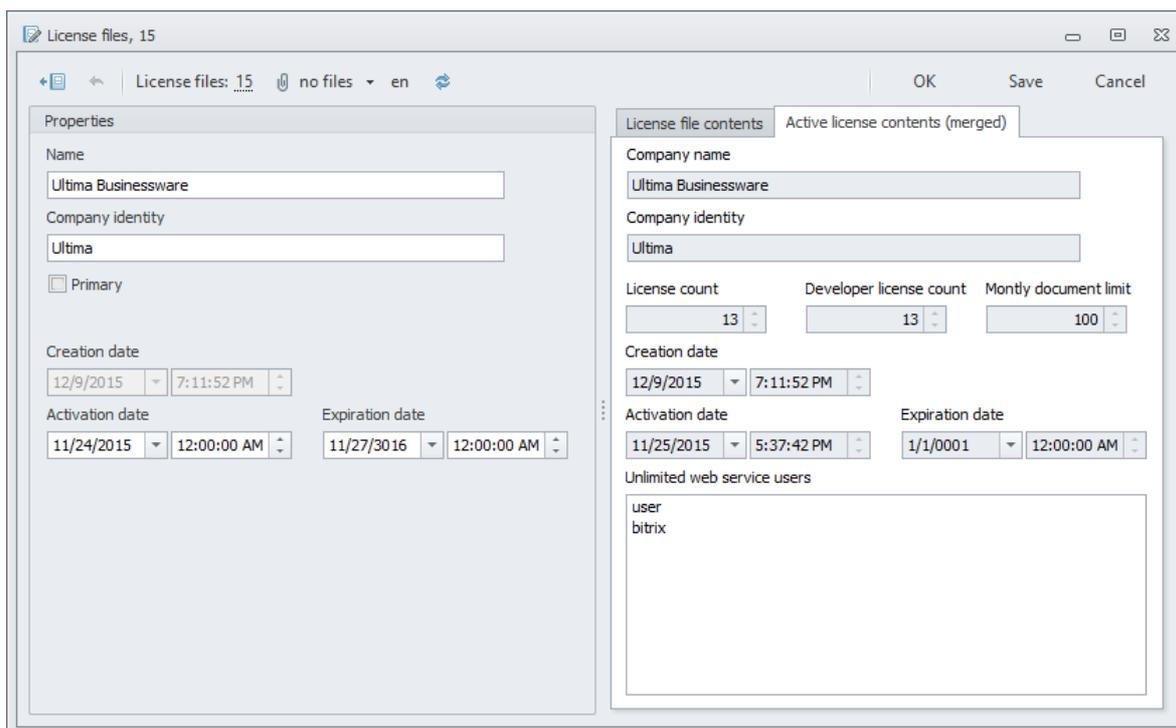
Посмотреть загруженные лицензии, загрузить новые, а также выбрать главную (primary) лицензию можно в справочнике лицензий:



При пометке главной лицензии с помощью команды контекстного меню прежняя главная лицензии лишается своего статуса. Справочник лицензий позволяет просматривать содержимое лицензий, дату выпуска, параметры и списки неограниченных веб-пользователей, но не позволяет менять параметры лицензирования:



На закладке **Active license contents (merged)** можно посмотреть текущие ограничения объединенной лицензии, включающей все активные на данный момент лицензии предприятия:



Сервер приложений

 Для распределения нагрузки и снижения времени простоя при поломках оборудования сервер приложений может быть установлен на нескольких серверах.

Сервер приложений можно запустить как **консольное приложение** –  **ConsoleServer.exe** (используется преимущественно разработчиками), или установить как **службу**, запустив тот же файл с ключом `install` (с правами администратора) – `UltimaService.exe install`.



Запускать сервер как консольное приложение следует с правами администратора.

Поддерживаются следующие опции командной строки – `ConsoleServer.exe [ключ] [-опция:значение] [-переключатель]`:

- **run** – запустить как консольное приложение (используется по умолчанию при запуске без ключа);
- **help** – показать помощь;
- **install** – установить службу:
 - `-username` – имя пользователя для запуска сервиса;
 - `-password` – пароль указанного пользователя;
 - `-instance` – имя экземпляра службы, если она устанавливается несколько раз;
 - `--autostart` – служба должна запускаться автоматически (опция по умолчанию);
 - `--disabled` – служба должна быть установлена со статусом запрещена;
 - `--manual` – служба должна запускаться вручную;
 - `--delayed` – служба должна запускаться автоматически (с задержкой);
 - `--localservice` – запускать службу с учетной записи локальной службы;
 - `--networkservice` – запускать службу с разрешениями сетевой службы;
 - `--interactive` – служба обратится к пользователю во время установки за полномочиями службы;
 - `--sudo` – запросы к UAC, если выполняется на W7/2008;
 - `-servicename` – имя службы, которое должно использоваться при установке. По умолчанию служба устанавливается с именем **UltimaService**;
 - `-description` – описание службы, которое должно использоваться при установке;
 - `-displayname` – отображаемое имя службы, которое должно использоваться при установке;
- **start** – запустить службу, если она еще не запущена:
 - `-instance` – экземпляр службы для запуска;
- **stop** – остановить службу, если она запущена:
 - `-instance` – экземпляр службы для остановки;
- **uninstall** – удалить службу:
 - `-servicename` – имя службы;
 - `-instance` – имя экземпляра службы, если она устанавливалась несколько раз;
 - `--sudo` – запросы к UAC, если выполняется на Vista/W7/2008;
- параметры, влияющие на запуск сервера приложений в консольном режиме:
 - `-AppServerID:N` – код сервера приложений N (целочисленный параметр, нет значения по умолчанию);
 - `-DataProviderName:Name` – использовать указанный провайдер. Поддерживаются следующие значения:
 - *OracleManaged* – ODP.NET Managed driver (используется по умолчанию);
 - *OracleUnmanaged* – ODP.NET Unmanaged driver, требует отдельной установки Oracle client software (OCI);
 - *Postgres* – Провайдер Npgsql для PostgreSQL.
 - `-DuplexProtocol:true/false` – использовать дуплексный или симплексный протокол (по умолчанию true);
 - `-SecureProtocol:true/false` – использовать протокол с шифрованием данных (по умолчанию false);
 - `-SkipLoadingMetadata:true/false` – пропустить загрузку метаданных, если они повреждены (по умолчанию false);

- -TaskSchedulerActive:true/false — включить или выключить планировщик задач (по умолчанию true);
- -TcpPort:N — номер TCP-порта N (целочисленный параметр, нет значения по умолчанию).



Если при установке служба была переименована (servicename) или было задано имя ее экземпляра (instance), эти параметры придется указывать для всех действий со службой – запуска, остановки или удаления.

Практически все настройки сервера приложений (и кластера) хранятся на сервере базы данных. Конфигурация кластера в минимальном объеме включает в себя перечень серверов. На самом сервере приложений настраиваются только опции запуска серверного модуля. Они хранятся в XML-файле (стандартном для любого .NET приложения) **ConsoleServer.exe.config**.

Настройки, которые может менять администратор, находятся в секции *userSettings*:

```
<userSettings>
  <Ultima.Server.Properties.Settings>
    <setting name="ServiceHostName" serializeAs="String">
      <value>UltimaServer</value>
    </setting>
    <setting name="TcpPort" serializeAs="String">
      <value>8192</value>
    </setting>
    <setting name="DataProviderName" serializeAs="String">
      <value>OracleManaged</value>
    </setting>
    <setting name="ConnectionString" serializeAs="String">
      <value>Data Source=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
        (HOST=192.168.0.24)(PORT=1521))
        (CONNECT_DATA=(SERVICE_NAME=UNEXT)));
        User ID=ultima;Password=*****;Enlist=True;
        Promotable Transaction=Local;</value>
    </setting>
    <setting name="DuplexProtocol" serializeAs="String">
      <value>True</value>
    </setting>
    <setting name="SecureProtocol" serializeAs="String">
      <value>False</value>
    </setting>
    <setting name="AssembliesFolderName" serializeAs="String">
      <value>ServerAssemblies</value>
    </setting>
    <setting name="AppServerID" serializeAs="String">
      <value>1</value>
    </setting>
    <setting name="ScriptCacheFolderName" serializeAs="String">
      <value>Scripts</value>
    </setting>
    <setting name="LicenseFileName" serializeAs="String">
      <value>UltimaLicense.xml</value>
    </setting>
    <setting name="TaskStatSyncInterval" serializeAs="String">
      <value>1000</value>
    </setting>
    <setting name="TaskExecutorLogin" serializeAs="String">
      <value>TaskExecutor</value>
    </setting>
    <setting name="TaskExecutorPassword" serializeAs="String">
      <value>UltimaDerPassword</value>
    </setting>
  </Ultima.Server.Properties.Settings>
</userSettings>
```

```

    </setting>
    <setting name="TaskSchedulerActive" serializeAs="String">
      <value>True</value>
    </setting>
    <setting name="EnableRestartService" serializeAs="String">
      <value>True</value>
    </setting>
  </Ultima.Server.Properties.Settings>
</userSettings>

```

ServiceHostName – имя сервера приложений, это имя будет отображаться в списке сервисов.

TcpPort – порт, по которому другие сервера и клиентские приложения будут получать доступ к данному серверу приложений.

DataProviderName – имя провайдера данных. Доступны следующие варианты:

- *OracleManaged* – управляемая версия ODP.NET (Oracle Data Provider Managed Driver), предпочтительный способ для работы с базой данных Oracle. Не требует администрирования;
- *OracleUnmanaged* – неуправляемая версия ODP.NET, альтернативный способ соединения с базой Oracle. Требуется установки дополнительного программного обеспечения Oracle Client;
- *Postgres* – управляемый провайдер для базы данных PostgreSQL.

ConnectionString – строка, содержащая настройки доступа к серверу базы данных.

При использовании управляемого драйвера ODP.NET (*UltimaDataProviderOracleManaged*) доступны следующие опции:

- *Data Source* или *Server* или *Host* – имя сервера базы данных Oracle к которой осуществляется подключение;
- *User ID* или *User* – учетная запись Oracle;
- *Password* – пароль для учетной записи Oracle;
- *Service Name* – имя экземпляра базы данных Oracle;
- *SID* – уникальное имя экземпляра базы данных Oracle;
- *Statement Cache Size* – включает или выключает кэширование операторов. Значение определяет максимальное количество операторов, которые могут кэшироваться для подключения. По умолчанию этот атрибут имеет значение 0 (отключено). Кэширование операторов начинается, если этот параметр установлен в значение больше 0. Он не должна быть больше, чем MAX_OPEN_CURSORS параметр в базе данных Oracle;
- *Connection Timeout* – время (в секундах) ожидания при попытке установить соединение перед завершением попытки и генерацией ошибки. Значение 0 означает отсутствие ограничения. Значение по умолчанию 15 (секунд);
- *Pooling* – в значение *true* используется пулинг соединений. Пулинг соединений позволяет сократить время обращения к БД. Значение по умолчанию *true*;
- *Min Pool Size* – минимальное количество подключений в пуле. Значение по умолчанию 0;
- *Max Pool Size* – максимальное количество подключений в пуле, может повлиять на производительность. Значение по умолчанию 100;
- *Connection Lifetime* – время жизни соединения. При соединении возвращается в пул, время его создания сравнивается с текущим временем, и соединение будет закрыто, если этот промежуток времени (в секундах) превышает значение, указанное в *Connection Lifetime*. Значение по умолчанию 0;
- *Default Command Timeout* – время в секундах при попытке выполнить команду перед завершением попытки и генерацией ошибки. 0 означает отсутствие ограничения;
- *Validate Connection* – определяет, проверять ли соединения, которые приходят из пула;
- *Enlist* – включает (в значении *true*) или выключает автоматическое прикрепление к распределенной транзакции в существующих транзакциях;

- *Promotable Transaction* – указывает, локальная ли транзакция (в значении *local*) или распределенная (в значении *promotable*) в течение ее жизни.

При использовании неуправляемой версии ODP.NET (*UltimaDataProviderOracleUnmanaged*) можно также задать следующие параметры:

- *Home* – значение Oracle Home, который будет использоваться;
- *Oci Session Pooling* – в значении *true* делает доступными функции пула сессий OCI (OCI Session Pooling);
- *Oci Session Pool Allow Waiting* – в значении *true* новые соединения будут ожидать закрытия существующих, если значение, установленное параметром *Oci Session Pool Max Size*, будет достигнуто, в значении *false* – отклонены;
- *Oci Session Pool Connection Lifetime* – определяет время жизни соединения в секундах. Перед возвращением подключения обратно в пул время его жизни проверяется. Если время жизни соединения превышает значение этого свойства, соединение закрывается и удаляется. Если этот параметр имеет значение 0, время жизни соединения никогда не проверяется;
- *Oci Session Pool Increment* – позволяет приложениям устанавливать следующее приращение для сессий, которые будут запущены, если текущее число сессий меньше чем *Oci Session Pool Max Size*. Допустимые значения 1 и выше;
- *Oci Session Pool Max Size* – определяет максимальное число сессий, которые могут быть открыты в пуле сессий. Как только это значение будет достигнуто, новые сессий больше не будут открыты. Допустимые значения 1 и выше;
- *Oci Session Pool Min Size* – задает минимальное число сессий в пуле сессии. Это количество сессий будет запущено изначально. После этого сессии будут открываться только при необходимости;
- *Oci Session Pool Password* – определяет пароль для пользователя прокси, если установлен;
- *Oci Session Pool User Id* – определяет логин для пользователя прокси, если установлен. Доступно только при установке параметра *OCI Session Pooling* в значение *true*.

При использовании провайдера данных PostgreSQL (*Postgres*), доступны следующие параметры:

- *Host* – имя машины, на которой работает сервер базы данных PostgreSQL;
- *Port* – TCP-порт сервера PostgreSQL;
- *Database* – имя базы данных PostgreSQL для соединения;
- *Username* – имя пользователя PostgreSQL, являющегося владельцем схемы *ultima*;
- *Password* – пароль пользователя PostgreSQL;
- *Enlist* – включает (при значении *true*) или выключает автоматическое прикрепление к распределенной транзакции в существующих транзакциях;
- *Pooling* – в значение *true* используется пулинг соединений. Пулинг соединений позволяет сократить время обращения к БД. Значение по умолчанию *true*;
- *MinPoolSize* – минимальное количество подключений в пуле. Значение по умолчанию 1;
- *MaxPoolSize* – максимальное количество подключений в пуле, может повлиять на производительность. Значение по умолчанию 20;
- *Timeout* – время ожидания (в секундах) при установке соединения перед отменой попытки и выдачей ошибки. Значение 0 означает отсутствие ограничения;
- *Command Timeout* – время ожидания (в секундах) при выполнении команды перед отменой попытки и выдачей ошибки. Значение 0 означает отсутствие ограничения.

DuplexProtocol – использовать встроенный канал драйвера связи с клиентскими приложениями или стандартный .NET Remoting, имеет следующие значения:

- *true* – используется дуплексный канал (рекомендуемое значение);
- *false* – используется стандартный канал .NET Remoting.

SecureProtocol – включить шифрование. Обратите внимание: настройка *SecureProtocol* в файле конфигурации клиента должна совпадать с серверной. Имеет следующие значения:

- *true* – шифрование включено;

- *false* – шифрование выключено (значение по умолчанию).

AssembliesFolderName – название папки, в которую загружаются модули сервера.

AppServerID – идентификатор сервера приложений, совпадает с идентификатором данного сервера приложений в системе Ultimate Solid (подробнее см. раздел [Сервера и кластеры](#)).

EnableRestartService – включает или выключает встроенный веб-сервис для перезапуска сервера приложений.

ScriptCacheFolderName – название папки, в которую кэшируются скрипты.

LicenseFileName – название файла лицензии;

TaskStatSyncInterval – интервал (в мс) опроса сервера БД планировщиком выполнения заданий;

TaskExecutorLogin – имя пользователя (системы Ultimate Solid), под которым будет выполняться задание;

TaskExecutorPassword – пароль для указанного логина;

TaskSchedulerActive – работает ли планировщик (*True* – да, *False* – нет) на этом сервере приложений. Задания должны выполняться только на одном сервере приложений системы (кластера).



Для настройки компьютера, предназначенного для функционирования сервера приложений, необходимо выполнить следующие действия:

- установить на компьютер операционную систему семейства Windows;
- пользователь операционной системы, под которым запускается сервер приложений, должен иметь права администратора;
- установить платформу .NET Framework версии не ниже 4.6.1;
- скопировать дистрибутив AppServer;
- отредактировать файл конфигурации *UltimaService.exe.config*;
- дополнительные настройки сервера приложений осуществляются в модуле администратора клиентского приложения Ultimate Solid и описаны в разделе [Сервера и кластеры](#).

На одном сервере может быть установлено несколько служб сервера приложений. У каждой службы должен быть свой файл настроек *.exe.config. Файл настроек имеет такое же имя, как исполняемый файл приложения, поэтому потребуется несколько исполняемых файлов. Можно установить службы в отдельные папки, или же просто скопировать файл ConsoleServer.exe в одной и той же директории. Каждому файлу сервера будет соответствовать свой файл конфигурации:

```
ConsoleServer.exe
ConsoleServer.exe.config
ConsoleServer2.exe
ConsoleServer2.exe.config
```

Для каждой службы сервера необходимо указать свое наименование экземпляра (instance). Для этого используется опция instance командной строки:

```
ConsoleServer.exe install -instance:First
ConsoleServer2.exe install -instance:Second
```

Для запуска, остановки или удаления службы необходимо указывать экземпляр:

```
ConsoleServer.exe start -instance:First
ConsoleServer2.exe stop -instance:Second
ConsoleServer.exe uninstall -instance:First
```

По умолчанию служба устанавливается с именем **UltimaService**. При установке ей можно указать другое имя с помощью опции командной строки `-servicename`. Эту опцию можно комбинировать с параметром `-instance`:

```
ConsoleServer.exe install -servicename:Ultima -instance:First
ConsoleServer.exe start -servicename:Ultima -instance:First
ConsoleServer.exe stop -servicename:Ultima -instance:First
ConsoleServer.exe uninstall -servicename:Ultima -instance:First
```

Таймаут транзакций

Во время длинных (дольше десяти минут) операций сервер может выдать следующую ошибку:

```
ExecuteNonQuery exception:
The transaction associated with the connection has completed but not yet disposed.
Dispose the transaction to execute SQL statements on this connection.
```

Это происходит вследствие того, что транзакции (классы из пространства имен *System.Transactions*, появившиеся в .NET 2.0) по умолчанию имеют максимальный таймаут, равный 10 минутам. При старте сервер приложений проверяет настройки системы и пишет предупреждение об этом:

```
Maximum transaction timeout: 10
To set the infinite timeout, adjust your machine.config file as follows...
```

Для решения проблемы следует отредактировать файл *machine.config*, который можно найти в папке:

```
%WINDIR%\Microsoft.NET\Framework64\%номер версии%\Config
```

Настройка, снимающая ограничение на максимальную длительность транзакции, выглядит следующим образом:

```
<configuration>
  <system.transactions>
    <machineSettings maxTimeout="00:00:00" />
  </system.transactions>
</configuration>
```

Сервер печати

Принимает задания на печать документов от серверов приложений расположенных в дата-центре. Как правило, сервер печати располагается в офисе, где используются клиентские приложения – это позволяет существенно снизить нагрузку на каналы связи за счет передачи гораздо меньшего объема данных.

Использование внутренних механизмов печати позволяет обеспечить гарантированную доставку печатных форм до принтера, централизованную запись журнала печати (кто, когда, что, сколько и куда напечатал) и прочие функции.

Сервер, на котором установлен сервер печати, и сеть, в которой он находится, должны быть настроены так, чтобы с серверов приложений можно было установить соединение на сервер печати и с сервера печати установить соединение к любому из серверов приложений.

Сервер приложений кеширует шаблоны печатных форм для снижения нагрузки на канал данных. Если ведется активная разработка конфигурации, при которой шаблоны часто меняются, может

наблюдаться значительный рост размера кеша. Кеш можно удалить в любой момент, он начнет наполняться заново.

Сервер печати поддерживает печать пакетов документов. Пакет документов будет распечатан на одном принтере в заранее определенной последовательности, никаких других документов при печати пакета напечатано не будет (при условии, что никто кроме сервера печати не имеет доступа к принтеру).

Сервер печати может быть запущен как служба и как консольное приложение.

Настройки сервера печати **PrintServer.exe** задаются в файле **PrintServer.exe.config** в секции **userSettings.Ultima.PrintServer.Settings**:

```
<userSettings>
  <Ultima.PrintServer.Settings>
    <setting name="AppServerUrl" serializeAs="String">
      <value>tcpex://localhost:8192/UltimaServer</value>
    </setting>
    <setting name="PrintServerID" serializeAs="String">
      <value>1</value>
    </setting>
    <setting name="AdditionalThreadsCount" serializeAs="String">
      <value>5</value>
    </setting>
    <setting name="AppServerLogin" serializeAs="String">
      <value>PrintServer</value>
    </setting>
    <setting name="AppServerPassword" serializeAs="String">
      <value>password</value>
    </setting>
    <setting name="TcpPort" serializeAs="String">
      <value>1024</value>
    </setting>
    <setting name="SecureProtocol" serializeAs="String">
      <value>False</value>
    </setting>
    <setting name="PrintWorkerApplicationPath" serializeAs="String">
      <value>PrintWorker\PrintWorkerApp.exe</value>
    </setting>
    <setting name="TaskBuildingTimeOut" serializeAs="String">
      <value>30</value>
    </setting>
    <setting name="PagePrintingTimeOut" serializeAs="String">
      <value>5</value>
    </setting>
    <setting name="UseTemplatesCache" serializeAs="String">
      <value>True</value>
    </setting>
    <setting name="TasksRepositoriesPath" serializeAs="String">
      <value>LocalRepository\TasksRepositories</value>
    </setting>
    <setting name="TemplatesRepositoryPath" serializeAs="String">
      <value>LocalRepository\TemplatesRepository</value>
    </setting>
    <setting name="RetryCount" serializeAs="String">
      <value>20</value>
    </setting>
  </Ultima.PrintServer.Settings>
</userSettings><?xml version="1.0" encoding="utf-8"?>
```

AppServerUrl – адрес сервера приложений. Этот сервер приложений будет использоваться для получения шаблонов, сохранения статистики и прочего. Подойдет любой гарантированно работающий сервер приложений.

PrintServerID – идентификатор сервера печати в системе.

AdditionalThreadsCount – доступное количество дополнительных потоков для обработки заданий. Не рекомендуется, чтобы число дополнительных потоков превышало число виртуальных ядер компьютера, на котором работает сервер печати.

AppServerLogin – логин для подключения к серверу приложений. В качестве него рекомендуется использовать изначально заведенного системного пользователя *PrintServer*.

AppServerPassword – пароль для подключения к серверу приложений.

PrintWorkerApplicationPath – путь к приложению Print Worker.

TcpPort – порт, на котором работает сервер печати.

SecureProtocol – включить шифрование. Обратите внимание: настройка *SecureProtocol* сервера печати должна совпадать с настройкой сервера приложений. Имеет следующие значения:

- *true* – шифрование включено;
- *false* – шифрование выключено (значение по умолчанию).

TaskBuildingTimeOut – максимальное время ожидания построения формы печати (рендеринга) в секундах.

PagePrintingTimeOut – время в секундах, отведенной на печать одной страницы. Рекомендуется устанавливать его чуть больше времени печати одной страницы самым медленным принтером в сети.

RetryCount – количество попыток печати задания в случае возникновения ошибки. По достижении указанного параметром числа неудачных попыток задание удаляется из очереди.

UseTemplatesCache – кэширование шаблонов печатных форм, имеет следующие значения:

- *true* – шаблоны кэшируются локально и при повторном их использовании обращения к серверу приложений не происходит. Кэширование позволяет сократить трафик между серверами и снизить нагрузку на сервер приложений.



При изменении печатной формы внутри версии конфигурации кэш не сбрасывается. Использовать эту функцию можно только с зафиксированной версией конфигурации.

- *false* – шаблоны каждый раз загружаются с сервера приложений. Эта возможность необходима для разработчиков, либо если кластер (сервер приложений) работает с незафиксированной версией конфигурации, изменения в которую могут вноситься в режиме реального времени.

TasksRepositoriesPath – путь к папке с кэшем заданий печати.

TemplatesRepositoryPath – путь к папке с кэшем шаблонов печатных форм.



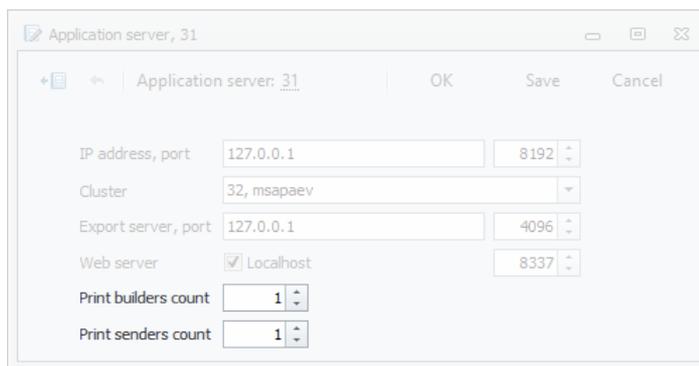
Для настройки компьютера, предназначенного для функционирования сервера печати, выполните следующие действия:

- установите на компьютер операционную систему семейства Windows;
- Если сервер будет использоваться только для сервиса печати отключите службу диспетчера очереди печати (Print Spooler). Отключение необходимо для того, чтобы гарантировать неразрывную печать пакетов документов и исключить потери заданий на печать. При отключении этой службы стандартное клиентское приложение не сможет получить список доступных принтеров;
- установите платформу .NET Framework версии не ниже 4.5;
- установите и подключите принтеры к серверу печати;

- беспроводные принтеры, используемые в одноранговой сети, также следует подключить к серверу печати;
- скопируйте на сервер печати дистрибутив PrintServer;
- в справочнике [Printers](#) приложения Ultimate Solid внесите сервер печати, указав IP адрес настраиваемого компьютера и порт, на котором будет функционировать PrintServer;
- добавьте в этом же справочнике для созданного сервера печати принтеры аналогично установленным на компьютере. Системное название принтера *System name* должно совпадать с названием принтера в операционной системе настраиваемого компьютера;
- выдайте пользователям [разрешения](#) печатать на созданных принтерах;
- отредактируйте файл конфигурации сервера печати *PrintServer.exe.config*, указав идентификатор сервера печати (ID созданного сервера печати в системе), порт, на котором будет работать сервер печати (аналогично указанному ранее в системе), и прочие параметры;
- убедитесь, что в файле конфигурации *ConsoleServer.exe.config* хотя бы одного сервера приложений кластера включен планировщик задач (задания отправляются на печать системной задачей):

```
<setting name="TaskSchedulerActive" serializeAs="String">  
  <value>True</value>  
</setting>
```

- также в [конфигурации](#) хотя бы одного из серверов приложений кластера должны быть заданы хотя бы по одному потоку для обработки (*Print builders count*) и отправки (*Print senders count*) заданий на печать:



Подробнее процесс печати описан в разделе [Описание процесса печати](#).

Сервер экспорта

Сервер экспорта (Export server) используется для преобразования печатных форм в файлы.

Адрес сервера экспорта прописывается в конфигурации кластера. Кроме того, адрес может быть прописан в конфигурации конкретного сервера приложений в случае, если необходимо, чтобы данный сервер приложений использовал отдельный сервер экспорта (для сервера приложений, входящего в кластер, будет использоваться сервер экспорта прописанный в его – сервера приложений – конфигурации).

Сервер экспорта использует системно-зависимые компоненты и может исполняться только в операционной системе, где установлена платформа .NET FrameWork. Сервер приложений и сервер экспорта рекомендуется размещать в одной (локальной) сети, чтобы снизить нагрузку на каналы связи, поскольку между ними будет передаваться значительный объем данных.

Сервера экспорта может быть установлен как **служба** – *UltimaExportServer.exe* или **консольное приложение** – *ExportConsoleServer.exe*. Параметры их конфигураций идентичны и хранятся в соответствующих файлах:

- UltimaExportServer.exe.config;
- ExportConsoleServer.exe.config:

```
<userSettings>
  <Ultima.Exports.ExportSettings>
    <setting name="TcpPort" serializeAs="String">
      <value>4096</value>
    </setting>
    <setting name="AppServerUrl" serializeAs="String">
      <value>tcpex://localhost:8192/UltimaServer</value>
    </setting>
    <setting name="AppServerLogin" serializeAs="String">
      <value>exportserver</value>
    </setting>
    <setting name="AppServerPassword" serializeAs="String">
      <value>password</value>
    </setting>
    <setting name="SecureProtocol" serializeAs="String">
      <value>False</value>
    </setting>
    <setting name="UseTemplatesCache" serializeAs="String">
      <value>True</value>
    </setting>
  </Ultima.Exports.ExportSettings>
</userSettings>
```

TcpPort – порт, на котором работает сервер экспорта.

AppServerUrl – адрес сервера приложений для загрузки шаблонов печатных форм. Подойдет любой гарантированно работающий сервер.

AppServerLogin – логин для подключения к серверу приложений. В качестве него рекомендуется использовать изначально заведенного системного пользователя *ExportServer*.

AppServerPassword – пароль для подключения к серверу приложений.

SecureProtocol – включить шифрование. Обратите внимание: настройка *SecureProtocol* сервера экспорта должна совпадать с настройкой сервера приложений. Имеет следующие значения:

- *true* – шифрование включено;
- *false* – шифрование выключено (значение по умолчанию).

UseTemplatesCache – кэширование шаблонов печатных форм, имеет следующие значения:

- *true* – шаблоны кэшируются локально и при повторном их использовании обращения к серверу приложений не происходит. Кэширование позволяет сократить трафик между серверами и снизить нагрузку на сервер приложений.



При изменении печатной формы внутри версии конфигурации кэш не сбрасывается. Использовать эту функцию можно только с зафиксированной версией конфигурации.

- *false* – шаблоны каждый раз загружаются с сервера приложений. Эта возможность необходима для разработчиков, либо если кластер (сервер приложений) работает с незафиксированной версией конфигурации, изменения в которую могут вноситься в режиме реального времени.



Для настройки компьютера, предназначенного для функционирования сервера экспорта, необходимо выполнить следующие действия:

- установить на компьютер операционную систему семейства Windows;
- установить платформу .NET Framework версии не ниже 4.5;
- скопировать дистрибутив ExportServer;
- отредактировать файл конфигурации сервера экспорта *ExportServer.exe.config*, указав порт, на котором он будет работать, и прочие параметры.
- в справочнике [Clusters](#) приложения Ultimate Solid прописать сервер экспорта в конфигурацию кластера (кластеров) и/или сервера (серверов) приложений, указав IP адрес настраиваемого компьютера и порт, прописанный ранее в конфигурации *ExportServer.exe.config*.

Клиентское приложение



Настройки клиентского приложения **e ClientLoader.exe** устанавливаются в файле **ClientLoader.exe.config** в секции *userSetting.Ultima.ClientProperties.Settings*:

```
<userSettings>
  <Ultima.Client.Properties.Settings>
    <setting name="AuthMethod" serializeAs="String">
      <value>Interactive</value>
    </setting>
    <setting name="Login" serializeAs="String">
      <value>root</value>
    </setting>
    <setting name="Password" serializeAs="String">
      <value>n/a</value>
    </setting>
    <setting name="AutomaticServerSelection" serializeAs="String">
      <value>False</value>
    </setting>
    <setting name="ModuleCacheFolderName" serializeAs="String">
      <value>ClientModules</value>
    </setting>
    <setting name="MapCacheFolder" serializeAs="String">
      <value>MapCache</value>
    </setting>
    <setting name="ClientApplicationID" serializeAs="String">
      <value>1</value>
    </setting>
    <setting name="UpdateModules" serializeAs="String">
      <value>True</value>
    </setting>
    <setting name="DuplexProtocol" serializeAs="String">
      <value>True</value>
    </setting>
    <setting name="SecureProtocol" serializeAs="String">
      <value>False</value>
    </setting>
    <setting name="ThreadingAlerts" serializeAs="String">
      <value>True</value>
    </setting>
    <setting name="ScreenDpiCheckEnabled" serializeAs="String">
      <value>True</value>
    </setting>
  </Ultima.Client.Properties.Settings>
</userSettings>
```

```

    <setting name="StartupCommands" serializeAs="String">
      <value />
    </setting>
    <setting name="ServerAddresses" serializeAs="Xml">
      <value>
        <ArrayOfServerAddress
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns:xsd="http://www.w3.org/2001/XMLSchema">
          <ServerAddress>
            <Host>localhost</Host>
            <Port>8192</Port>
            <DuplexProtocol>true</DuplexProtocol>
            <Quality>1</Quality>
          </ServerAddress>
        </ArrayOfServerAddress>
      </value>
    </setting>
  </Ultima.Client.Properties.Settings>
</userSettings>

```

AuthMethod — выбирает метод авторизации пользователя при запуске приложения, имеет следующие значения:

- *interactive* — графический интерфейс ввода логина/пароля;
- *automatic* — полностью автоматический вход.

Login — логин, под которым осуществляется автоматический вход в приложение.

Password — пароль для автоматического входа.

AutomaticServerSelection — определяет возможность выбора пользователем при запуске приложения сервера приложений:

- *true* — пользователь не имеет возможности выбирать сервер приложений самостоятельно;
- *false* — пользователь имеет возможность выбирать сервер приложений самостоятельно.

ModuleCacheFolderName — папка, в которой сохраняются модули клиентского приложения.

MapCacheFolder — папка, используемая для хранения кэша карт.

ClientApplicationID — код клиентского приложения (перечень клиентских приложений хранится в базе данных). Стандартное клиентское приложение Windows Forms на базе модулей имеет код 1. Примерами других приложений могут быть терминал печати накладных, монитор выдачи и т. п. Для обычной разработки клиентских модулей эту настройку менять не нужно.

UpdateModules — загрузка модулей клиентского приложения, имеет следующие значения:

- *true* — при запуске клиентского приложения его модули загружаются с сервера приложений и сохраняются в папку, заданную параметром *ModuleCacheFolderName*;
- *false* — модули загружаются локально из папки, заданной параметром *ModuleCacheFolderName*.

DuplexProtocol — канал связи с сервером приложений, должен быть таким же, что и для сервера приложений, имеет следующие значения:

- *true* — используется дуплексный канал (рекомендуемое значение);
- *false* — используется стандартный канал .NET Remoting.

SecureProtocol — включить шифрование. Обратите внимание: настройка *SecureProtocol* в файле конфигурации клиента должна совпадать с серверной. Имеет следующие значения:

- *true* — шифрование включено;
- *false* — шифрование выключено (значение по умолчанию).

ThreadingAlerts — выдавать предупреждения при выполнении в клиентском приложении синхронных вызовов удаленных методов. Настройка полезна в первую очередь прикладным разработчикам и пользователям, работающим с тестовой версией конфигурации:

- *true* — выдавать предупреждения;
- *false* — не выдавать предупреждения.

ScreenDpiCheckEnabled — проверка DPI экрана при старте приложения.

- *true* — проверять настройки системы и прекращать работу при неподдерживаемых значениях DPI;
- *false* — пропустить проверку.

StartupCommands — список команд, выполняемых при старте клиентского приложения (GUID-ы команд модулей, перечисленные через запятую). В качестве команды, выполняемой при старте системы, может использоваться, к примеру, наиболее часто используемый журнал документов (или несколько таких журналов), окно приветствия или совет дня.

ServerAddresses — перечень серверов приложений, с которыми может работать клиентское приложение. Задается массивом *ArrayOfServerAddress*, который включает один или более элементов *ServerAddress*, имеющих следующие параметры:

- *Host* — IP-адрес сервера приложений;
- *Port* — порт сервера приложений;
- *DuplexProtocol* — канал связи с сервером приложений, аналогично описанному выше;
- *Quality* — вес, согласно которому вычисляется вероятность подключения клиентского модуля именно к этому серверу приложений, если их в массиве *ArrayOfServerAddress* более одного. Вероятность равна отношению веса данного сервера к сумме весов всех серверов вместе взятых. Например, в массиве прописано три сервера приложений, у первых двух из которых параметр *Quality* равен "1", а у третьего "2". В этом случае клиентский модуль при подключении к серверу приложений с вероятностью 25% выберет один из двух первых серверов и с вероятностью 50% последний.



Для настройки компьютера, предназначенного для функционирования клиентского приложения, необходимо выполнить следующие действия:

- установить на компьютер операционную систему семейства Windows;
- установить платформу .NET Framework версии не ниже 4.6;
- скопировать дистрибутив Client;
- отредактировать файл конфигурации *ClientLoader.exe.config*.
- запуск клиентского приложения настоятельно рекомендуется осуществлять через [утилиту обновления ClientUpdater.exe](#).

Обновление клиентского приложения

Обновление клиентского приложения осуществляется с помощью утилиты  **ClientUpdater.exe**, которая входит в дистрибутив Ultimate Solid и находится в одной папке с клиентским приложением.

Настройки утилиты хранятся в XML-файле **ClientUpdater.exe.config** (при первом запуске утилиты можно скопировать или переименовать файл *ClientUpdater.exe.config.default*). Настройки, которые может менять администратор, находятся в секции *appSettings*:

```
<appSettings>
  <add key="UpdateServiceUrl" value="http://localhost:8337" />
  <add key="WindowShowingDelay" value="1000" />
</appSettings>
```

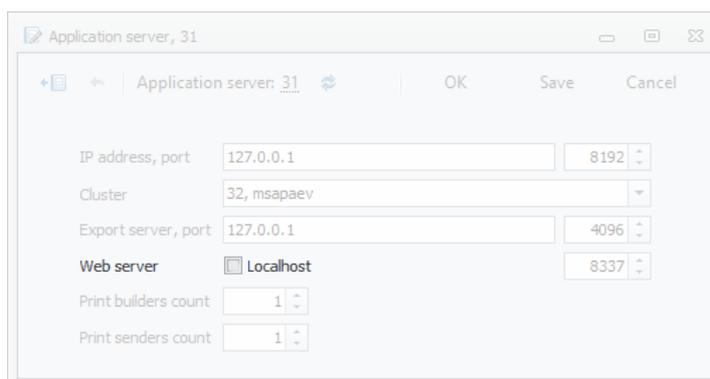
UpdateServiceUrl – адрес сервера приложений, на котором выложены обновления. На сервере приложений должен быть обязательно настроен web сервер. В качестве порта указывает порт именно web сервера.

WindowShowingDelay – время отображения формы утилиты обновления в миллисекундах в случае отсутствия необходимости обновлять клиентское приложение (отсутствия обновлений).



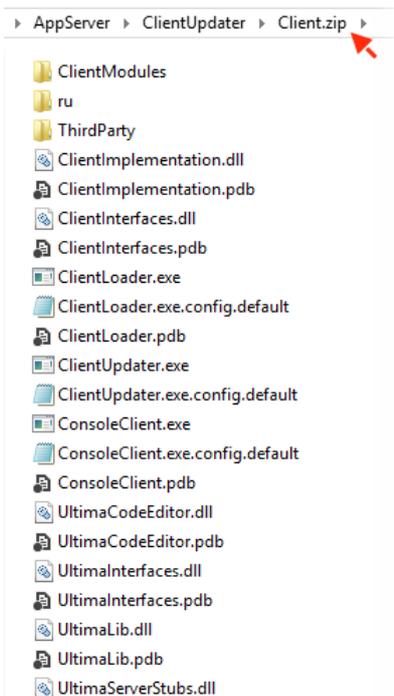
Для работы утилиты обновления клиентского приложения необходимо:

- на сервере приложений, который будет выступать в роли сервера обновлений:
 - убедиться, что на сервере приложений настроен web сервер. По умолчанию он запущен локально. Чтобы он был доступен извне, следует снять в [настройках сервера приложений](#) флаг *Localhost*:



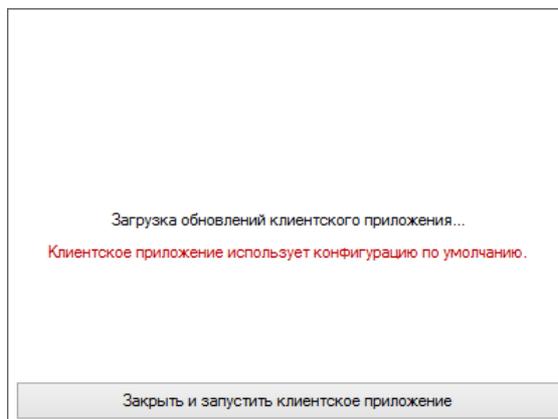
- если сервер приложений запускается как консольное приложение, он должен быть запущен с правами администратора;

- архив с актуальным дистрибутивом клиентского приложения должен быть размещен в папке: *AppServer/ClientUpdater/Client.zip*;
- в архив *Client.zip* должно быть добавлено только содержимое папки *Client* клиентского приложения, но не сама папка *Client*;
- в архиве с дистрибутивом обновления не должно быть файла конфигурации клиентского приложения *ClientLoader.exe.config*. В противном случае в процессе обновления файл конфигурация пользователя будет перезаписан;
- на компьютере клиента необходимо отредактировать конфигурационный файл утилиты обновления *ClientUpdater.exe.config*, задав корректный адрес сервера обновлений *UpdateServiceUrl* (в качестве порта указывается порт web сервера).



При запуске утилиты *ClientUpdater.exe*:

- проверяется текущая версия клиентского приложения;
- если обновление не требуется, клиентское приложение запускается автоматически по истечении времени задержки, заданного параметром *WindowShowingDelay* файла конфигурации;
- если обновление требуется:
 - последовательно осуществляется загрузка и применение обновлений клиентского приложения;
 - в случае успешного обновления клиентское приложение запускается автоматически;
 - если в папке клиентского приложения нет файла конфигурации, используется файл конфигурации по умолчанию, о чем выдается сообщение в форме утилиты;



Журналирование

Системой Ultimate Solid поддерживается два способа журналирования:

- основной вариант – централизованное журналирование на базе библиотеки [Serilog](#), поддерживающей большой ассортимент хранилищ разной степени структурированности: от текстовых файлов до документоориентированных баз данных. Встроенный в систему [анализатор логов](#) работает с базой данных MongoDB 3.0, которая позволяет писать достаточно сложные запросы к свойствам документов, в том числе с регулярными выражениями и поиском с учетом русской морфологии. При желании, однако, можно использовать любое поддерживаемое библиотекой Serilog хранилище, в том числе и NLog, используя собственные инструменты анализа;
- дополнительный вариант – журналирование на платформе [NLog](#), которая поддерживает множество способов вывода логов – от файла и консоли, в которой запущено приложение, до журнала событий и электронной почты.

Каждый метод имеет свои преимущества и недостатки.

Так Serilog предлагает:

- структурированное хранение событий (в виде текста и дополнительных свойств);
- большой набор баз, форматов и доступных инструментов для анализа;
но при этом:
- не умеет перегружать настройки из файла конфигурации на лету;
- не позволяет нескольким процессам писать в один лог-файл.

В то же время NLog:

- позволяет нескольким процессам писать в один лог-файл. Это актуально для приложения [PrintWorker](#);
однако имеет и ряд недостатков:
- на каждом сервере свои файлы логов, для анализа которых необходимо иметь доступ к каждому из них;
- файлы логов еще необходимо найти (этот и предыдущий пункты актуальны в большей степени для прикладного разработчика);
- файлы довольно велики и не всегда под рукой есть программа, способная их открыть;
- когда файлов много, сложно определить, какие из них актуальны на текущий момент;
- поскольку текст не структурирован, поиск часто натывается на лишние совпадения;
- поиск с регулярными выражениями доступен не во всякой программе;
- иногда логи обрываются на самом интересном месте.

При этом Serilog умеет в качестве хранилища использовать NLog, что полностью нивелирует его недостатки. В этом случае потребность писать в один файл из нескольких процессов легко разрешить путем перенаправления событий в NLog.

Выбор метода журналирования – только Serilog на базе MongoDB или своего хранилища, или же Serilog в связке с NLog – остается за системным администратором/прикладным разработчиком.

Serilog

Библиотеки Serilog уже входят в дистрибутив приложения Ultimate Solid и в дополнительной установке не нуждаются.

Базу данных MongoDB 3.0, которая поддерживается встроенным в систему [анализатором логов](#), следует предварительно запустить и настроить:

- подготовить пустую папку для базы данных и выполнить команду:

```
mongod.exe --dbpath D:\Tools\MongoDB30\data
```

- для запуска на движке Wired Tiger потребуется другая пустая папка и дополнительный параметр `--storageEngine`:

```
mongod.exe --storageEngine wiredTiger --dbpath D:\Tools\MongoDB30\data\wt
```

- пример конфигурирования работающей базы событий:

```
// ограничить размер коллекции событий объемом в 4 гигабайта
db.runCommand({"convertToCapped": "log", size: 4*1024*1024*1024});

// проиндексировать интересующие свойства
db.log.createIndex({"Level": 1});
db.log.createIndex({"Timestamp": 1});
db.log.createIndex({"Properties.SourceContext": 1});
db.log.createIndex({"Properties.Application": 1});
db.log.createIndex({"Properties.UserID": 1});
db.log.createIndex({"Properties.SessionID": 1});
db.log.createIndex({"Properties.ServerCallID": 1});
```

- поскольку в журнал сохраняются SQL-запросы, в логе может попадаться секретная информация. По этой причине необходима авторизация. Для записи логов создается пользователь, имеющий права только на запись данных, пароль которого не нужно прятать от посторонних глаз. Для чтения логов создаются пользователи по числу программистов:

```
// Execute script using
// mongo.exe localhost/logs/setupSerilog.js
// In the interactive shell, type: use logs;

// create admin user
db.createUser(
  {
    user: "logAdminUser",
    pwd: "*****",
    roles: [
      { role: "userAdmin", db: "logs" },
      { role: "readWrite", db: "logs" }
    ]
  }
);

// create a write-only role for serilog writer
db.runCommand(
  {
    createRole: "logWriter",
    privileges: [
      { resource: {db: "logs", collection: "log"}, actions: ["insert"]}
    ],
    roles: []
  }
);

// create serilog user
db.createUser(
```

```
{
  user: "serilog",
  pwd: "serilog",
  roles: [
    { role: "logWriter", db: "logs" }
  ]
}
);
```

- после настройки пользователей MongoDB следует перезапустить с ключом `--auth` для включения авторизации:

```
mongod.exe --auth --storageEngine wiredTiger --dbpath D:\Tools\MongoDB30\datawt
```

Подробную информацию о конфигурировании хранилища MongoDB можно найти на сайте проекта www.mongodb.org. Ниже приведены лишь некоторые замечания:

- файлы базы данных могут расти довольно быстро;
- MongoDB 3.0 поддерживает упакованный формат хранилища [Wired Tiger](#);
- размеры коллекций [можно ограничивать](#) (новые события перезапишут самые старые);
- интересные с точки зрения фильтрации свойства [следует индексировать](#);
- настраивать ограничения и индексировать можно на лету;
- для перехода на упакованный формат следует перезапустить сервер MongoDB;
- встроенная оболочка Mongo не слишком удобна, возможно стоит обратить внимание на [Robomongo](#) (пока не работает с Wired Tiger) или [MongoChef](#).

Настройка записи логов осуществляется для каждого из приложений в отдельности в файле его конфигурации. В приведенном примере настроено журналирование в файл, консоль и хранилище MongoDB:

```
<appSettings>
  <add key="serilog:minimum-level" value="Verbose" />
  <add key="serilog:enrich:with-property:ProjectName" value="ProjectName" />
  <add key="serilog:write-to:Console.restrictedToMinimumLevel" value="Information" />
  <add key="serilog:write-to:File.path" value="ConsoleServer.log" />
  <add key="serilog:write-to:File.restrictedToMinimumLevel" value="Debug" />
  <add key="serilog:using:Mongo" value="Serilog.Sinks.MongoDB" />
  <add key="serilog:write-to:MongoDB.databaseUrl"
    value="mongodb://serilog:serilog@192.168.102.14:27017/logs" />
</appSettings>
```

Serilog поддерживает концепцию минимального уровня *MinimumLevel*. Если он задан, события ниже минимального уровня не журналируются. Существуют следующие уровни событий в порядке повышения их критичности:

- *Verbose*;
- *Debug*;
- *Information*;
- *Warning*;
- *Error*;
- *Fatal*.

Если минимальный уровень не задан, журналируются все события уровня *Information* и выше.

Более подробную информацию о Serilog можно найти на сайте библиотеки serilog.net.

NLog

Библиотеки платформы NLog уже входят в дистрибутив приложения Ultimate Solid и в дополнительной установке не нуждаются.

Настройка журналирования осуществляется для каждого из приложений в отдельности в файле его конфигурации. Однако, для корректного журналирования платформой NLog необходимо в тех же файлах конфигурации также сконфигурировать и Serilog, чьи настройки выделены тегом *appSettings*. В приведенном примере Serilog пишет логи в NLog и базу MongoDB:

```
<appSettings>
  <add key="serilog:minimum-level" value="Verbose" />
  <add key="serilog:enrich:with-property:ProjectName" value="ProjectName" />
  <add key="serilog:using:NLog" value="Serilog.Sinks.NLog" />
  <add key="serilog:using:Mongo" value="Serilog.Sinks.MongoDB" />
  <add key="serilog:write-to:NLog.restrictedToMinimumLevel" value="Verbose" />
  <add key="serilog:write-to:MongoDB.databaseUrl"
    value="mongodb://serilog:serilog@192.168.102.14:27017/logs" />
</appSettings>
```

Подробно информацию по настройке NLog можно найти на сайте платформы → nlog-project.org. Ниже приведены лишь несколько типовых примеров конфигурации.

Настройка журналирования NLog для [сервера приложений](#) (*ConsoleServer.exe.config*). Реализован вывод логов в консоль и файлы. Для некоторых быстрорастущих файлов логов настроено архивирование с перезаписью (архивируется после 100 Мб, хранится 10 последних архивов, нумерация файлов по счетчику):

```
<nlog autoReload="true" xmlns="http://www.nlog-project.org/schemas/NLog.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <targets>
    <target name="Console" xsi:type="ColoredConsole"
      layout="${pad:inner=${level}:padding=1:fixedLength=true}:${date:format=HH:mm:ss}
        ${logger}: ${message}${onexception:inner=${newline}
        ${exception:format=Message:maxInnerExceptionLevel=10:innerFormat=Message}}">
      <highlight-row backgroundColor="NoChange"
        condition="level == LogLevel.Info" foregroundColor="Gray"/>
    </target>
    <target name="TasksLog" xsi:type="File" fileName="${basedir}/TasksScheduler.log"
      layout="${date:format=yyyy.MM.dd HH:mm:ss} ${logger}: ${message}
        ${onexception:inner=${exception}}"/>
    <target name="ServerLog" xsi:type="File" fileName="${basedir}/ConsoleServer.log"
      layout="${date:format=yyyy.MM.dd HH:mm:ss} ${logger}: ${message}
        ${onexception:inner=${newline}
        ${exception:format=tostring:maxInnerExceptionLevel=10}"
      maxArchiveFiles="10" archiveNumbering="Sequence" archiveAboveSize="10000000"
      archiveFileName="${basedir}/ConsoleServer.Archive{####}.log" />
    <target name="ExceptionLog" xsi:type="File" fileName="${basedir}/Exceptions.log"
      layout="${newline}${date:format=yyyy.MM.dd HH:mm:ss}:
        ${exception:format=tostring,data:inner=tostring,data:maxInnerExceptionLevel=10}"
      maxArchiveFiles="10" archiveNumbering="Sequence" archiveAboveSize="10000000"
      archiveFileName="${basedir}/Exceptions.Archive{####}.log" />
    <target name="SqlLog" xsi:type="File" fileName="${basedir}/SqlStatements.log"
      layout="${newline}${date:format=yyyy.MM.dd HH:mm:ss}: ${message}"
      maxArchiveFiles="10" archiveNumbering="Sequence" archiveAboveSize="10000000"
      archiveFileName="${basedir}/SqlStatements.Archive{####}.log" />
    <target name="ServerCallLog" xsi:type="File" fileName="${basedir}/ServerCalls.log"
      layout="${date:format=yyyy.MM.dd HH:mm:ss}: ${message}"
      maxArchiveFiles="10" archiveNumbering="Sequence" archiveAboveSize="10000000"
      archiveFileName="${basedir}/ServerCalls.Archive{####}.log" />
  </targets>
  <rules>
```

```

<logger name="*" minlevel="Info" writeTo="Console"/>
<logger name="*" minlevel="Debug" writeTo="ServerLog"/>
<logger name="TasksScheduler" minlevel="Info" writeTo="TasksLog"/>
<logger name="FirstChanceException" minlevel="Trace" writeTo="ExceptionLog"/>
<logger name="Sql" minlevel="Trace" writeTo="SqlLog"/>
<logger name="ServerCallTracker" minlevel="Trace" writeTo="ServerCallLog"/>
</rules>
</nlog>

```

Настройка журналирования NLog для приложения PrintWorker (*PrintWorkerApp.exe.config*) [сервера печати](#). Реализована запись в один файл несколькими процессами (PID):

```

<nlog autoReload="true" xmlns="http://www.nlog-project.org/schemas/NLog.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <targets>
    <target name="Console" xsi:type="ColoredConsole"
      layout="${pad:inner=${level}:padding=1:fixedLength=true}:${date:format=HH:mm:ss}
        ${logger}: ${message}${onexception:inner=${newline}}
        ${exception:format=Message:maxInnerExceptionLevel=10:innerFormat=Message}">
      <highlight-row backgroundColor="NoChange"
        condition="level == LogLevel.Info" foregroundColor="Gray"/>
    </target>
    <target name="Log" xsi:type="File" fileName="${basedir}/PrintWorkerApp.log"
      layout="${date:format=yyyy.MM.dd\ HH:mm:ss} PID=${processid}: ${logger}:
        ${message}${onexception:inner=${newline}}
        ${exception:format=toString:maxInnerExceptionLevel=10}" autoFlush="true" />
  </targets>
  <rules>
    <logger name="*" minlevel="Info" writeTo="Console"/>
    <logger name="*" minlevel="Debug" writeTo="Log"/>
  </rules>
</nlog>

```

Описание процесса печати

Рассмотрим всю цепочку, которую проходит задание на печать.

1. Пользователь вызывает функцию печати на сервере приложений. Печать может быть:

- *синхронная (обычная)* – в этом случае:
 - выполняется обработчик, который сразу же вычисляет и создает объект с данными для печатной формы;
 - при этом клиент ожидает завершения работы этого обработчика;
- *асинхронная (отложенная)* – в этом случае данные будут сформированы позднее.

Если пользователь вызывает функцию печати непосредственно в экранной форме, например, печатая документ из его формы редактирования или записи справочника из его списковой формы, печать всегда будет синхронной. В случае, если печатать вызывается скриптом или обработчиком, выбор способа печати остается за прикладным разработчиком.

Прикладной разработчик может использовать асинхронную печать в том случае, когда он уверен, что по факту отложенного формирования данных для печати, пользователь все равно получит необходимый результат. Например, асинхронно можно напечатать документ отгрузки товара на склад, когда статус заказа изменится на "оплачено".

2. Сервер приложений сохраняет задание в базе данных.

3. Все сервера приложений, которым это разрешено [соответствующими настройками](#), обращаются к серверу базы данных и отправляют полученные от него задания на сервис печати (Print Service).

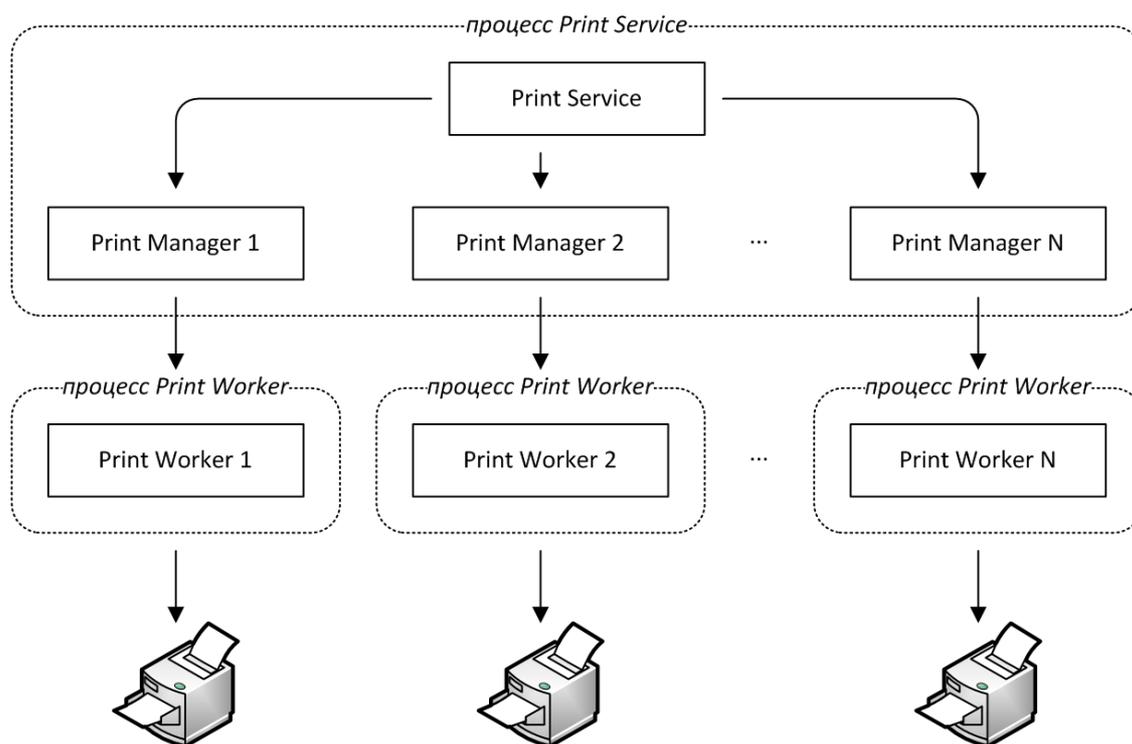
Сервера приложения распределяют задания неблокирующим образом по соответствующим заданиям принтсерверам. В очереди на каждый конкретный принтер задания раздаются в порядке их поступления – первым отсылается первое поступившее на печать задание, затем второе и т.д. Разбиение одной общей очереди печати на очереди по принтерам реализовано для того, чтобы принтеры не простаивали без работы. Это может случиться в том случае, когда, на один загруженный принтер попадает последовательно большое количество заданий, например, сто, а затем на печать отправляется сто первое задание на другой – простаивающий принтер, которому в случае общей очереди пришлось бы ожидать отправки предыдущих ста заданий.

Задания могут печататься пакетами. Это реализовано для случаев, когда нужно распечатать несколько печатных форм на один принтер и чтобы при этом заданная последовательность печати не была нарушена другими вклинившимся заданиями. Пакет заданий всегда будет печататься в определенной последовательности в независимости от очередности поступления на печать заданий пакета.

Если данные для печати не были рассчитаны сразу (случай асинхронной печати), перед отправкой на Print Service сервер приложений их вычислит. Сервер базы данных раздает такие задания только тем серверам приложений, которым операция расчета разрешена [соответствующими настройками](#).

В случае удачной отправки задания на Print Service оно удаляется из базы данных.

4. Print Service принимает задания от сервера приложений и распределяет их по менеджерам печати (Print Manager):



Для каждого принтера запускается отдельный Print Manager, который работает с заданиями на печать только на этот принтер. Если Print Service не находит Print Manager для какого-либо принтера (задание на принтер отсылается впервые с момента запуска Print Service) он создает новый Print Manager.

Все менеджеры печати и Print Service работают в рамках одного сервиса на сервере печати.

Каждый Print Manager:

- хранит локально в файле собственную очередь заданий на печать. Это позволяет восстановить задания на печать в случае сбоя и перезапуска сервиса печати;

- отсылает задание на принтер через отдельный специальный процесс Print Worker;
- задания хранятся в очереди до тех пор, пока Print Worker не напечатает их;

Print Worker – это отдельный процесс, который:

- имеет свою собственную очередь заданий, которые отсылаются на печать в порядке их поступления на Print Worker – первым отсылается первое поступившее на печать задание, затем второе и т.д.;
- строит (рендерит) печатную форму;
- отсылает созданную форму на принтер;
- непосредственно работает с драйвером принтера.

Каждое задание перед печатью рендерится. На это действие требуется определенное количество времени, в процессе которого принтер бездействует. Чтобы принтер по возможности не простаивал, рендеринг заданий возможен в отдельных дополнительных потоках. Основной поток получает очередное задание и рендерит его, только если этого еще не было сделано дополнительным потоком, после чего отсылает на печать.

Print Service управляет выделением дополнительных потоков пропорционально загрузке процессов Print Worker. Число дополнительных потоков, которым управляет Print Service, определяется параметром *AdditionalThreadsCount* в файле его конфигурации.

На процесс рендеринга задан таймаут, величина которого определяется в файле конфигурации Print Service параметром *TaskBuildingTimeOut*. По истечении этого таймаута задание на печать переходит в невалидное состояние и удаляется из очередей Print Worker и Print Manager. Пользователю, который отправил задание, отсылается уведомление о невозможности его напечатать.

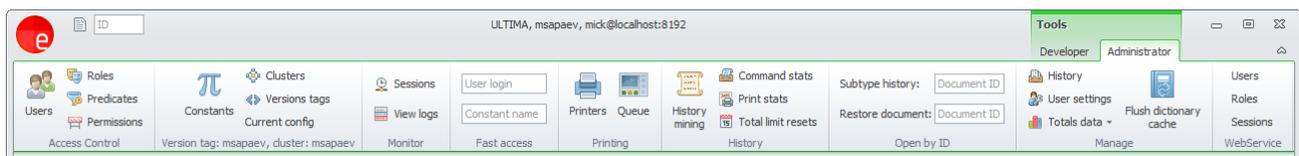
На процесс непосредственно печати также задан таймаут. Величина таймаута печати одной страницы определяется в файле конфигурации Print Service параметром *PagePrintingTimeOut*. Величина таймаута печати задания пропорциональна количеству страниц в нем и равна произведению [количество страниц задания]*[количество копий печати]*[величина таймаута]. По прошествии таймаута, если печать не завершилась, Print Service перезапускает соответствующий процесс Print Worker.

Смысл выделения Print Worker в отдельный процесс как раз и состоит в возможности его перезапуска при зависании драйвера принтера без потери заданий печати и перезапуска всего сервиса печати. В случае перезапуска Print Worker, Print Manager заново отправляет ему очередь заданий.

Функционал модуля администратора

Для получения доступа к модулю администратора пользователь должен обладать правом *Administrator*.

Все основные инструменты администраторы системы Ultimate Solid организованы на закладке "Administrator":



Инструменты разбиты по группам согласно их предназначению:

- *Access control* – инструменты работы с пользователями и их правами;
- инструменты настройки конфигурации системы;
- *Monitor* – инструменты контроля работы системы;

- *Fast access* – инструменты быстрого доступа к объектам;
- *Printing* – инструменты работы с печатью;
- *History* – инструменты работы с историей действий и логами;
- *Open by ID* – инструменты быстрого доступа к востребованным операциям над объектами;
- *Manage* – инструменты управления настройками;
- *WebService* – инструменты работы с web сервисами.

Кроме того, в меню  доступен инструмент [настройки главного меню](#) (пользовательского интерфейса) клиентского приложения.

Группа инструментов *Access control*

Любые манипуляции с данными в системе Ultimate Solid совершаются под каким-либо пользователем с конечным набором прав. Права определяют, может ли пользователь совершить то или иное действие или получить доступ к какому-либо объекту.

Идеология системы прав такова, что любое действие или возможность доступа к объекту изначально необходимо разрешить, то есть – что не разрешено, то запрещено.

В системе различаются следующие объекты прав:

 Права, сформулированные и проверяемые разработчиком. Эти права проверяются на уровне исполняемого кода. Таковыми могут быть, например, *Вход в систему под чужим именем* или *Разрешение администратора*;

 Разрешения на доступ к объектам метаданных:

-  справочникам;
-  документам;
-  итогам;
-  пользовательским командам, командам над справочниками и документами;
-  принтерам;

 Предикаты, позволяющие ограничить доступ пользователя на уровне отдельных строк;

 Клиентские модули.

 Роли объединяют все вышеперечисленные объекты прав:



Будучи назначенной пользователю, роль наделяет его соответствующими правами. Одному пользователю может быть назначена только одна роль. Роль в свою очередь может состоять из других ролей. Также обычно одна и та же роль назначается пользователям с одинаковым функционалом.

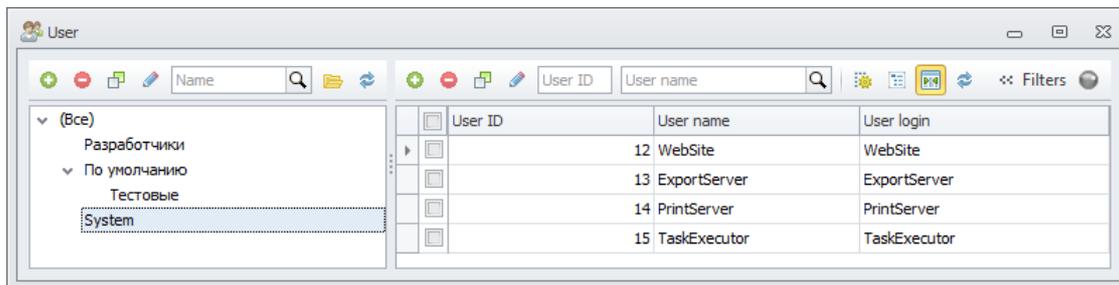


Предикаты не поддерживаются в версии системы для PostgreSQL.

Пользователи



Список всех пользователей системы можно найти в справочнике "User":

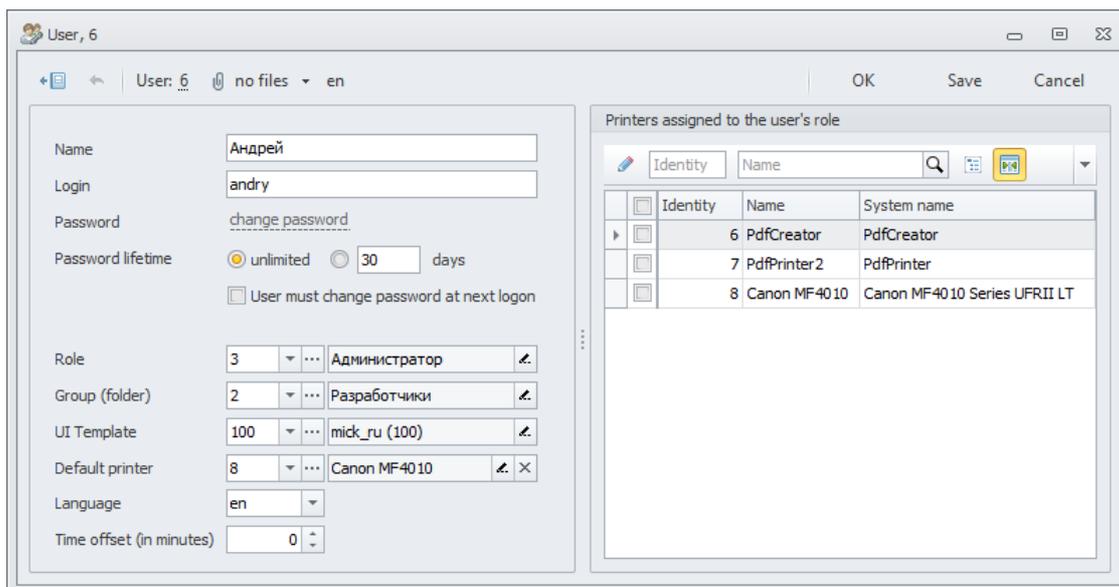


Окно справочника разделено на две части: слева отображается дерево групп пользователей, справа – список пользователей выбранной слева группы.

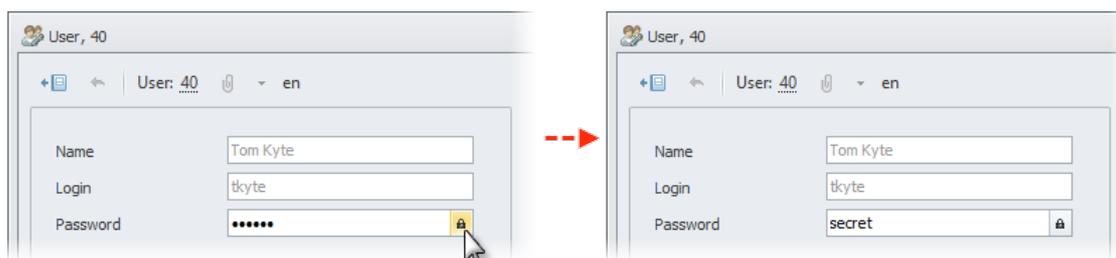
Группа *System* используется для *системных пользователей*, под которыми выполняются служебные обращения к базе данных без непосредственного участия сотрудников компании, например, работают *сервер печати* и *сервер экспорта*, запускаются *задачи*.

Группы пользователей можно фильтровать по *Названию* группы (*Name*), а пользователей по *Имени* (*User name*).

Каждый пользователь обладает следующими свойствами:



- *Name* – имя пользователя;
- *Login* – логин;
- *Password* – ссылка на смену пароля. Для созданного ранее пользователя пароль можно поменять по клику на ссылку *change password*. Для создаваемого пользователя пароль необходимо ввести. Вводимый пароль можно отобразить:



- *Password lifetime* – время действия пароля:
 - *unlimited* – не лимитировано по времени;
 - *days* – ограничено введенным в это поле количеством дней. По истечении указанного срока при очередном входе в систему пользователю будет предложено изменить пароль.
 - При выборе ограничения действия пароля по сроку, отображается дополнительное поле *Valid until*, где указана дата, по которую пароль пользователя действителен, в случае, если время его действия ограничено;
- *User must change password at next logon* – пользователь, которому установлен данный флаг, должен будет сменить пароль при следующем входе в систему. После смены пароля флаг будет автоматически снят;
- *Role* – роль пользователя, определяющая его права;
- *Group (folder)* – группа, которой принадлежит пользователь. Для создаваемого пользователя задается автоматически в соответствии с группой, выбранной в фильтре списковой форме справочника на момент создания;
- *UI Template* – пользовательский интерфейс;
- *Default printer* – принтер по умолчанию;
- *Language* – язык пользователя. Для создаваемого по умолчанию задается русский язык;
- *Time offset (in minutes)* – смещение времени пользователя в минутах относительно принятого в компании основного часового пояса. Применяется для учета разницы во времени между пользователями, работающими в разных часовых поясах. Основным часовым поясом может быть, например, часовой пояс головного офиса или UTC+0 (GMT+0). Для пользователей, работающих в основном часовом поясе, смещение времени равно "0".
Например, основным в компании принят часовой пояс Москвы (UTC+4). Тогда смещение времени для пользователей, работающих в Калининграде (UTC+3), будет "-60", а для пользователей Екатеринбурга (UTC+6) – "120";
- *Printers assigned to the user's role* – в списке отображаются принтеры, доступ к которым предоставляется ролью пользователя *Role*.

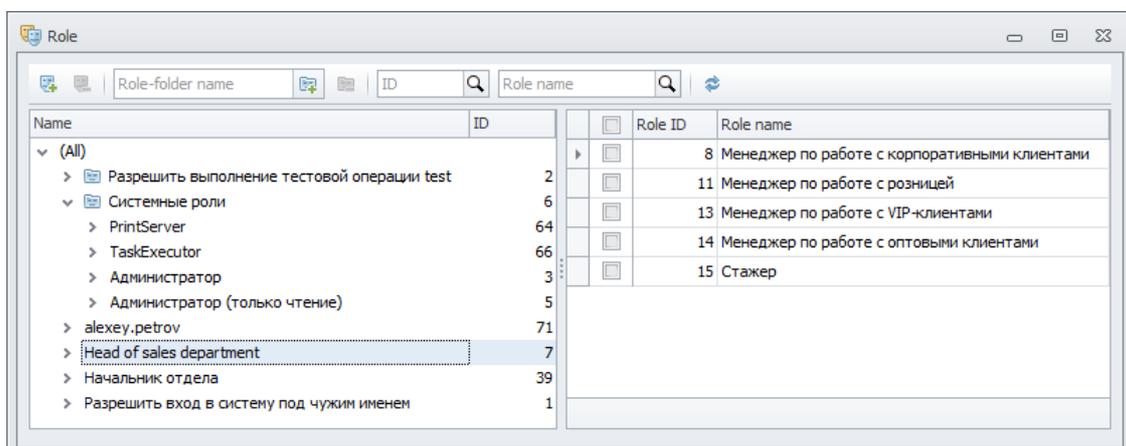


Автоматическая синхронизация времени с часовым поясом пользователя не поддерживается в версии для PostgreSQL, поэтому поле *Time offset* не отображается.

Роли



Роли организованы в справочнике Roles в древовидной структуре, где у каждой роли может быть несколько родителей и потомков:



Окно справочника разделено на две части: слева отображается дерево ролей, справа – дочерние роли первого уровня выбранной слева роли.

В дереве слева роли всегда отсортированы по алфавиту, а папки  всегда идут перед обычными ролями.

Записи справочника можно искать по *Названию роли (Role name)*. Поиск осуществляется по вхождению искомого текста в название роли при нажатии на кнопку  элемента управления или клавишу **Enter**. При удачном поиске курсор устанавливается на первую удовлетворяющую условиям найденную запись, при неудачном – остается на прежней. Повторное нажатие на кнопку поиска  приведет к нахождению следующей удовлетворяющей условию поиска записи и установке курсора на нее.

Роль также может быть **папкой**. Такие роли обозначаются в справочнике иконкой  перед названием. Роли-папки не наделены никакими правами. Они предназначены исключительно для удобства организации структуры ролей и используются, например, для их группировки. Функциональность ролей-папок ограничена рядом запретов:

- с помощью роли-папки нельзя выдать доступ к какому-либо объекту прав;
- роль-папку нельзя назначить пользователю;
- роль-папку нельзя сделать потомком для другой роли, если она также не является папкой.

Функционал справочника ролей позволяет выполнять следующие действия:

-  создавать новые роли. Новая роль будет создана на самом верхнем уровне независимо от того, какая роль выбрана слева в дереве;
-  удалять выбранные роли, не являющиеся папками;
-  создавать роли-папки с именем, введенным в поле *Folder role name*, не открывая форму создания новой роли. Новая роль-папка будет создана на самом верхнем уровне независимо от того, какая роль-папка выбрана слева в дереве;
-  удалять выбранные роли-папки.



Каждая роль может обладать неограниченным числом потомков, а также может быть потомком для неограниченного числа родителей.



Определить для роли роль-родитель нельзя, только наоборот. Именно поэтому при создании новой роли она всегда оказывается на верхнем уровне дерева.

Если создаваемая новая роль (назовем ее "А") должна быть потомком для какой-либо другой роли (например, "В"), следует сохранить роль "А", а затем открыть в редакторе роль "В" и привязать к ней в качестве потомка "А".

Древовидная организация справочника Ролей несет также важную функциональную нагрузку: роль-родитель наследует все права, которыми обладают ее роли-потомки в не зависимости от их уровня вложенности. Например, роли, определяющие функционал подчиненных, будучи потомками роли начальника отдела дают ему возможность совершать все доступные им операции.

- ▼ Руководитель отдела продаж
 - Менеджер по работе с корпоративными клиентами
 - Менеджер по работе с розницей
 - Менеджер по работе с VIP-клиентами
 - Менеджер по работе с оптовыми клиентами
 - Стажер

Таким образом, объекты прав могут быть либо в явном виде разрешены (allow) для роли, либо унаследованы ей от роли-потомка (derived).

Унаследованные от потомков права могут быть также разрешены для роли-родителя в явном виде, чтобы в случае исключения их из роли-потомка роль-родитель не лишилась требуемого права.

Может, однако, возникнуть такая ситуация, когда роль-потомок обладает некими правами, а роль-родитель этими правами обладать не должна. Например, роль кладовщика имеет право на осуществление отгрузки товара со склада. Одновременно она является потомком роли начальника склада. Но начальник склада не является материально-ответственным лицом и не имеет права

осуществлять такую отгрузку. Лишить роль-родитель начальника склада этого функционала можно отменив (revoke) соответствующее унаследованное им от роли-потомка разрешение.

Такое отмененное разрешение не будет унаследовано ролью-родителем. Например, выдавая права роли *Младшего подчиненного*, мы тем самым выдаем их и ролям-родителям – *Старшему подчиненному* и *Начальнику отдела*.

▼ Начальник отдела
▼ Старший подчиненный
Младший подчиненный

Отменив эти права для роли *Старшего подчиненного*, мы тем самым отменим их и для *Начальника отдела* (если только эти права не получены *Начальником отдела* также и из других дочерних ролей).



Если у роли роли-родителя есть две роли-потомка одного уровня, у одной из которых некое право выдано (allow), а у другой то же самое наоборот отменено (revoke), данное право ролью-родителем будет унаследовано. То есть разрешение доступа (allow) при наследовании приоритетнее запрета (revoke).

▼ Руководитель отдела продаж
Менеджер по работе с розницей
Менеджер по работе с оптом

Резюмируя вышесказанное, доступ к объекту прав в форме редактирования роли устанавливается с помощью флагов:

- Derived – доступ унаследован от роли-потомка. Снять этот флаг нельзя;
- Allow – разрешен. Этот флаг можно проставить в том числе и при уже установленном флаге Derived. Является взаимоисключающим с флагом Revoke;
- Revoke – отменен. Установка данного флага отменяет унаследованный доступ к объекту прав для текущей роли, а также для ее родителей. Соответственно его установка имеет смысл только при установленном флаге Derived. Является взаимоисключающим с флагом Allow.

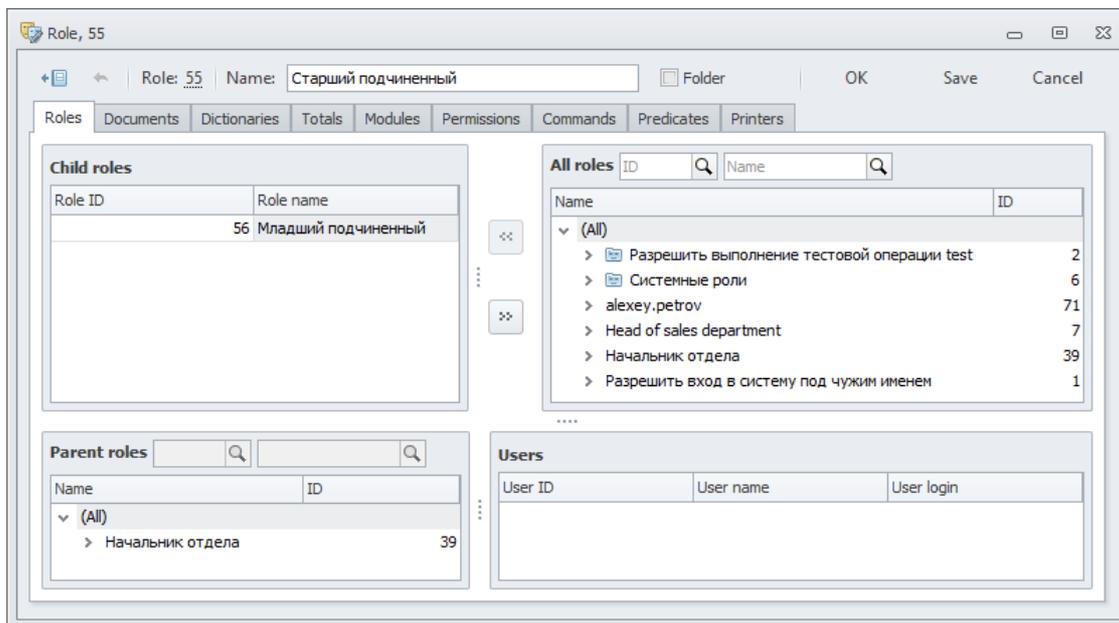
Кроме того, для ряда объектов прав доступ может выдаваться не на сам объект, а на выполнение над объектом операций четырех типов (CRUD):

- Read – чтение объектов и открытие списковой формы объектов данного типа;
- Create – создание нового объекта;
- Update – редактирование объекта;
- Delete – удаление объекта.

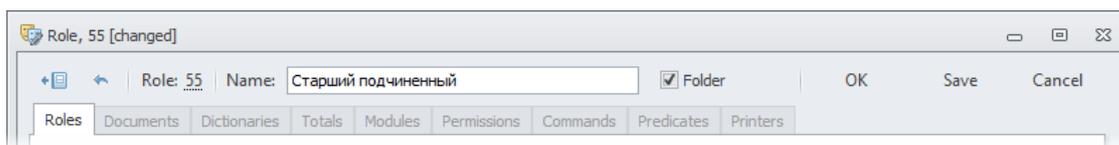
Разобравшись с механизмом наследования прав можно перейти к описанию формы редактирования ролей.

Форма редактирования ролей

Все свойства роли, кроме вынесенных в панель инструментов формы названия роли *Name* и флага *Folder*, определяющего что роль является папкой, сгруппированы по смысловым закладкам:



Для роли-папки доступна только закладка "Roles":



На закладке "Roles" (роли) можно изменить список ролей, являющихся дочерними для редактируемой роли, а так же посмотреть структуру родительских ролей и список пользователей, чьи права будут затронуты при изменении редактируемой роли:

- в области "Child roles" отображены роли-потомки редактируемой роли. Их можно удалить, выбрав и нажав кнопку ">";
- в области "All roles" выведены в виде дерева все существующие роли. Их можно добавить в качестве потомков к редактируемой, выбрав и нажав кнопку "<" (роли в дереве всегда отсортированы по алфавиту, а папки  всегда идут перед обычными ролями);
- в области "Parent roles" выведены в виде дерева все родительские роли, для которых редактируемая роль является потомком (роли в дереве всегда отсортированы по алфавиту, а папки  всегда идут перед обычными ролями);
- в области "Users" отображаются все пользователи, которым назначены редактируемая роль или роли из списка "Parent roles". Таким образом можно увидеть весь список пользователей на котором отразятся изменения редактируемой роли.

На закладке "Documents" устанавливается доступ к подтипам документов:

Roles Documents Dictionaries Totals Modules Permissions Commands Predicates Printers					
Doc type name <input type="text"/> <input type="button" value="Filter"/>					
<input type="checkbox"/> Derived <input type="checkbox"/> Allow <input type="checkbox"/> Revoke <input type="button" value="Sort"/>					
Doc Subtype ID	Operation Name	Derived	Allow	Revoke	
▼ Invoice					
▼ Order					
112	Read	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
112	Create	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
112	Update	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
112	Delete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
▼ Waybill					
▼ Invoice					
115	Read	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
115	Create	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
115	Update	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
115	Delete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Над документами каждого подтипа можно совершить четыре типа операций (CRUD), которые сгруппированы по этому подтипу. Подтипы в свою очередь сгруппированы по типу документов.

Закладка "Documents" имеет собственную панель инструментов, которая позволяет осуществлять:

- фильтрацию отображенных типов документов в соответствии с введенным в поле "Doc type name" текстом. Для фильтрации следует ввести значение в текстовое поле и нажать кнопку . Сбросить фильтр можно очисткой текстового поля и повторным поиском;
- фильтрацию по *разрешенным*, *унаследованным* и *отмененным* операциям над подтипами документов путем установки флагов "Derived", "Allow" и "Revoke" соответственно. При выборе одного или более флагов фильтра в панели инструментов, содержимое закладки будет отфильтровано таким образом, что отображенными останутся только операции с соответствующими проставленными флагами:

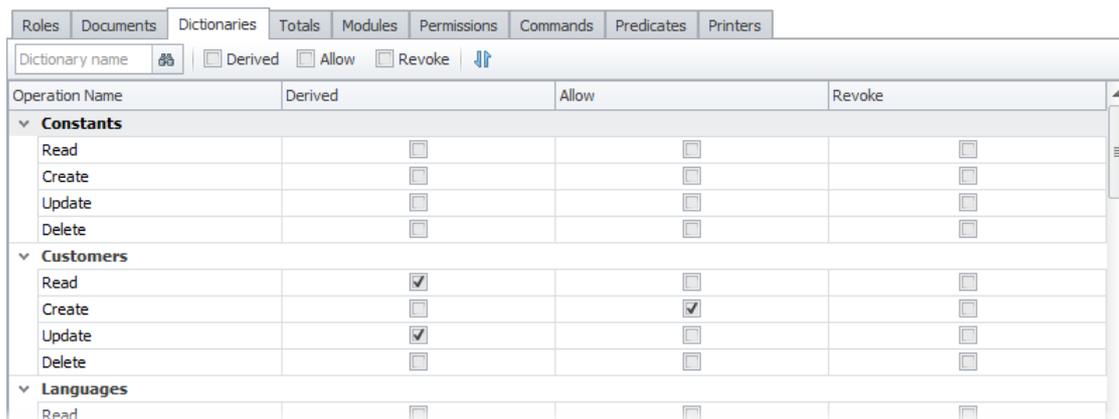
Roles Documents Dictionaries Totals Modules Permissions Commands Predicates Printers					
Doc type name <input type="text"/> <input type="button" value="Filter"/>					
<input type="checkbox"/> Derived <input checked="" type="checkbox"/> Allow <input type="checkbox"/> Revoke <input type="button" value="Sort"/>					
Doc Subtype ID	Operation Name	Derived	Allow	Revoke	
▼ Invoice					
▼ Order					
112	Create	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Два установленных флага фильтра работают будучи связанными оператором "И". То есть удовлетворяющими условиям фильтрации будут операции с обоими одновременно установленными флагами.

- дополнительную группировку содержимого по типу операции нажатием на кнопку . При этом сначала идут все операции типа Create, все также сгруппированные по типу и подтипу документа, затем Delete, Read и Update. Повторное нажатие кнопки  убирает дополнительную группировку:

Roles Documents Dictionaries Totals Modules Permissions Commands Predicates Printers					
Doc type name <input type="text"/> <input type="button" value="Filter"/>					
<input type="checkbox"/> Derived <input type="checkbox"/> Allow <input type="checkbox"/> Revoke <input type="button" value="Sort"/>					
Doc Subtype ID	Doc Subtype Name	Derived	Allow	Revoke	
▼ Operation Name: Create					
▼ Invoice					
112	Order	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
▼ Waybill					
116	Purchase order	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
115	Invoice	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
▼ Operation Name: Delete					
▼ Invoice					
112	Order	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
▼ Waybill					
116	Purchase order	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
115	Invoice	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

На закладке "Dictionaries" устанавливается доступ к справочникам:



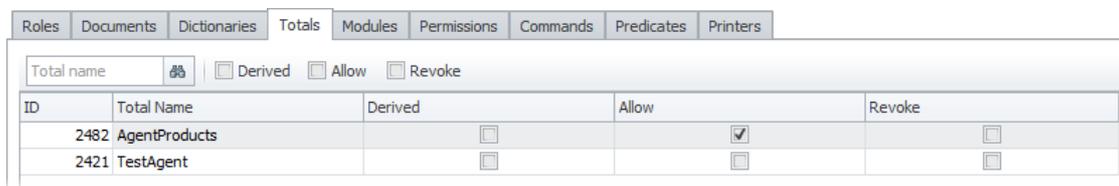
Operation Name	Derived	Allow	Revoke
Constants			
Read	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Create	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Update	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Delete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Customers			
Read	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Create	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Update	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Delete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Languages			
Read	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Для каждого справочника доступ выдается на совершение операций четырех типов (CRUD), которые сгруппированы по этому справочнику.

Закладка "Dictionaries" имеет собственную панель инструментов, которая позволяет осуществлять:

- фильтрацию отображенных справочников в соответствии с введенным в поле "Dictionary name" текстом;
- фильтрацию по *разрешенным*, *унаследованным* и *отмененным* операциям над справочниками путем установки флагов "Derived", "Allow" и "Revoke" соответственно (два установленных флага фильтра работают будучи связанными оператором "И");
- дополнительную группировку содержимого по типу операции нажатием на кнопку . Повторное нажатие кнопки  убирает дополнительную группировку.

На закладке "Totals" устанавливается доступ к итогам:



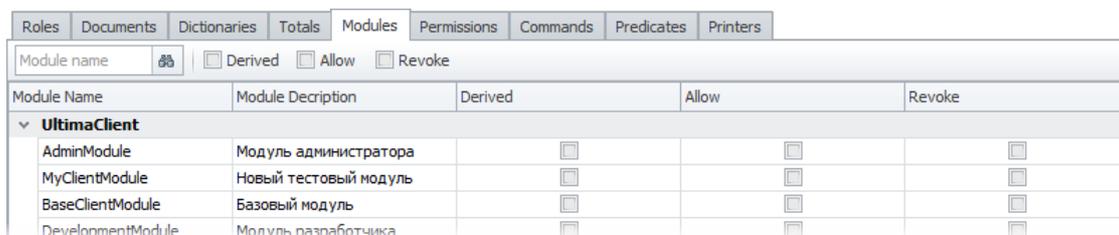
ID	Total Name	Derived	Allow	Revoke
2482	AgentProducts	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2421	TestAgent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Получение доступа к итогу позволяет пользователю строить отчеты по данному итогу, используя внутреннюю систему отчетов.

Закладка "Totals" имеет собственную панель инструментов, которая позволяет осуществлять:

- фильтрацию отображенных итогов в соответствии с введенным в поле "Total name" текстом;
- фильтрацию по *разрешенным*, *унаследованным* и *отмененным* итогам путем установки флагов "Derived", "Allow" и "Revoke" соответственно (два установленных флага фильтра работают будучи связанными оператором "И").

На закладке "Modules" устанавливается доступ к клиентским модулям:



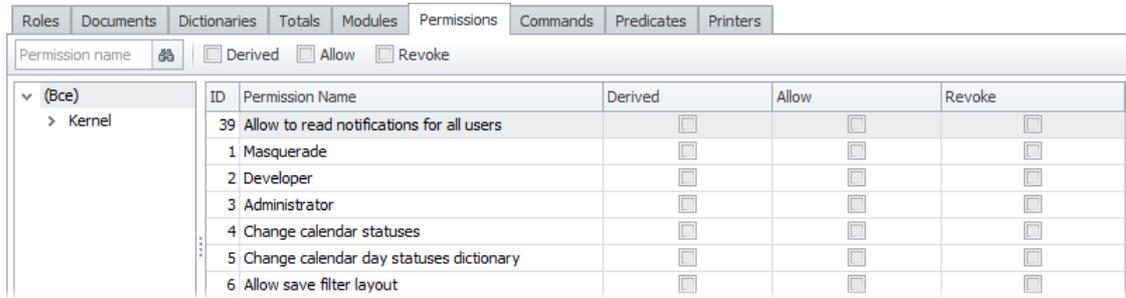
Module Name	Module Description	Derived	Allow	Revoke
UltimaClient				
AdminModule	Модуль администратора	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
MyClientModule	Новый тестовый модуль	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
BaseClientModule	Базовый модуль	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DevelopmentModule	Модуль разработчика	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Пользователь, чья роль обладает доступом к клиентскому модулю, сможет загружать и запускать его.

Закладка "Modules" имеет собственную панель инструментов, которая позволяет осуществлять:

- фильтрацию отображенных клиентских модулей в соответствии с введенным в поле "Module name" текстом;
- фильтрацию по *разрешенным*, *унаследованным* и *отмененным* клиентским модулям путем установки флагов "Derived", "Allow" и "Revoke" соответственно (два установленных флага фильтра работают будучи связанными оператором "И").

На закладке "Permissions" устанавливается доступ на сформулированные и проверяемые разработчиком права:



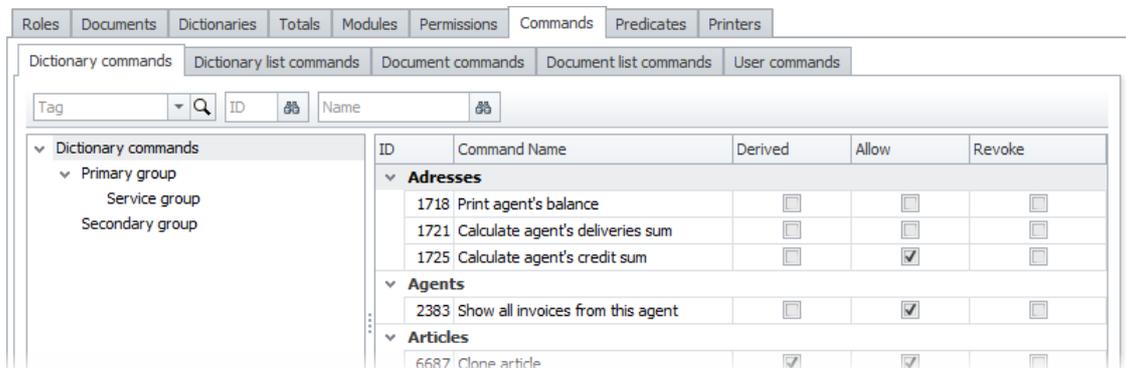
ID	Permission Name	Derived	Allow	Revoke
39	Allow to read notifications for all users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1	Masquerade	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	Developer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	Administrator	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	Change calendar statuses	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	Change calendar day statuses dictionary	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	Allow save filter layout	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Закладка прав разделена на две части: слева отображается дерево групп прав, справа – права выбранной слева группы (включая права всех дочерних групп).

Закладка "Permissions" имеет собственную панель инструментов, которая позволяет осуществлять:

- фильтрацию отображенных прав в соответствии с введенным в поле "Permission name" текстом;
- фильтрацию по *разрешенным*, *унаследованным* и *отмененным* правам путем установки флагов "Derived", "Allow" и "Revoke" соответственно (два установленных флага фильтра работают будучи связанными оператором "И").

На закладке "Commands" устанавливается доступ на выполнение команд:



ID	Command Name	Derived	Allow	Revoke
Addresses				
1718	Print agent's balance	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1721	Calculate agent's deliveries sum	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1725	Calculate agent's credit sum	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Agents				
2383	Show all invoices from this agent	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Articles				
6687	Clone article	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Все команды разбиты с помощью закладок на пять типов:

- Dictionary commands – команды, выполняемые над записью справочника;
- Dictionary list commands – команды, выполняемые над несколькими записями справочника;
- Document commands – команды, выполняемые над документом;
- Document list commands – команды, выполняемые над несколькими документами;
- User commands – пользовательские команды.

Каждая из закладок команд разделена по вертикали на две части: слева отображается дерево групп команд, справа – команды выбранной слева группы (включая команды всех дочерних групп).

В списке на закладке "Dictionary commands" и "Dictionary list commands" команды сгруппированы по справочникам, над записями которых они выполняются. В списке на закладке "Document commands" команды сгруппированы по подтипам документов, которые в свою очередь сгруппированы по типам документов. В списке на закладке "Document list commands" команды сгруппированы по типам документов.

Команды каждого типа можно фильтровать по тегу, идентификатору и названию, в соответствии с введенным в поля "Tags", "ID" и "Name" текстом. Для фильтрации следует ввести значение в текстовое поле и нажать кнопку . Сбросить фильтр можно очисткой текстового поля и повторным поиском.



Для выполнения команды у пользователя должны быть только права на ее запуск. При этом команда будет корректно выполнена, даже если она совершает действия недоступные пользователю в обычном режиме работы, например, добавляет запись в справочник к которому у пользователя вообще нет доступа.

На закладке "Predicates" перечислены предикаты, которые можно применить для данной роли:

Roles Documents Dictionaries Totals Modules Permissions Commands Predicates Printers				
Dictionary Document Total				
Dictionary name	Predicate name	<input type="checkbox"/> Derived	<input type="checkbox"/> Allow	<input type="checkbox"/> Revoke
Operation Name	Derived	Allow	Revoke	
Agents				
Predicate Name: Show only suppliers in dictionary 'Agents'				
Read	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Create	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Update	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Delete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Все предикаты разбиты с помощью закладок на три группы по объектам, к которым они применяются:

- Dictionary – справочники;
- Documents – документы;
- Totals – итоги.

Для *итогов* предикаты применяются непосредственно к объекту (в момент построения пользователем отчетов), для *справочников* и *документов* – к каждой операции (Read, Create, Update, Delete) в отдельности. Смысл применения предикатов к каждому типу операций в отдельности в том, что пользователю может быть необходимо ограничить не весь функционал действий над объектом, а лишь его часть. Например, пользователь должен видеть весь список записей справочника контрагентов, но создавать, править и удалять может лишь его часть – физических лиц. В этом случае предикат применяется для всех типов операций, кроме операции Read.



Предикаты не поддерживаются в версии системы для PostgreSQL, поэтому закладка *Predicates* в этой версии недоступна.

В списке на закладке "Dictionary" предикаты сгруппированы по справочникам, которым они назначены, в списке на закладке "Documents" по типам документов.

Поскольку предикат ограничивает доступ к объекту, роль, в которой применяется предикат, должна также иметь и доступ к этому объекту. Например, если пользователю необходимо предоставить ограниченный доступ к справочнику контрагентов, следует для его роли:

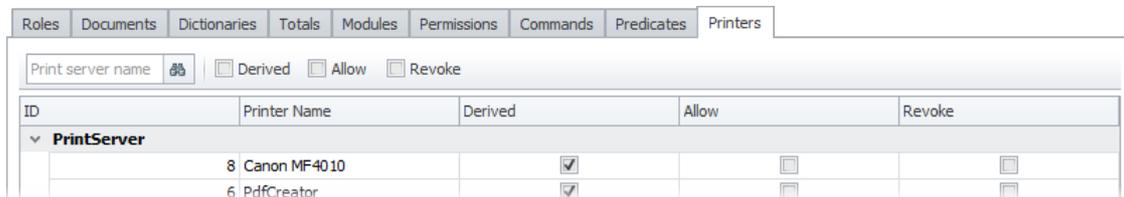
- дать доступ к справочнику контрагентов;
- применить предикат, ограничивающий этот доступ.

При этом операция, на которую накладываются ограничения предиката, также должна быть разрешена. Например, применение предиката, накладывающего ограничения на операцию удаления записей справочника контрагентов, для роли, которой предоставлен доступ только на чтение, добавление и изменение записей этого справочника, будет бессмысленной, так как в этом случае имеет место попытка ограничения доступ на операцию, которая и так не разрешена.

Каждая из подзакладок по типу предиката "Dictionary", "Documents" и "Totals" на закладке "Predicates" имеет собственную панель инструментов, которая позволяет осуществлять:

- фильтрацию отображенных предикатов по типу объекта, к которому они применяются, в соответствии с введенным в поле "Dictionary name" (для справочников, "Doc type name" для типов документов, "Total name" для итогов) текстом;
- фильтрацию отображенных предикатов в соответствии с введенным в поле "Predicate name" текстом;
- фильтрацию по *разрешенным*, *унаследованным* и *отмененным* предикатам путем установки флагов "Derived", "Allow" и "Revoke" соответственно (два установленных флага фильтра работают будучи связанными оператором "И").

На закладке "Printers" устанавливается доступ к принтерам:



Доступные пользователю принтеры отображаются в стандартной форме печати системы Ultimate Solid, недоступные же не видны.

В списке на закладке "Printers" принтеры сгруппированы по серверам печати.

Закладка "Printers" имеет собственную панель инструментов, которая позволяет осуществлять:

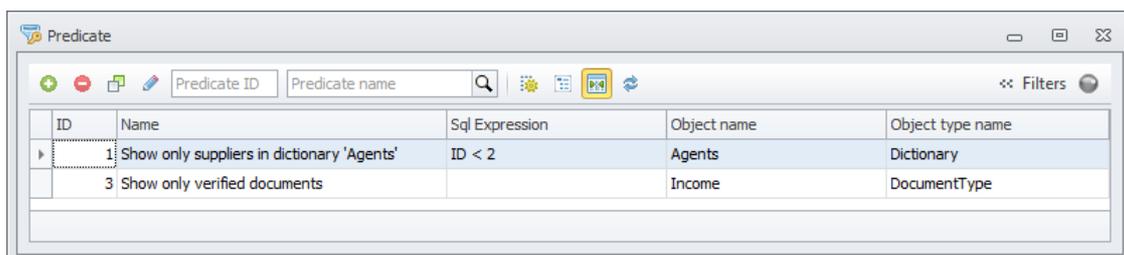
- фильтрацию отображенных принтеров в соответствии с введенным в поле "Printer name" текстом;
- фильтрацию по *разрешенным*, *унаследованным* и *отмененным* принтерам путем установки флагов "Derived", "Allow" и "Revoke" соответственно (два установленных флага фильтра работают будучи связанными оператором "И").

Предикаты



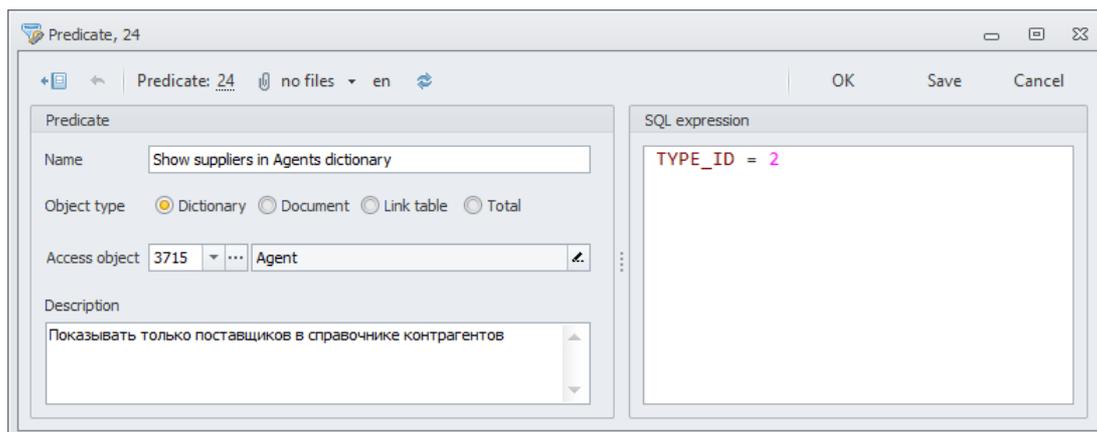
Предикаты это объекты прав, которые позволяют ограничить доступ пользователя на уровне отдельных строк.

Предикаты организованы в справочнике Predicates:



Записи справочника можно фильтровать по *Названию предиката (Predicate name)*.

Для нового предиката необходимо указать:



- *Name* – название предиката;
- *Access object* тип объекта – *Dictionary*, *Document* или *Total* (справочник, документ или итог) – и непосредственно сам объект, к которому применяется предикат;
- *SQL expression* – собственно выражение на языке SQL. Это выражение будет подставлено при выполнении любого обращения к соответствующей таблице;
- *Description* – описание действия предиката.

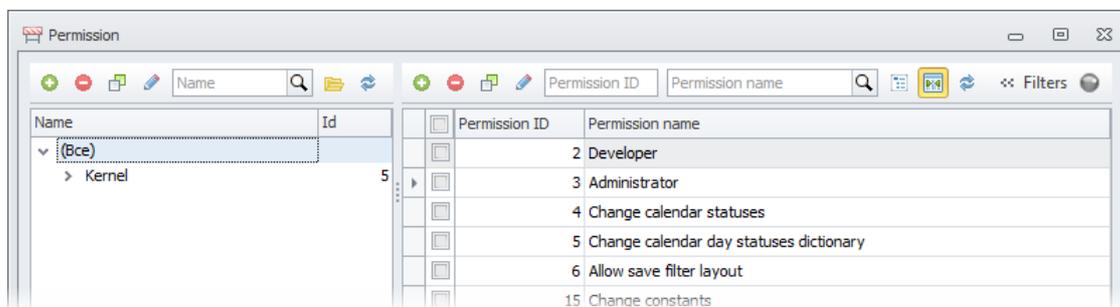


Предикаты не поддерживаются в версии системы для PostgreSQL.

Права



Права, сформулированные и проверяемые разработчиком на уровне исполняемого кода, организованы в справочнике Permissions:

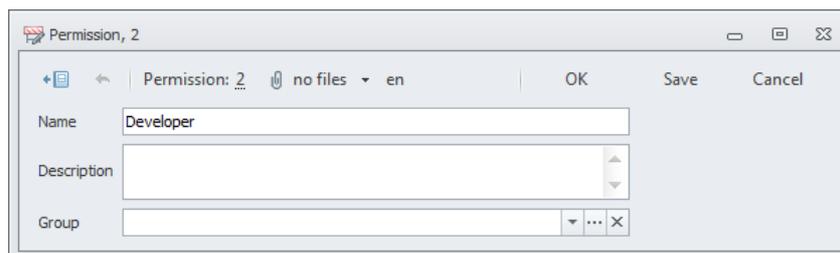


Permission ID	Permission name
2	Developer
3	Administrator
4	Change calendar statuses
5	Change calendar day statuses dictionary
6	Allow save filter layout
15	Change constants

Справочник прав разделен на две части: слева отображается дерево групп прав, справа – список прав выбранной слева группы.

Записи справочника можно фильтровать по *Названию права (Permission name)*.

Для нового права необходимо указать:



- *Name* – название права;

- *Description* – описание действия/функционала, предоставляемого правом;
- *Group* – группу, к которой относится право.

Добавление записей в справочник прав имеет смысл только в том случае, если разработчик также напишет соответствующий код, проверяющий эти права.

Права доступа к системным таблицам БД

База данных системы Ultimate Solid состоит из двух схем: *схемы ядра* и *прикладной схемы*. Описывающие бизнес-логику и бизнес-объекты системы метаданные хранятся в *схеме ядра*. В *прикладной схеме* на основе этих метаданных создаются таблицы и хранятся данные прикладной области (справочники, развязочные таблицы, табличные части документов, итоги и т.д.).

Если доступ к таблицам *прикладной схемы* базы данных определяется выдачей разрешений на выполнение операций CRUD (Create, Read, Update, Delete) непосредственно над объектами (справочниками, подтипами документов и т.п.), то для таблиц *схемы ядра* БД – системных таблиц – этот доступ определяется выдачей соответствующих *прав (Permissions)*. При этом к ряду служебных таблиц *схемы ядра* доступ закрыт для всех пользователей.

Доступ к системным таблицам подчиняется следующим правилам.

1. Любой пользователь независимо от его роли и наличия каких-либо *прав* обладает возможностью чтения (выполнения операции Read) данных из всех таблиц.

Также любой пользователь может выполнять все операции (CRUD) над следующими таблицами:

- ATTACHMENTS – вложения, прикрепляемые к объектам метаданных;
- ATTACHMENT_TYPES – типы вложений;
- NOTIFICATIONS – уведомления;
- PARENT_DOCUMENTS – отношения родитель-потомок между документами;
- SPELL_CHK_WORDS – словарь, дополняемый пользователями при проверке орфографии.

2. Пользователь, обладающий *правом Administrator*, имеет возможность (помимо чтения всех таблиц) совершать следующие дополнительные операции над следующими таблицами:

- операции CREATE, UPDATE, DELETE:

- APP_CLUSTERS;
- APP_SERVERS;
- APP_SERVER_TASKS;
- CLUSTER_CONFIGS;
- PREDICATES;
- PRINTER_PERMISSIONS;
- PRINTERS;
- PRINT_SERVERS;
- ROLE_MODULES;
- ROLE_PERMISSIONS;
- ROLE_PREDICATES;
- ROLES;
- ROLE_TREE;
- SCRIPT_PERMISSIONS;
- STANDBY_SERVERS;
- ST_DRV_TO_PRINTERS;
- TOTAL_PERMISSIONS;
- UI_TEMPLATES;
- USER_GROUPS;
- USERS;
- VERSION_TAGS;

- операции UPDATE, DELETE:
 - PRINT_PACK_QUEUE;
 - PRINT_QUEUE;
 - PRINT_QUEUE_LOCKS;
 - PRINT_TASK_QUEUE;
 - операцию DELETE:
 - USER_SETTINGS;
 - операцию UPDATE:
 - TASKS (только для свойства EXEC TIME).
3. Пользователь, обладающий *правом* **Developer**, имеет возможность выполнять все операции (CRUD) над всеми таблицами, кроме следующих:
- только операции READ, UPDATE, DELETE:
 - PRINT_PACK_QUEUE;
 - PRINT_QUEUE;
 - PRINT_QUEUE_LOCKS;
 - PRINT_TASK_QUEUE;
 - только операции READ, DELETE:
 - USER_SETTINGS;
 - только операцию READ:
 - GENDERS;
 - LAYOUT_TYPES;
 - OBJECT_TYPES;
 - PARAM_MODES;
 - PRINT_PACKAGE_STATES;
 - PRINT_TASK_STATES;
 - PROPERTY_TYPES;
 - SCRIPT_TYPES;
 - SPELL_CHK_DICT_TYPES.
4. Пользователь, обладающий *правом* на управление календарем **Calendar management**, имеет возможность совершать все операции (CRUD) над следующими таблицами:
- CALENDAR_DAY_STATUSES;
 - CALENDAR_STATUSES.
5. Пользователь, обладающий *правом* на изменение констант **Edit constants**, имеет возможность совершать все операции (CRUD) над следующими таблицами:
- CONSTANT_GROUPS;
 - CONSTANTS.
6. Пользователь, обладающий *правом* на локализацию исключений **Edit exception translations**, имеет возможность совершать операции CRU над следующими таблицами:
- EXCEPTION_TRANSLATIONS;
 - EXCEPTION_TRANSL_TEXTS.
7. Пользователь, обладающий *правом* на перевод **Edit metadata translations**, имеет возможность совершать операции CRU над таблицей METADATA_TRANSLATIONS и все операции (CRUD) над таблицей SPELL_CHK_DICTIONARIES.
8. Пользователь, обладающий *правом* на создание документации **Edit metadata user comments**, имеет возможность совершать операции CRU над таблицей METADATA_USER_COMMENTSS.
9. Пользователь, обладающий *правом* на управление очередью печати **Manage print queues**, имеет возможность совершать операции CRU над следующими таблицами:
- PRINT_PACK_QUEUE;

- PRINT_QUEUE;
- PRINT_QUEUE_LOCKS;
- PRINT_TASK_QUEUE.

10. Пользователь, обладающий *правом* на редактирование интерфейса **Edit UI templates**, имеет возможность совершать все операции (CRUD) над таблицей UI_TEMPLATES.

Группа инструментов конфигурации системы

Инструменты группы помимо справочников π констант (*Constants*),  кластеров (*Clusters*) и  тегов версий (*Versions tags*) содержат следующий функционал:

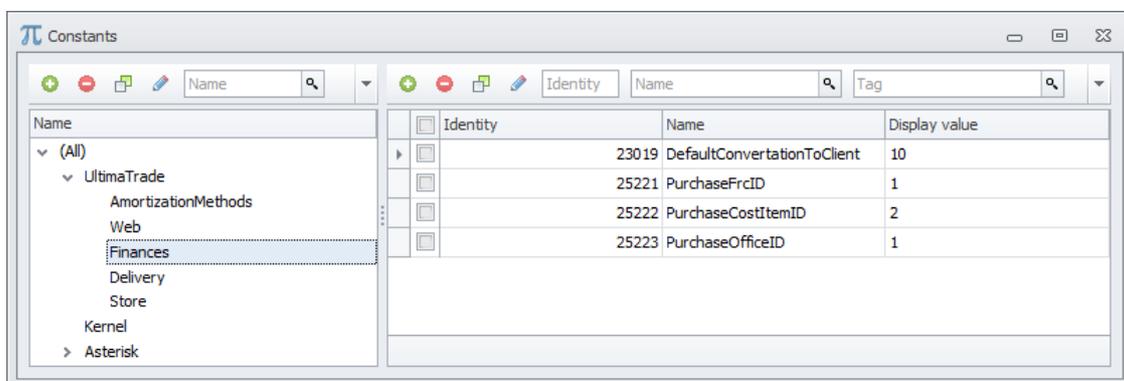


- *Current config* – конфигурация кластера, с которым работает запущенный клиент. По нажатию на кнопку откроется [форма редактирования конфигурации кластера](#);
- *Version tag* – тег версии, с которой запущен сервер приложений, с которым работает клиентское приложение. В приведенном примере это тег *temp*;
- *cluster* – имя кластера, в который входит сервер приложений, с которым работает клиентское приложение. В приведенном примере это кластер *Test*.

Константы

π Константы предназначены для хранения значений переменных, используемых в программном коде, которые со временем могут меняться. Например, это может быть дата закрытия отчетного периода, текст e-mail рассылки и т.п.

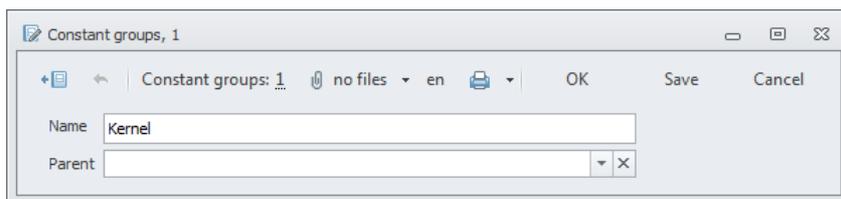
Посмотреть существующие и создать новые константы можно в справочнике Constants:



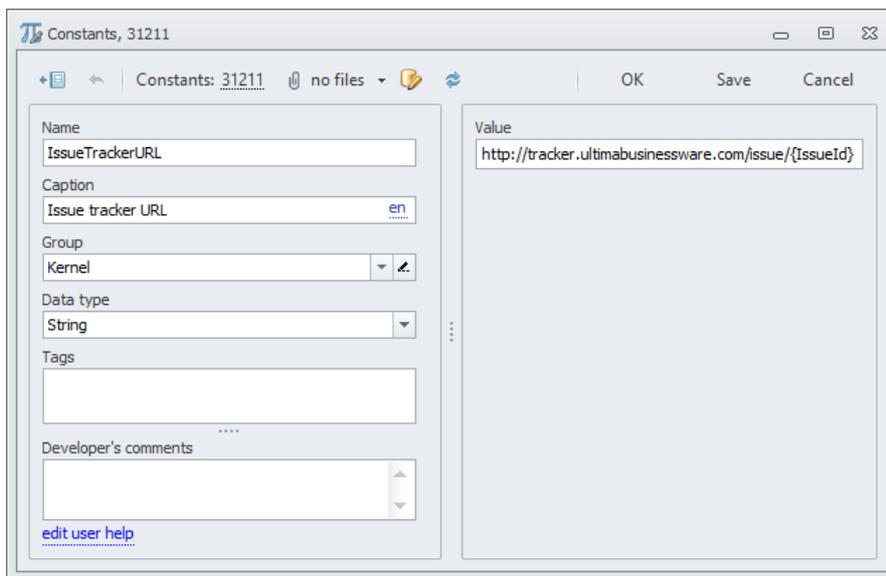
Справочник констант разделен на две части: слева отображается дерево групп констант, справа – список констант выбранной слева группы.

Группы констант можно фильтровать по *Названию* группы (*Name*), а константы по *Названию* константы (*Name*) и *Тегам* (*Tag*).

Форма редактирования группы констант позволяет задать ее *Название (Name)* и выбрать *Родителя* в дереве (*Parent*). Если родитель не выбран, группа будет находиться на верхнем уровне дерева:



Форма редактирования константы позволяет задать следующие свойства:



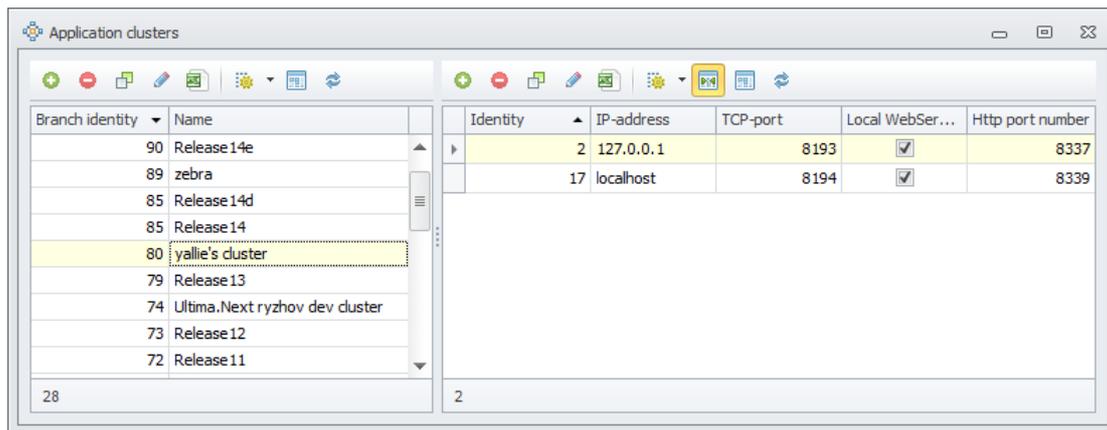
- *Name* – имя константы, используемое в скриптах;
- *Caption* — локализуемое название константы;
- *Group* – группа, которой принадлежит константа. При создании новой константы в качестве значения свойства выбирается отмеченная в справочнике *Constants* группа;
- *Data type* – тип константы. Константы в зависимости от типа могут принимать следующие значения:
 - *Integer number* – целое число, 64-разряда;
 - *Decimal number* – дробное число, 96-разрядов;
 - *Text* – текст размером не более 2Kb;
 - *String* – строка размером не более 2Kb;
 - *HTML text* – текст неограниченного размера;
 - *Date with time* – дата и время (транслируется по часовым поясам);
 - *Date* – дата (без времени);
 - *Flag* – используется для логических значений;
 - *Dictionary record ID* – запись справочника;
 - *Records id list* – список записей справочника;
- *Tags* – теги, используемые для описания функционала константы. Используются для поиска объектов, реализованных под определенным функционал, ассоциированный с этим тегом. Добавление тега осуществляется клавишами **Space** или **Enter**. Удаление – кнопкой  после тега. Поскольку пробел используется для ввода тега, можно заменить его символами "_" или "-" в тегах с названием из нескольких слов;
- *Developer's comments* – комментарии прикладного разработчика;

- *edit user help* – комментарий к объекту, который конечный пользователь может видеть в виде подсказки (hint), выпадающей при наведении курсора мыши на объект. Комментарий вводится для каждого языка системы;
- *Value* – значение константы.

Кластеры и сервера

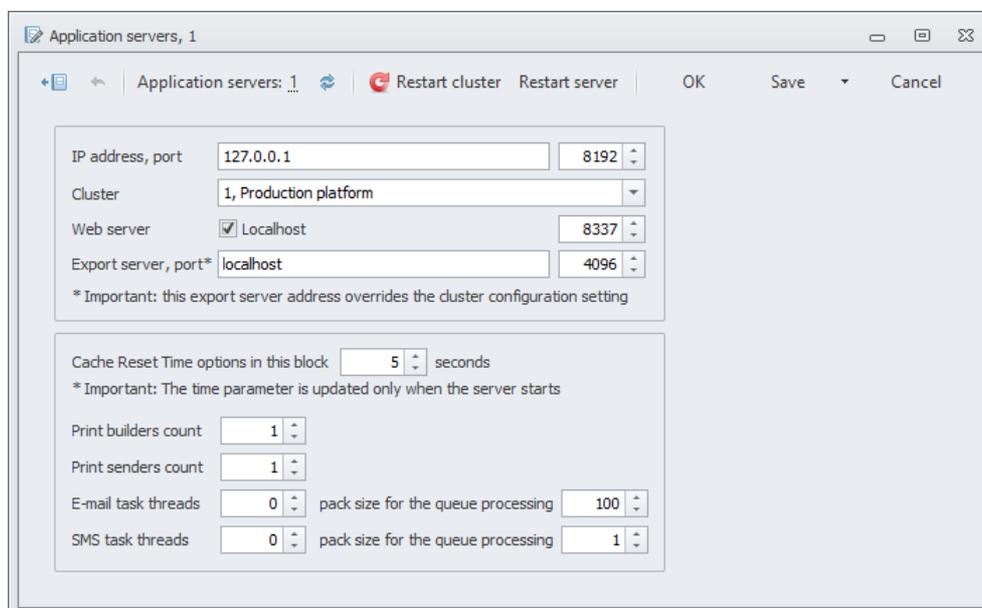


Управление составом и конфигурацией кластеров осуществляется через форму «Application clusters»:



Форма справочника разделена на две части: слева отображается список кластеров, справа — серверов. Выбор кластера слева фильтрует список серверов справа, оставляя в списке только принадлежащие ему. Текущий кластер и текущий сервер приложений подсвечены в списке бледно-желтым цветом.

Форма редактирования конфигурации сервера приложений позволяет задать следующие свойства:



- *IP address, port* — адрес и порт сервера приложений;
- *Cluster* — выбрать кластер, в который будет входить сервер приложений;
- *Export server* — адрес и порт сервера экспорта, с которым сервер приложений будет взаимодействовать. Если оставить поле пустым, адрес сервера экспорта будет взят из настроек кластера;

- *Web server* — адрес и порт локального web сервера. С установленным флагом *Localhost* web сервер будет запущен локально. Если же web сервер был развернут по конкретному адресу и доступен для всех, флаг *Localhost* должен быть снят. Web сервер предоставляет REST/SOAP сервисы, которые, как правило, используются сайтом компании или сторонними приложениями для интеграции. Сервер должен быть настроен таким образом, чтобы соответствующие приложения (сайты и т.д.) могли установить соединение к нему.



Если на одном компьютере запущены два сервера приложений, и у обоих в настройках включены web сервера, следует убедиться, что у web серверов заданы разные порты во избежание конфликта.

- *Cache reset time* — время жизни кеша загруженных настроек сервера приложений;
- *Print builders count* — доступное количество потоков для обработки [отложенных \(асинхронных\) заданий печати](#). Если значение *Print builders count* будет равно 0, сервер приложений не будет обрабатывать отложенные задания печати;
- *Print senders count* — доступное количество потоков для отправки заданий печати на сервер печати. Если значение *Print senders count* будет равно 0, сервер приложений не будет отправлять задания на печать. Не рекомендуется, чтобы общее число дополнительных потоков обработки и отправки заданий печати превышало число виртуальных ядер компьютера, на котором работает сервер приложений.
- *E-mail task threads* — количество потоков для обработки очереди рассылки сообщений электронной почты.
- *SMS task threads* — количество потоков для обработки очереди рассылки СМС-сообщений.
- *Pack size for the queue processing* — размеры пакетов для обработки очередей рассылки. Размер пакета определяет, сколько сообщений будут загружены для обработки одновременно. Чем меньше размер пакета — тем надежнее, но медленнее рассылка. При ошибках рассылки сервер будет повторно рассылать пакет сообщений, поэтому часть сообщений может быть отправлена повторно. Чтобы исключить эту ситуацию, установите размер пакета в 1.

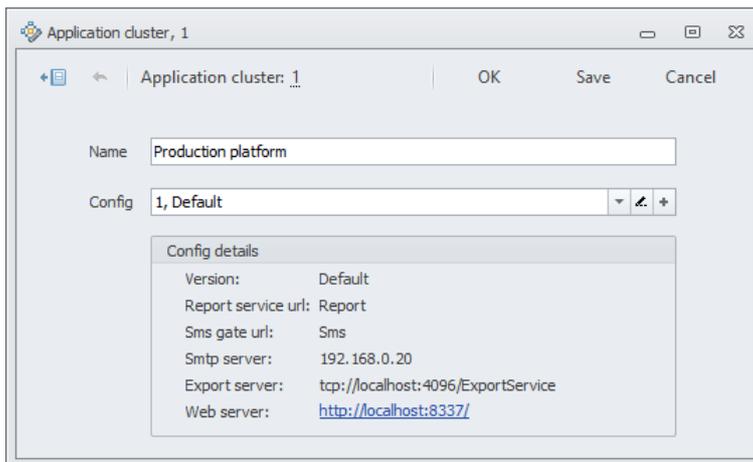


Хотя бы один из серверов приложений в кластере должен иметь хотя бы один поток для обработки отложенных заданий печати, чтобы такие отложенные задания могли быть обработаны, и/или один поток для отправки заданий на печать, чтобы задания могли быть отправлены на сервер печати. Аналогичным образом, в кластере должен быть по крайней мере один сервер со включенными задачами обработки очередей электронной почты и СМС.

В панели инструментов сервера есть две дополнительные команды:

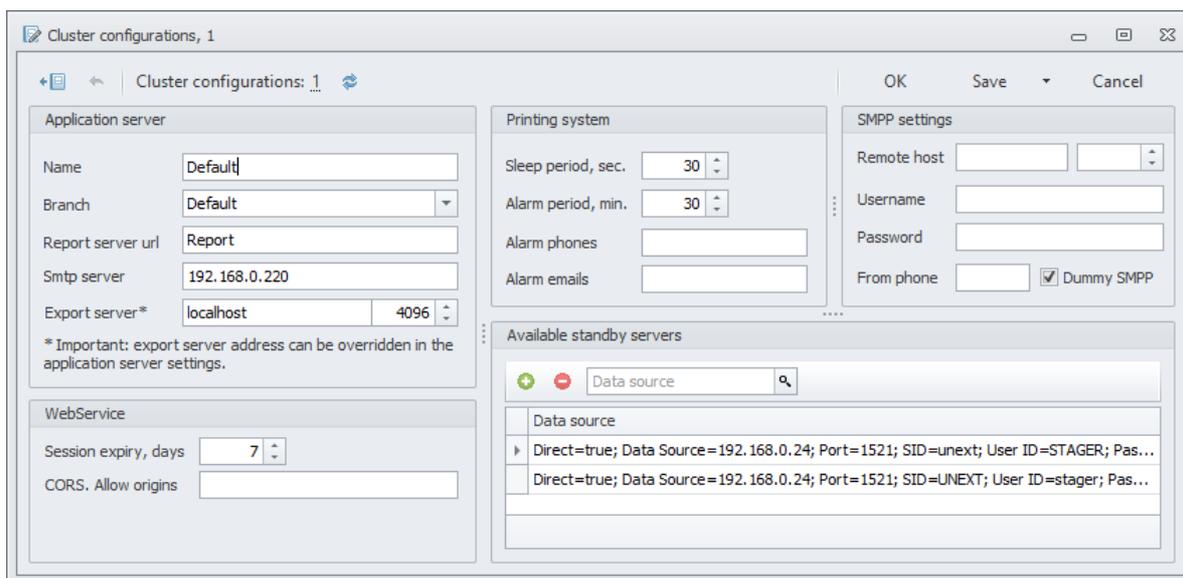
- *Restart cluster* — перезапуск кластера, к которому относится выбранный сервер приложений. Для выполнения этой команды нужно, чтобы в конфиг-файле указанного сервера был включен веб-сервис для перезапуска сервера.
- *Restart server* — перезапуск выбранного сервера приложений.

Форма редактирования кластера позволяет задать следующие свойства кластера:



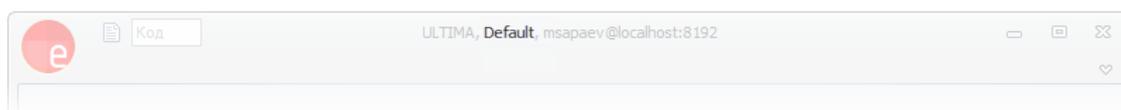
- *Name* — имя кластера;
- *Cluster* — выбрать конфигурацию кластера. Под конфигурацией выбранного кластера расположено ее краткое описание — *Config details*. Элемент управления выбора конфигурации позволяет:
 - ▼ — выбрать конфигурацию из существующих;
 - ✎ — отредактировать выбранную конфигурацию;
 - ⊕ — создать новую конфигурацию.

В форме редактирования конфигурации кластера свойства разбиты на группы:



Application server — настройки системы печати:

- *Name* — название конфигурации. Данное название отображается в заголовке главной формы приложения:



- *Branch* — ветка конфигурации, с которой будут работать входящие в кластер сервера приложений;



Изменение конфигурации кластера применяется входящими в него серверами приложений на лету.

То есть после выбора новой ветки в конфигурации кластера и ее сохранения, очередное обращение клиентского приложения к серверу приложений, входящему в этот кластер, будет обращением уже к новой ветке метаданных. Редактируемый, например, документ или запись справочника, открытые в старой версии, будут сохранены уже в новой. В ряде случаев это может привести к разного рода коллизиям и ошибкам.

Настоятельно рекомендуется проводить такие изменения при остановленном сервере приложений (или, как минимум, в период наименьшей активности пользователей).

- *Report server url* — сервер отчетов;
- *Smtп server* — адрес почтового сервера (порт стандартный — 25);
- *Export server* — адрес и порт сервера экспорта. Если в настройках сервера приложений (см. выше) указан другой адрес сервера экспорта, то приоритет будет у адреса, указанного в сервере приложений.

Print system — настройки системы печати:

- *Sleep period, sec.* — период бездействия потоков печати на сервере приложений в секундах. Если при очередном обращении к базе данных сервер приложений не получил задание печати на обработку или отправку, он будет ожидать заданный промежуток времени перед следующим обращением. Также в случае неудачной отправки задания на сервер печати, например, по причине его недоступности, сервер приложений будет ожидать заданный промежуток времени перед повторной попыткой отправки задания;
- *Alarm period, min.* — периодичность отправки тревожных уведомлений о проблемах серверов печати или принтеров в минутах. При ошибках во время отправки задания на сервера печати или попытке непосредственно печати администраторам будет выслано уведомление. Рассылка уведомлений будет осуществляться не каждый раз при возникновении проблемы, а кумулятивно с периодичностью не чаще заданной;
- *Alarm phones* — список телефонов администраторов для рассылки тревожных уведомлений посредством SMS;
- *Alarm emails* — список почтовых адресов администраторов для рассылки тревожных уведомлений посредством электронных писем.

SMPP settings — настройки SMS портала:

- *Remote host* — адрес и порт портала;
- *Username* — логин для подключения к portalу;
- *Password* — пароль для подключения к portalу;
- *From phone* — номер, который будет использоваться как номер отправителя в сообщениях;
- *Dumpy SMPP* — заглушка для проверки отправки SMS в обработчиках. С установленным флагом текст SMS вместо отправки на портал пишется в лог сервера.

WebService — настройки web сервисов:

- *Session expiry, days* — время жизни сессии web сервиса в днях.
- *CORS. Allow origins* — список доверенных доменов для поддержки запроса совместно используемых ресурсов (CORS). Доверенные домены перечисляются через запятую.

Available standby servers — список резервных серверов Oracle, доступных для серверов кластера.

Теги версий



В системе Ultimate Solid реализован механизм версионирования метаданных, который удобно представить в виде дерева.

Каждая версия метаданных может быть помечена тегом, которые делятся на:

- *branch-теги* – используются для пометки последнего узла ветви в дереве версий. Каждая ветвь версий (ее последний узел) обязательно отмечена своим уникальным branch-тегом. Обычно практика такова, что каждый прикладной разработчик имеет свой branch-тег, которым он помечает свою ветвь разработки.

Одна версия метаданных не может быть помечена более чем одним branch-тегом.

Изначально в системе заведен единственный branch-тег *Default*, которым помечена основная ветвь дерева версий метаданных;

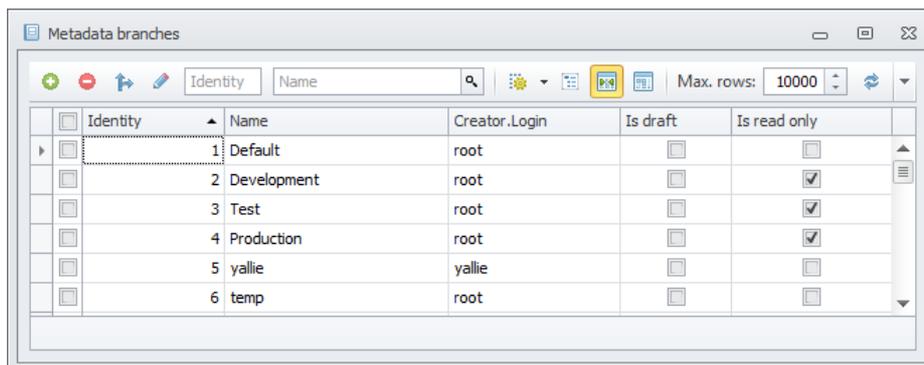
- *обычные теги* – ими может быть помечена любая (кроме последней), версия метаданных в ветви. Причем, одна версия может быть помечена более чем одним обычным тегом.

Любым тегом может быть помечена не более чем одна версия метаданных. При отметке тегом другой версии, он автоматически снимается с предыдущей.

Помимо branch-тега *Default* в системе изначально заведены еще два (обычных) тега:

- *Production* – тег, используемый для пометки последней стабильной версии метаданных, предназначенной для работы рядовых пользователей системы;
- *Debug* – тег, используемый для пометки версии метаданных, предназначенной для тестирования, например, выборочно несколькими пользователями или отделом тестирования.

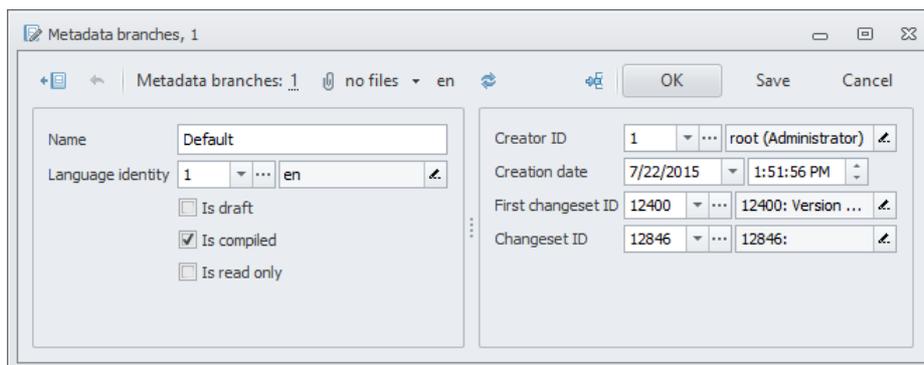
Посмотреть существующие и создать новые теги, которыми помечаются версии метаданных, можно в справочнике *Version tags*:



Identity	Name	Creator.Login	Is draft	Is read only
1	Default	root	<input type="checkbox"/>	<input type="checkbox"/>
2	Development	root	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	Test	root	<input type="checkbox"/>	<input checked="" type="checkbox"/>
4	Production	root	<input type="checkbox"/>	<input checked="" type="checkbox"/>
5	yallie	yallie	<input type="checkbox"/>	<input type="checkbox"/>
6	temp	root	<input type="checkbox"/>	<input type="checkbox"/>

Записи справочника можно фильтровать по *Названию* тега (*Name*).

Теги обладают следующими свойствами:



Metadata branches, 1

Name: Default

Language identity: 1 en

Is draft

Is compiled

Is read only

Creator ID: 1 root (Administrator)

Creation date: 7/22/2015 1:51:56 PM

First changeset ID: 12400 12400: Version ...

Changeset ID: 12846 12846:

- *Name* – имя тега;
- *Version* – версия метаданных, помеченная данным тегом:
 - для branch-тега изменить версию нельзя;
 - для обычного тега нельзя выбирать незафиксированную версию метаданных;
 - также для обычного тега нельзя выбирать версию, метаданные которой не были скомпилированы.



Чтобы изменение версии для тега, указанного в конфигурации кластера, начало действовать, необходимо:

- перезапустить сервера приложений, входящие в этот кластер;
- опционально, если после перезапуска серверов приложений возникают ошибки, перезапустить также клиентские приложения.

По этой причине настоятельно рекомендуется проводить изменения версии для тега *Production* в период наименьшей активности пользователей.

- *Owner user* – пользователь-владелец тега. Необязательное свойство, носящее информационный характер и предназначенное для указания прикладного разработчика, который ведет разработку под этим branch-тегом;
- *Draft branch* – изменения с ветки метаданных, помеченной branch-тегом с этим флагом, нельзя протолкнуть на ветку *Default*.

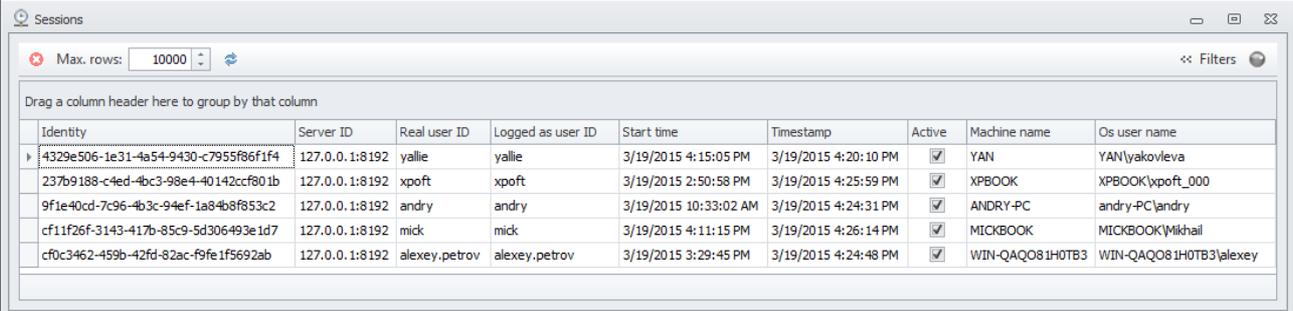
В справочнике тегов можно создать только обычные теги. Для branch-тегов можно лишь изменить название и владельца.

Группа инструментов Monitor

Сессии

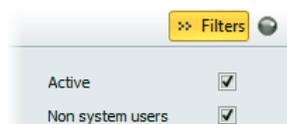


Список всех подключений к серверам приложений, можно найти в справочнике "Sessions":



Identity	Server ID	Real user ID	Logged as user ID	Start time	Timestamp	Active	Machine name	Os user name
4329e506-1e31-4a54-9430-c7955f86f1f4	127.0.0.1:8192	yalie	yalie	3/19/2015 4:15:05 PM	3/19/2015 4:20:10 PM	<input checked="" type="checkbox"/>	YAN	YAN\yakovleva
237b9188-c4ed-4bc3-98e4-40142ccf801b	127.0.0.1:8192	xpoft	xpoft	3/19/2015 2:50:58 PM	3/19/2015 4:25:59 PM	<input checked="" type="checkbox"/>	XPBOOK	XPBOOK\xpoft_000
9f1e40cd-7c96-4b3c-94ef-1a84b8f853c2	127.0.0.1:8192	andry	andry	3/19/2015 10:33:02 AM	3/19/2015 4:24:31 PM	<input checked="" type="checkbox"/>	ANDRY-PC	andry-PC\andry
cf11f26f-3143-417b-85c9-5d306493e1d7	127.0.0.1:8192	mick	mick	3/19/2015 4:11:15 PM	3/19/2015 4:26:14 PM	<input checked="" type="checkbox"/>	MICKBOOK	MICKBOOK\Mikhail
cf0c3462-459b-42fd-82ac-f9fe1f5692ab	127.0.0.1:8192	alexey.petrov	alexey.petrov	3/19/2015 3:29:45 PM	3/19/2015 4:24:48 PM	<input checked="" type="checkbox"/>	WIN-QAQO81H0TB3	WIN-QAQO81H0TB3\alexey

Изначально справочник сессий открывается с включенным фильтром, настройки которого ограничивают отображаемый список только активными сессиями несистемных пользователей.



Каждая сессия обладает следующими свойствами:

- *Identity* – идентификатор сессии;
- *Server* – сервер приложений, открывший сессию;
- *Real user* – логин пользователя, под чьим паролем было осуществлено подключение к серверу приложений;
- *Logged as user* – логин пользователя, под которым было осуществлено подключение к серверу приложений (если пользователь вошел в клиентское приложение под логином другого пользователя, используя для этого свои логин/пароль).

Например, администратор *admin* входит в клиентское приложение со своими учетными данными под логином пользователя *user*, чтобы проверить настройки его роли. В открытой при этом сессии в свойстве *User who logged in using his own password* будет логин *admin*, а в свойстве *User who logged in* – логин *user*. В случае же, когда администратор входит под собой, в обоих свойствах будет логин *admin*;

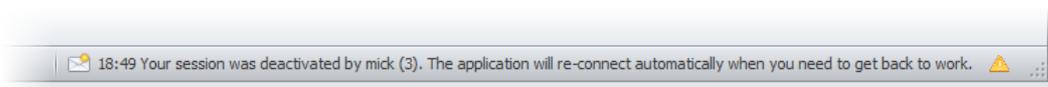
- *Start time* – время открытия сессии;
- *Timestamp* – время последнего обращения к БД;

- *Active* – флаг активности сессии. Сессии без флага *Active* – закрыты;
- *Machine name* – имя компьютера пользователя, с которого была открыта сессия;
- *OS user name* – имя пользователя в операционной системе на компьютере, под которым была открыта сессия.

По нажатию на кнопку  в панели инструментов выбранную сессию можно деактивировать (снять флаг *Active*). После этого сессия перестает учитываться как активная. Это помогает в двух типичных ситуациях:

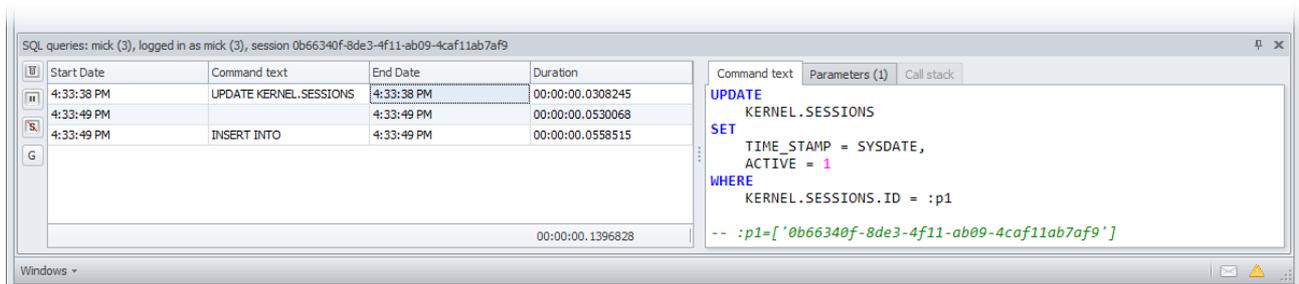
- приложение было завершено некорректно и сессия осталась активной;
- пользователь запустил несколько экземпляров программы и отошел от компьютера на длительное время.

Если деактивировать сессию живого пользователя, ему будет отправлено сообщение о том, что текущая сессия была приостановлена администратором. При первой необходимости программа сможет автоматически восстановить соединение и возобновить работу, как будто ничего не произошло:



Некорректное закрытие клиентского приложения может привести к тому, что флаг активности сессии (*Active*) не будет снят. В этом случае определить закрытые сессии можно по времени последнего обращения к серверу приложений (*Last access time*). Клиентское приложение сообщает о своем присутствии (активности) серверу приложений каждые пять минут даже при отсутствии активности пользователя. Таким образом, все сессии с существенно большим временем последнего обращения можно считать закрытыми.

По двойному клику левой кнопкой мыши по сессии в списке для нее открывается трассировщик SQL запросов:



Просмотр логов



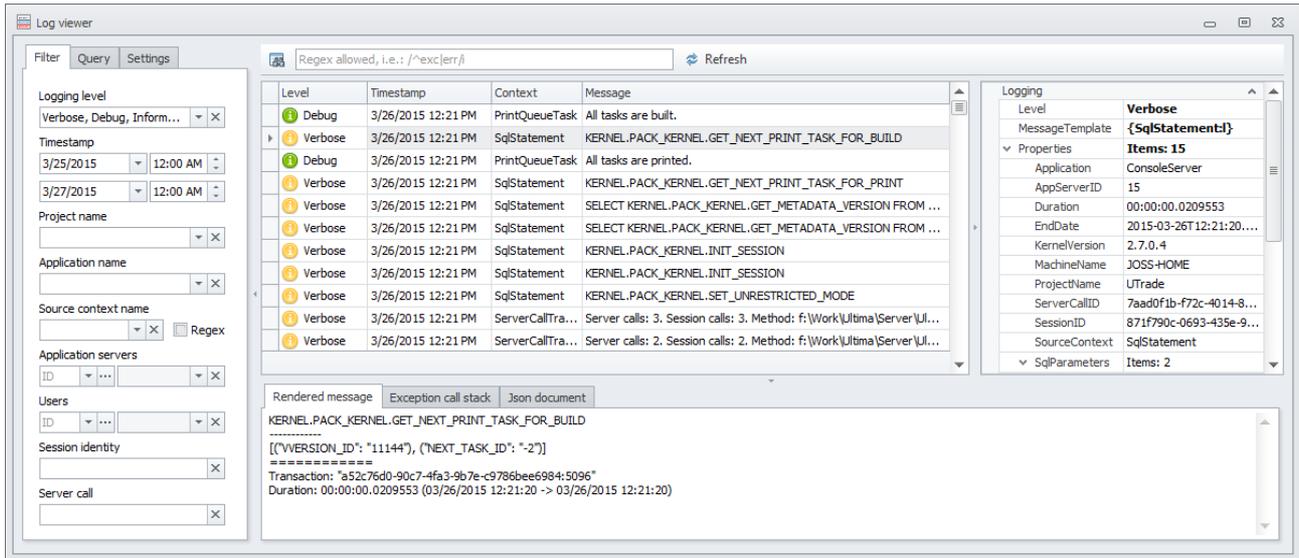
В системе Ultimate Solid реализовано центральное структурированное журналирование на базе библиотеки Serilog (<http://serilog.net/>):

- встроенный в систему анализатор логов "Log viewer", описанный ниже, работает только с базой данных MongoDB и подразумевает конкретную структуру документов-событий, генерируемую библиотекой Serilog;
- запуск и настройка MongoDB описаны в разделе [Аппаратный комплекс](#);
- настройки приложений для журналирования с использованием Serilog описана [там же](#);
- по умолчанию системой журналируются:
 - все SQL-запросы, для них в лог пишутся: время, текст, параметры, транзакция и серверный вызов;
 - все исключения на момент возникновения – *FirstChanceException*. Необходимость этого обусловлена тем, что сервер приложений не всегда действует от имени клиента. В заданиях,

обратных вызовах и явно созданных потоках может возникнуть исключение, которое приведет к прекращению работы сервера и не будет транслировано клиенту. Чтобы эта информация не потерялась и необходимо журналирование всех исключений;

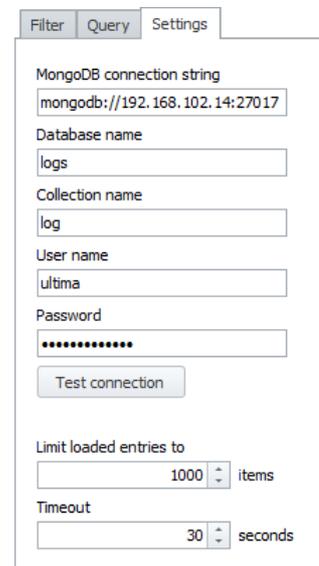
- любой оператор *throw*. По этой причине в логе можно увидеть даже исключения, которые чем-то перехватываются и не видны конечному пользователю;
- журналирование остальных событий должно быть инициализировано прикладным разработчиком. Это делается через интерфейсы *ILogger* и *ILogManager*, описанные в документации разработчика в разделе "Инструментарий разработчика / Скрипты / Специальные менеджеры".

Командой "View logs" открывается форма просмотра событий "Log viewer":



При первом запуске на закладке "Settings" в левой части формы необходимо выполнить настройки для подключения к базе MongoDB:

- Mongo DB connection string* – адрес базы в формате: `mongodb://127.0.0.1:27017`;
- User name* и *Password* – для каждого пользователя в MongoDB создается своя учетная запись с правами на чтение логов;
- Limit loaded entries to* – ограничение на количество отображаемых в форме событий;
- Timeout* – период ожидания в секундах отклика от базы.



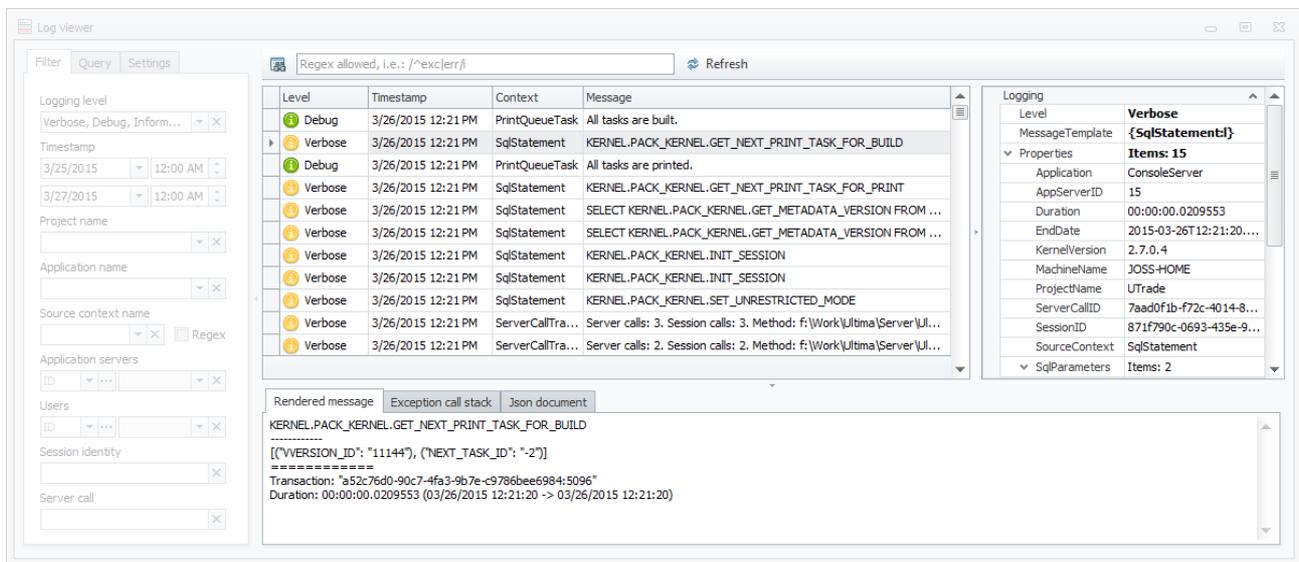
На закладке "Filter" осуществляется фильтрация отображаемых в форме событий.

При журналировании к событиям автоматически добавляются свойства контекста выполнения, как-то: код пользователя, идентификатор транзакции, текущего вызова и сессии. По коду сессии удобно отслеживать все события, которые произвел в системе некий конкретный пользователь в заданный промежуток времени. По идентификатору вызова можно выяснить, какие SQL-запросы были выполнены в течение некоего серверного вызова, и так далее.

На закладке "Query" приведен запрос на JSON к базе Mongo, который генерируется согласно настройкам фильтра за закладке "Filter". При необходимости можно писать произвольные запросы к хранилищу, изменив данный запрос вручную.

Запросы в Mongo формулируются в виде документов-образцов. В образце указываются либо значения интересующих полей (например, *Level: Debug*), либо специальные операторы сравнения (*\$in*, *\$gt*, *\$gte*, *\$lt*, *\$lte*, *\$regex* и так далее). Чтобы ограничить значения вложенного документа (скажем, *Properties.UserID*), используется точечная нотация. Более подробно тема освещена в документации к MongoDB.

Отфильтрованный список событий выводится в правой части формы. События отсортированы по убыванию даты. Список уже выведенных событий можно дополнительно фильтровать, в том числе и с использованием регулярных выражений (подробное описание синтаксиса которых можно найти на сайте MSDN [eng/rus](#)), в строке поиска  в верхней части формы. Данный фильтр применяется нажатием кнопки " Refresh":



The screenshot shows the Log viewer application. On the left, there is a 'Filter' panel with options for 'Logging level' (Verbose, Debug, Inform...), 'Timestamp' (3/25/2015, 12:00 AM), 'Project name', 'Application name', 'Source context name', 'Application servers', 'Users', 'Session identity', and 'Server call'. The main area displays a table of log events with columns: Level, Timestamp, Context, and Message. The selected event is: Level: Verbose, Timestamp: 3/26/2015 12:21 PM, Context: SqlStatement, Message: KERNEL.PACK_KERNEL.GET_NEXT_PRINT_TASK_FOR_BUILD. Below the table, there are tabs for 'Rendered message', 'Exception call stack', and 'Json document'. The 'Rendered message' tab is active, showing a JSON document: [{"VERSION_ID": "11144"}, {"NEXT_TASK_ID": "-2"}]. On the right, there is a 'Logging' panel showing properties for the selected event, including MessageTemplate, Properties (Items: 15), Application, AppServerID, Duration, EndDate, KernelVersion, MachineName, ProjectName, ServerCallID, SessionID, SourceContext, and SqlParameters (Items: 2).

События каждого уровня важности *Logging level* предваряются в списке соответствующей иконкой:

-  – уровень Verbose;
-  – уровень Debug;
-  – уровень Information;
-  – уровень Warning;
-  – уровень Error;
-  – уровень Fatal.

Справа в секции "Logging" отображаются свойства выбранного в списке события, которые могут иметь сложную древовидную структуру: исключения сохраняются вместе со вложенными исключениями и набором свойств Data. Снизу – подробная информация о выбранном в списке событии:

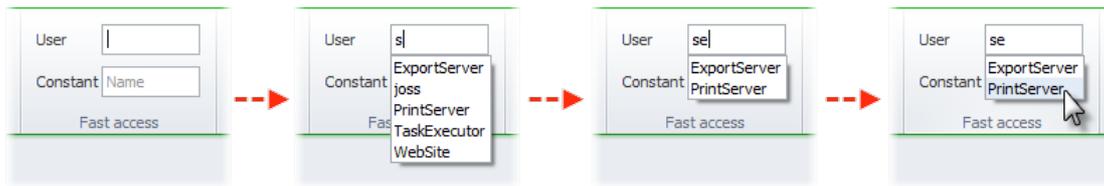
- на закладке "Rendered message" – текстовое описание события;
- на закладке "Exception call stack" – стек вызовов исключения;
- на закладке "Json document" – JSON-представление события.

Группа инструментов *Fast access*

Инструменты группы используются для быстрого доступа к объектам:

- *User* – выбор пользователя по его логину (*Login*);
- *Constant* – выбор константы по названию (*Name*).

При вводе значений в поле инструмента действует функция автоподстановки, которая предлагает выбор из списка объектов, название которых включает вводимый фрагмент:



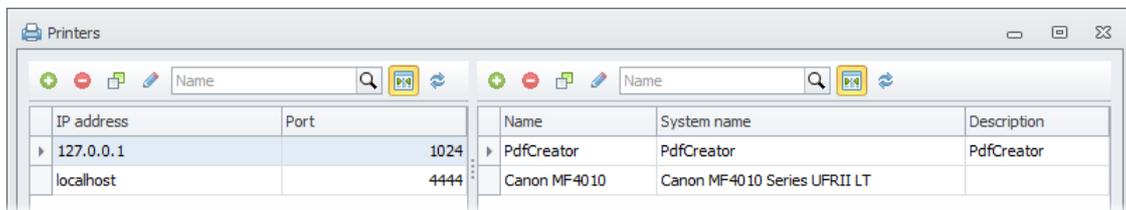
Выбранный объект открывается в форме редактирования соответствующего справочника. Выбор осуществляется кликом левой кнопкой мыши по объекту в списке автоподстановки или по нажатию на клавишу **Enter** (в этом случае вводимое название должно полностью совпадать со значением соответствующего свойства объекта).

Группа инструментов Printing

Принтеры



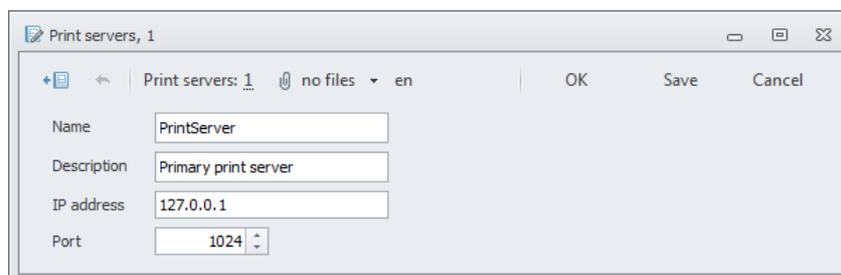
Управление принтерами и серверами печати осуществляется через форму "Printers":



Окно справочника разделено на две части: слева отображается список серверов печати, справа – принадлежащих им принтеров. Выбор сервера печати слева фильтрует список принтеров справа, оставляя только принадлежащие ему.

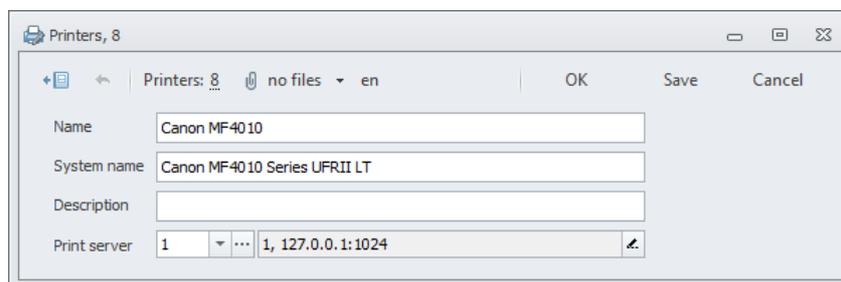
Сервера печати можно фильтровать по *Названию (Name)*, также по *Названию* можно фильтровать принтеры (*Name*).

Форма редактирования сервера печати позволяет задать, помимо *Названия (Name)* и *Описания (Description)*, *IP адрес* компьютера (*IP address*) и *Порт (Port)*, на котором он функционирует:



Идентификатор сервера печати, отображаемый в панели инструментов формы редактирования, используется при настройке приложения сервера печати (параметр *PrintServerID* в файле конфигурации *PrintServer.exe.config*).

Форма редактирования принтера позволяет задать:

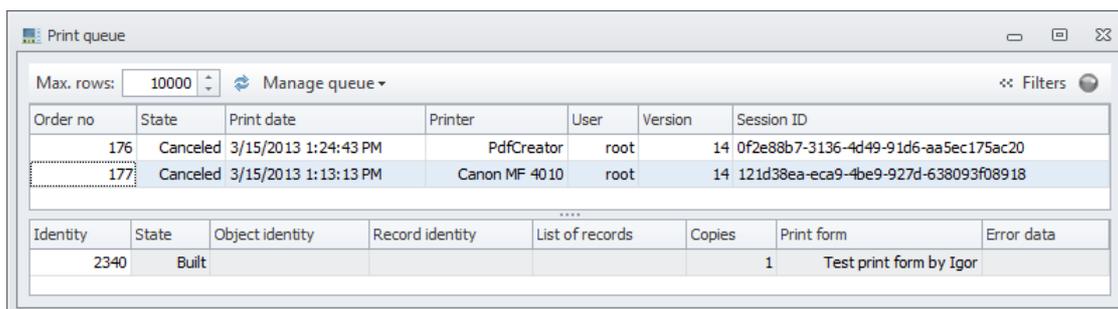


- *Name* – название принтера, которое будет отображаться в интерфейсе системы Ultimate Solid;
- *System name* – системное имя принтера, должно совпадать с именем соответствующего принтера в операционной системе компьютера, на котором функционирует PrintServer;
- *Description* – служебное поле, служащее для подробного, если это необходимо, описания принтера (эта информация нигде более не отображается);
- *Print server* – сервер печати, на котором функционирует принтер.

Очередь печати



Управление очередью печати осуществляется через форму "Print queue":



Order no	State	Print date	Printer	User	Version	Session ID
176	Canceled	3/15/2013 1:24:43 PM	PdfCreator	root	14	0f2e88b7-3136-4d49-91d6-aa5ec175ac20
177	Canceled	3/15/2013 1:13:13 PM	Canon MF 4010	root	14	121d38ea-eca9-4be9-927d-638093f08918

Identity	State	Object identity	Record identity	List of records	Copies	Print form	Error data
2340	Built				1	Test print form by Igor	

Окно очереди разделено на две части: сверху отображаются пакеты заданий, ожидающие печати, снизу – задания выбранного сверху пакета.

В списке пакетов заданий отображается следующая информация:

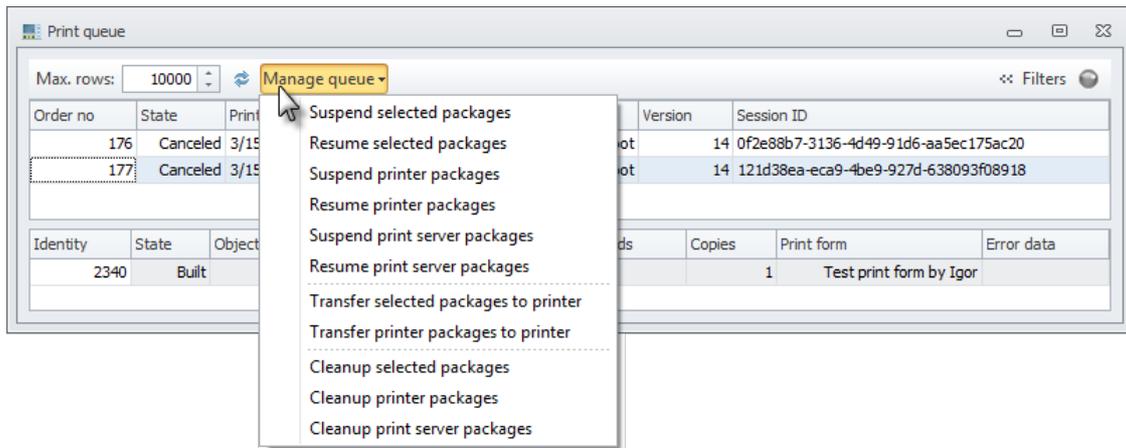
- *Order no* – номер пакета в очереди печати. Очередность печати пакетов соответствует порядку их номеров в очереди;
- *State* – статус пакета:
 - *Active* – пакет в печати;
 - *Canceled* – пакет отменен;
- *Print date* – дата отправки пакета на печать;
- *Printer* – принтер, на который был отправлен пакет;
- *User* – логин пользователя, отправившего пакет на печать;
- *Version* – версия метаданных, с которой работает клиентское приложение Ultimate Solid пользователя, отправившего пакет;
- *Session ID* – сессия, в которой был отправлен пакет.

Для заданий выбранного пакета отображается следующая информация:

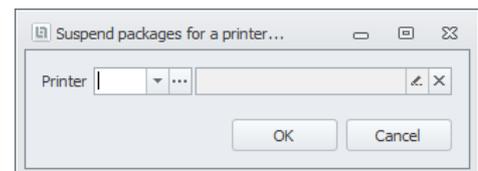
- *Identity* – идентификатор задания;

- *State* – статус задания:
 - *NonActive* – задание не активно;
 - *Building* – печатная форма задания строится;
 - *Built* – печатная форма задания построена;
 - *Printing* – задание печатается;
 - *Printed* – задание напечатано;
 - *Invalid* – ошибка при печати задания;
- *Object identity* – идентификатор объекта, запись или записи которого была отправлены на печать;
- *Record identity* – идентификатор записи, отправленной на печать;
- *List of records* – список записей, отправленных на печать;
- *Copies* – количество копий задания, отправленных на печать;
- *Print form* – печатная форма, использованная при печати задания;
- *Error data* – информация об ошибках.

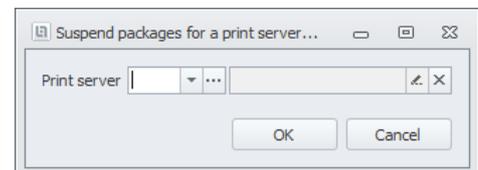
Функционал формы позволяет осуществлять следующие операции над пакетами заданий:



- *Suspend selected packages* – приостановить печать выбранных пакетов;
- *Resume selected packages* – возобновить печать выбранных пакетов. При возобновлении приостановленной печати пакеты будут перемещены в конец очереди, соответственно их номера *Order no* также будут изменены;
- *Suspend printer packages* – приостановить печать всех пакетов, отправленных на выбранный принтер. Операция предваряется открытием формы выбора принтера;

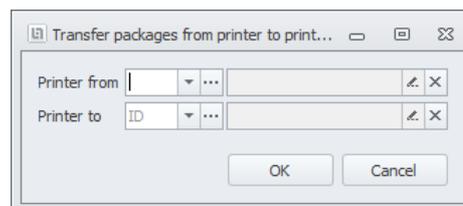


- *Resume printer packages* – возобновить печать всех пакетов, отправленных на выбранный принтер. Операция предваряется открытием формы выбора принтера. При возобновлении приостановленной печати пакеты будут перемещены в конец очереди;
- *Suspend print server packages* – приостановить печать всех пакетов, отправленных на выбранный сервер печати. Операция предваряется открытием формы выбора сервера печати;



- *Resume print server packages* – возобновить печать всех пакетов, отправленных на выбранный сервер печати. Операция предваряется открытием формы выбора сервера печати. При возобновлении приостановленной печати пакеты будут перемещены в конец очереди;
- *Transfer selected packages to printer* – переместить печать выбранных пакетов на выбранный принтер. Операция предваряется открытием формы выбора принтера. При изменении принтера печати пакеты будут перемещены в конец очереди;

- *Transfer printer packages to printer* – переместить на выбранный принтер печать всех пакетов, отправленных на определенный принтер. Операция предваряется открытием формы выбора принтеров. При изменении принтера печати пакеты будут перемещены в конец очереди;



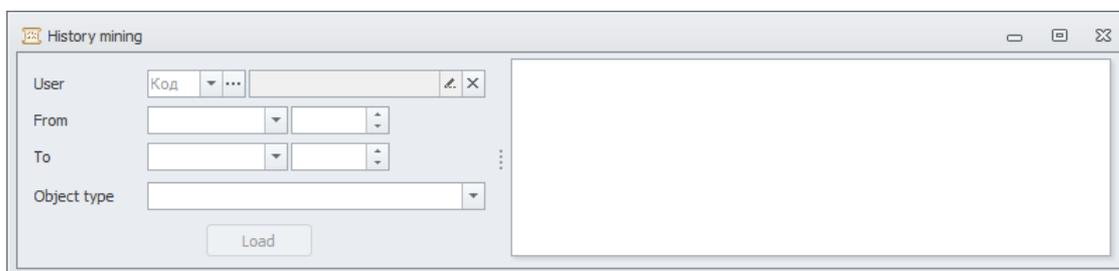
- *Cleanup selected packages* – отменить печать выбранных пакетов;
- *Cleanup printer packages* – отменить печать всех пакетов, отправленных на выбранный принтер. Операция предваряется открытием формы выбора принтера;
- *Cleanup print server packages* – отменить печать всех пакетов, отправленных на выбранный сервер печати. Операция предваряется открытием формы выбора сервера печати.

Группа инструментов History

Поиск по истории



Поиск по истории изменения объектов метаданных можно осуществить с помощью формы "History mining":

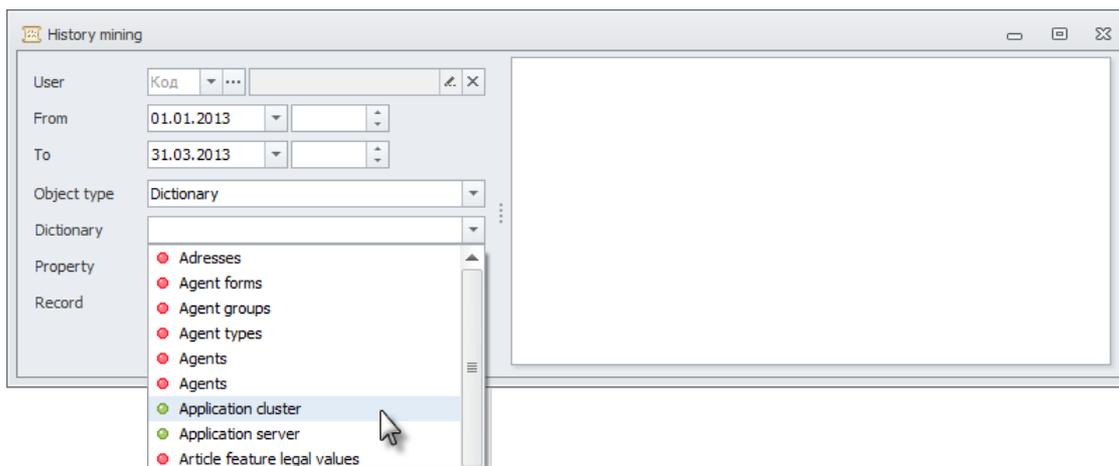


Инструмент поиска позволяет предельно детально локализовать область поиска с помощью следующих свойств:

- *User* – пользователь, совершивший изменений. Эта единственная опция, выбор которой не обязателен для осуществления поиска;
- *From* – дата (и опционально время) начала периода, в котором осуществляется поиск;
- *To* – дата (и опционально время) конца периода, в котором осуществляется поиск;
- *Object type* – тип объекта, поиск по истории изменений которого осуществляется:
 - *Dictionary* – словарь;
 - *Document* – документ;
 - *Link table* – развязочная таблица;
 - *Table part of document* – табличная часть документа.

Поиск осуществляется только по несистемным объектам. То есть, например, поиск по истории изменения ядерных (системных) справочников, содержащих метаданные, осуществить нельзя.

В зависимости от выбранного типа объекта станет доступной соответствующая ему дополнительная информация:



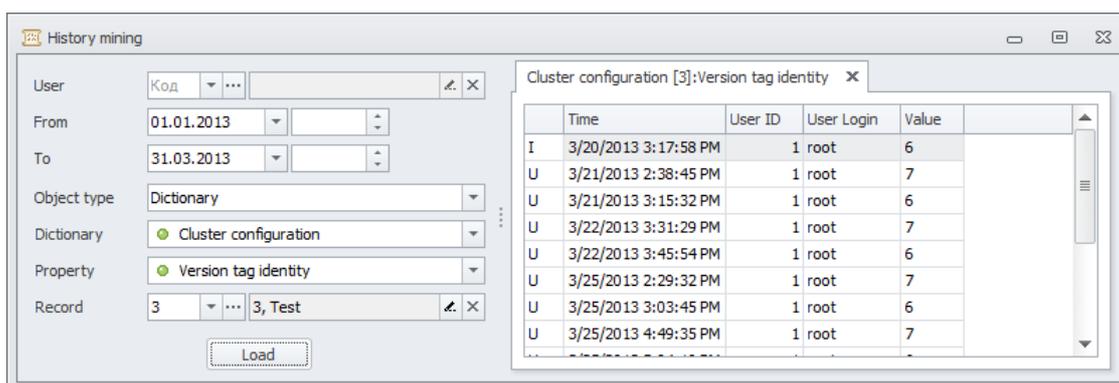
При локализации объекта, например, выборе конкретного справочника и его свойства, поиск изменений которого необходимо выполнить, можно увидеть дополнительные иконки перед названием объекта. Зеленая иконка ● означает, что в данный момент журналирование по данному объекту включено. Красная иконка ● означает, что в данный момент журналирование по объекту выключено ([Настройка журналирования объектов метаданных](#) подробно описана в соответствующем разделе).



Иконки ● и ● показывают текущий статус журналирования объекта на момент осуществления поиска.

Вполне возможна ситуация, когда журналирования для объекта, помеченного красной иконкой ●, было выключено только накануне, и поиск более ранних изменений по нему будет все равно результативен.

Результаты поиска по истории изменений выводятся в правой части формы на закладке, в заголовке которой отражены критерии поиска. Новый результат поиска будет выведен на новой закладке:



В заголовке закладки, отражающей критерии поиска, информация приведена в следующем формате *ObjectName [ObjectID]: ObjectPropertyName*, где:

- *ObjectName* – объект метаданных, по которому осуществляется поиск;
- *ObjectID* – идентификатор записи объекта;
- *ObjectPropertyName* – дочернее свойство объекта, поиск изменений которого осуществляется.

По найденным изменениям доступна следующая информация:

- *первая безымянная колонка* – тип операции, в результате которой произошло изменение значения объекта:
 - I – insert, значение было вставлено (впервые);

- *U* – update, значение было изменено;
- *D* – delete, значение было удалено;
- *Time* – время совершения изменения;
- *User ID* – идентификатор пользователя, совершившего изменения;
- *User* – логин пользователь, совершившего изменения;
- *Value* – значение объекта после изменения.

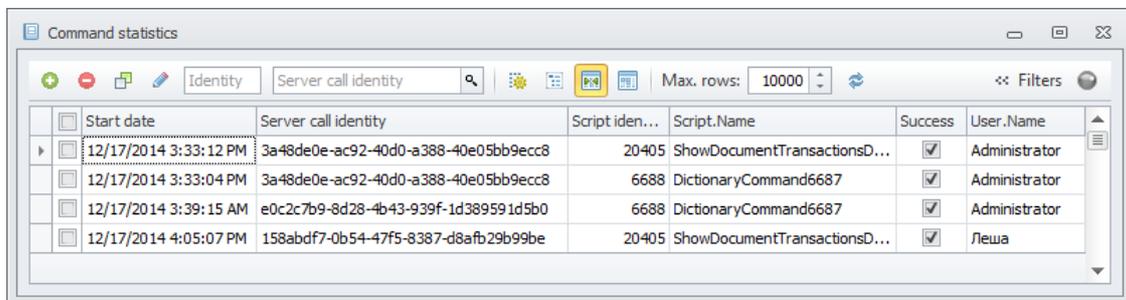


Поиск по истории также встроен прямо в формы редактирования справочников и документов. Если выбрано любое поле справочника или документа, нажатие горячей клавиши **Alt + H** показывает историю изменений по этому полю. Аналогичным образом это сочетание клавиш работает в табличных частях.

Статистика вызова команд



Поиск по истории вызова интерактивных команд можно осуществлять в справочнике "Command statistics":



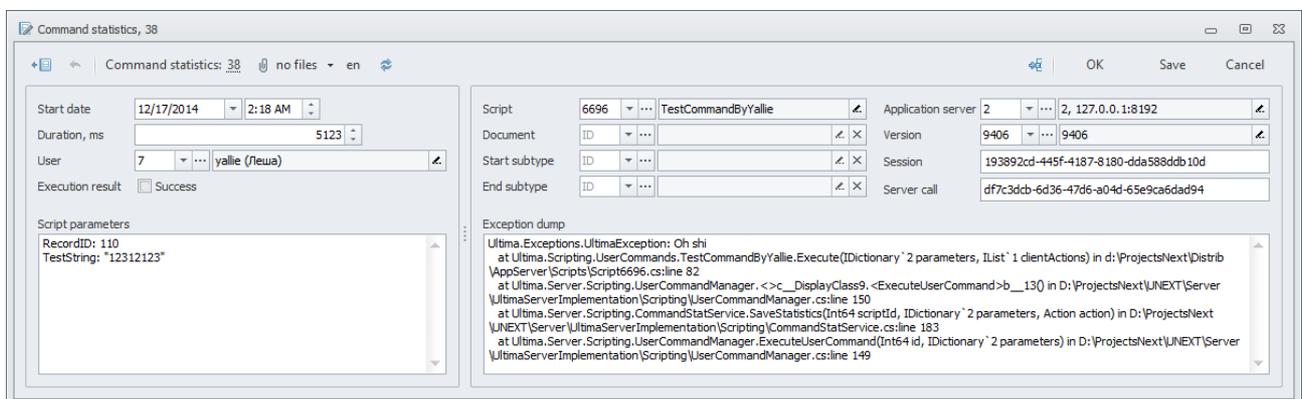
Start date	Server call identity	Script iden...	Script.Name	Success	User.Name
12/17/2014 3:33:12 PM	3a48de0e-ac92-40d0-a388-40e05bb9ecc8	20405	ShowDocumentTransactionsD...	<input checked="" type="checkbox"/>	Administrator
12/17/2014 3:33:04 PM	3a48de0e-ac92-40d0-a388-40e05bb9ecc8	6688	DictionaryCommand6687	<input checked="" type="checkbox"/>	Administrator
12/17/2014 3:39:15 AM	e0c2c7b9-8d28-4b43-939f-1d389591d5b0	6688	DictionaryCommand6687	<input checked="" type="checkbox"/>	Administrator
12/17/2014 4:05:07 PM	158abdf7-0b54-47f5-8387-d8afb29b99be	20405	ShowDocumentTransactionsD...	<input checked="" type="checkbox"/>	Леша

Системой фиксируются вызовы:

- пользовательских команд;
- команд над документом;
- команд над списком документов;
- команд над записью справочника;
- команд над записями справочника.

Записи справочника можно фильтровать по *Идентификатору серверного вызова (Server call identity)*.

Для каждой вызванной команды в системе сохраняется следующая информация:



Start date	12/17/2014 2:18 AM	Script	6696 TestCommandByYalie	Application server	2 2, 127.0.0.1:8192
Duration, ms	5123	Document	ID	Version	9406 9406
User	7 yalie (Леша)	Start subtype	ID	Session	193892cd-445f-4187-8180-dda588ddb10d
Execution result	Success	End subtype	ID	Server call	df7c3dcb-6d36-47d6-a04d-65e9ca6dad94
Script parameters	RecordID: 110 TestString: "12312123"	Exception dump	<pre> Ultima.Exceptions.UltimaException: Oh shi at Ultima.Scripting.UserCommands.TestCommandByYalie.Execute(IDictionary`2 parameters, IList`1 clientActions) in d:\Projects\Next\Distri VAppServer\Scripts\Script6696.cs:line 82 at Ultima.Server.Scripting.UserCommandManager.<>c__DisplayClass9.<ExecuteUserCommand>b__130 in D:\Projects\Next\JNEXT\Server \UltimaServerImplementation\Scripting\UserCommandManager.cs:line 150 at Ultima.Server.Scripting.CommandStatService.SaveStatistics(Int64 scriptId, IDictionary`2 parameters, Action action) in D:\Projects\Next \JNEXT\Server\UltimaServerImplementation\Scripting\CommandStatService.cs:line 183 at Ultima.Server.Scripting.UserCommandManager.ExecuteUserCommand(Int64 id, IDictionary`2 parameters) in D:\Projects\Next\JNEXT\Server \UltimaServerImplementation\Scripting\UserCommandManager.cs:line 149 </pre>		

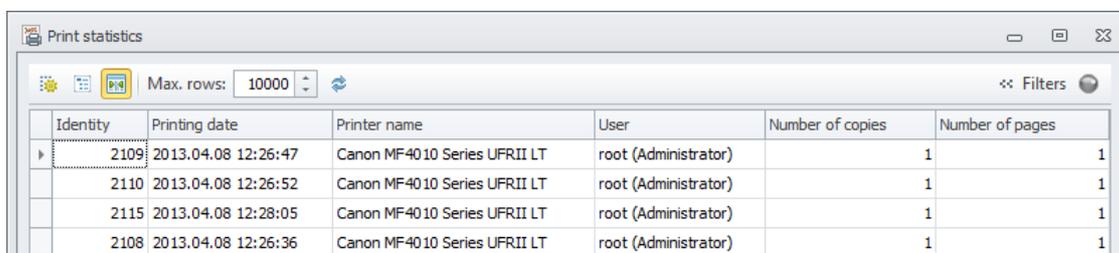
- *Start date* – дата и время запуска команды;
- *Duration* – продолжительность выполнения команды в миллисекундах;
- *User* – пользователь, запустивший команду;

- *Execution result* – результат выполнения команды. Выполнение успешно, если установлен флаг *Success*;
- *Script parameters* – параметры и их значения, переданные при выполнении команды скрипту *Script*, в формате [*Параметр*]: [*Значение*];
- *Script* – скрипт вызванной команды;
- *Document* – документ, над которым выполнялась команда;
- *Start subtype* – начальный подтип документа перед выполнением команды;
- *End subtype* – конечный подтип документа по завершении команды;
- *Application server* – сервер приложений, на котором выполнялась команда;
- *Version* – версия метаданных, на которой выполнялась команда;
- *Session* – идентификатор сессии, в которой выполнялась команда;
- *Server call* – серверный вызов, в котором выполнялась команда;
- *Exception dump* – исключение, если таковое возникло в результате выполнения команды.

Статистика печати



Статистику печати можно посмотреть с помощью формы "Print statistic":



Identity	Printing date	Printer name	User	Number of copies	Number of pages
2109	2013.04.08 12:26:47	Canon MF4010 Series UFR II LT	root (Administrator)	1	1
2110	2013.04.08 12:26:52	Canon MF4010 Series UFR II LT	root (Administrator)	1	1
2115	2013.04.08 12:28:05	Canon MF4010 Series UFR II LT	root (Administrator)	1	1
2108	2013.04.08 12:26:36	Canon MF4010 Series UFR II LT	root (Administrator)	1	1

По каждой записи статистики можно получить о печати следующую информацию:

- *Printing date* – дата и время завершения печати и удаления задания из [очереди печати](#);
- *Printing user* – пользователь, инициировавший печать;
- *Printer* – принтер, на который осуществлялась печать;
- *Printer Name* – системное имя принтера, на который осуществлялась печать;
- *Print from* – печатная форма, которая была использована при печати;
- *Version* – версия метаданных;
- *Number of pages* – число страниц напечатанного задания (одной копии);
- *Number of copies* – количество копий напечатанного задания. Общее количество напечатанных страниц задания будет равно произведению числа страниц задания (*Number of pages*) на число копий его (*Number of copies*).

Группа инструментов Open by ID

Инструменты группы используются для выполнения следующих операций над объектам:

- *Document subtype history* – просмотр истории изменения подтипа выбранного по *идентификатору (Document ID)* документа ;
- *Restore document* – восстановление (попытка) удаленного документа по его *идентификатору (Document ID)*.

Для осуществления операции над объектом необходимо ввести его идентификатор в соответствующее поле и нажать на клавишу **Enter**.

Группа инструментов Manage

Настройка журналирования объектов метаданных



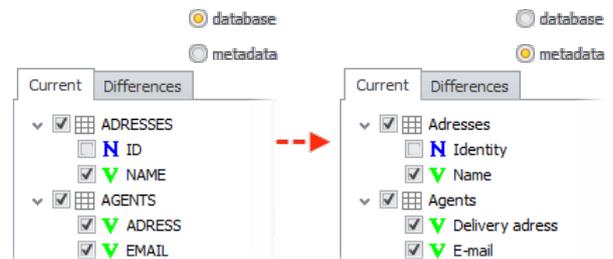
В системе Ultimate Solid реализована функция журналирования, которая предоставляет возможность хранения истории изменений, вносимых в несистемные таблицы базы данных. Инструмент [поиска по истории](#) изменений можно найти в группе инструментов History.

Настройки журналирования осуществляются с помощью формы "Manage history":



Суть настроек сводится к отметке флагами свойств объектов, изменения которых необходимо журналировать:

- в левом верхнем углу формы флагами можно отметить типы объектов, которые необходимо отображать на закладке "Current":
 - *dictionaries* – словари;
 - *link table* – развязочные таблицы;
 - *documents* – типы документов;
 - *table parts* – табличные части;
 - *other* – прочие объекты;
 - *totals* – итоги;
- в правом верхнем углу формы можно выбрать формат отображения названий объектов (*Names from*) на закладках "Current" и "Differences":
 - *database* – названия объектов (таблиц и их колонок) в базе данных;
 - *metadata* – названия соответствующих объектам классов метаданных;
- *SQL script* – по нажатию на кнопку открывается форма, содержащая SQL скрипт, который предназначен для применения сделанных настроек журналирования. Скрипт может быть необходим в случае, если настройки сделаны в не самое оптимальное для их применения время, и их следует выполнить позже:



```

1 KERNEL.PACK_LOG.ENABLE_LOGGING('ADRESSES', 'NAME', 'ULTIMA');
2 KERNEL.PACK_LOG.ENABLE_LOGGING('AGENTS', 'ADDRESS,EMAIL, ID,NAME,PHONE, SOME_COLUMN,TESTTTT_ID', 'ULTIMA');
3 KERNEL.PACK_LOG.ENABLE_LOGGING('GOODS', 'FULL_NAME,NAME', 'ULTIMA');
4

```

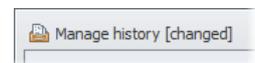
Скрипт следует выполнить в схеме *Kernel* в каком-либо приложении для работы с реляционными базами данных, поддерживающем SQL (PL SQL Developer, TOAD и т.п.);

- *Apply* – по нажатию на кнопку сделанные настройки журналирования будут применены немедленно.

На закладке "Current" в древовидной структуре (объект-родитель и его свойства-потомки) отображаются текущие настройки журналирования для свойств объектов, типы которых отмечены флагами в левом верхнем углу формы. Журналируются все отмеченные флагами свойства. Здесь же можно включить журналирование для новых свойств, отметив их флагами, или снять флаги для свойств, в журналировании которых уже нет необходимости. Названия свойств предваряются иконками, обозначающими их тип:

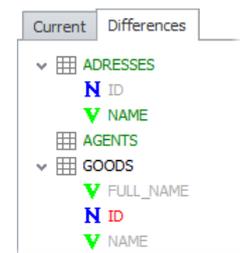
- *N* – числовые свойства (number) – типы *long*, *decimal* или *bool*;
- *D* – даты (date) – типы *date* или *DateTime*;
- *V* – текст (varchar), длиной не превышающий 4'000 символов – типы *string*, *text* и *LargeText*;
- *C* – текст (clob), длиной превышающий 4'000 символов – тип *LargeText*;
- *B* – массивы/бинарные файлы (blob) – тип *byte[]*.

Если в настройки журналирования внесены правки, которые еще не были применены, в заголовке формы будет отображаться соответствующее предупреждение: *changed*.



На закладке "Differences" в древовидной структуре отображаются внесенные в настройки журналирования и еще не сохраненные правки:

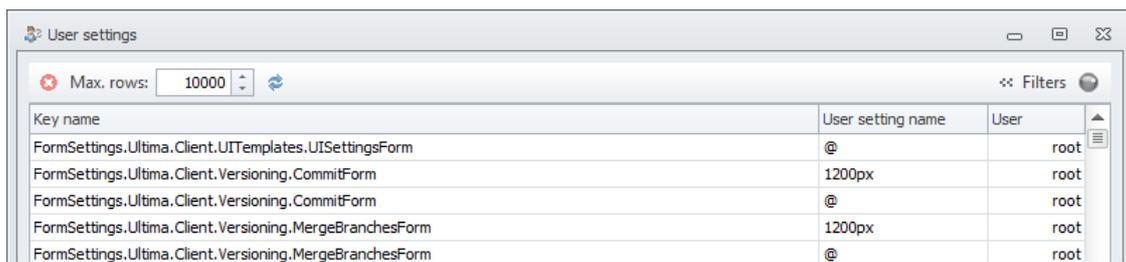
- **зеленым** отмечены объекты и их свойства, журналирование которых было включено;
- при этом, если **зеленым** отмечен объект, у которого нет вложенных свойств, значит журналирование было включено для всех свойств этого объекта;
- если журналирование было включено не для всех свойств объекта, нежурналируемые свойства будут отмечены **серым**, при этом сам объект будет отмечен **зеленым**;
- **красным** отмечены объекты и их свойства, журналирование которых было выключено;
- при этом, если **красным** отмечен объект, у которого нет вложенных свойств, значит журналирование было выключено для всех свойств этого объекта;
- если журналирование было выключено не для всех свойств объекта, по прежнему журналируемые свойства будут отмечены **серым** (при этом сам объект будет отмечен черным).



Пользовательские настройки



Пользователи могут сохранять настройки активного окна/закладки (размер окна, размер и расположение колонок, их сортировку и т.п.). Настройки можно сохранить в шаблон по умолчанию или задать шаблону произвольное имя. Список всех пользовательских настроек можно найти в форме "User settings":

Key name	User setting name	User
FormSettings.Ultima.Client.UITemplates.UISettingsForm	@	root
FormSettings.Ultima.Client.Versioning.CommitForm	1200px	root
FormSettings.Ultima.Client.Versioning.CommitForm	@	root
FormSettings.Ultima.Client.Versioning.MergeBranchesForm	1200px	root
FormSettings.Ultima.Client.Versioning.MergeBranchesForm	@	root

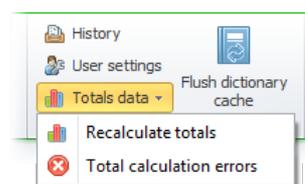
О каждой настройке предоставлена следующая информация:

- *Key name* – объект, настройки которого были сохранены. Название объекта представляет собой запись формата *FormSettings.XXX*, где *XXX* – название формы;
- *User setting name* – название шаблона, в который были сохранены настройки. Шаблону по умолчанию соответствует название "@";
- *User* – логин пользователя, сохранившего настройки.

Выбранные шаблоны с настройками можно удалить кнопкой  в панели инструментов. Удаление может быть необходимо в случае, если изменения шаблона препятствуют открытию формы, конфликту, например, с изменениями, внесенными в эту форму прикладным разработчиком.

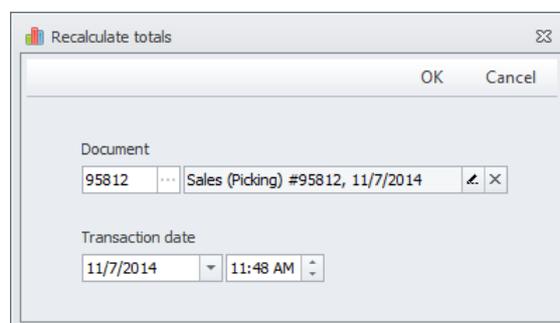
Пересчет итогов

 Пересчет итогов можно запустить принудительно с помощью команды *Recalculate totals*.



В открывшейся форме "Recalculate totals" необходимо указать дату, с которой следует пересчитать итоги. Сделать это можно двумя способами:

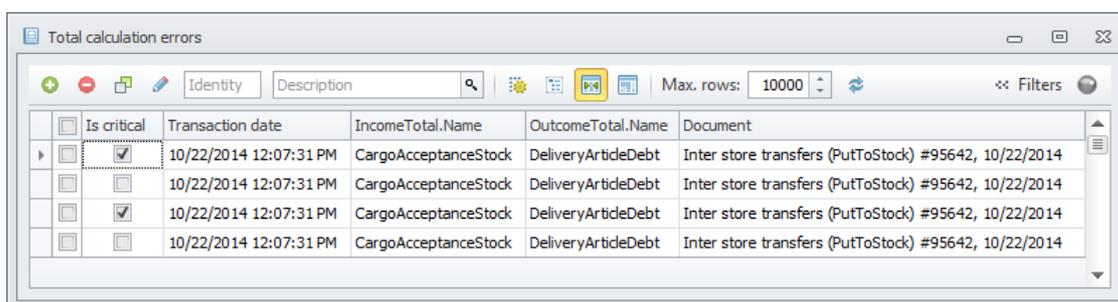
- указать дату и время непосредственно в поле *Transaction date*;
- ввести в поле *Document* код документа, изменение которого вызвало необходимость пересчитать итоги. Дата его проведения автоматически подставится в поле *Transaction date*.



По нажатию на кнопку "OK" будет запущен процесс пересчета итогов с указанной даты.

 Ошибки, возникшие при расчете итогов, можно посмотреть с помощью команды *Total calculation errors*.

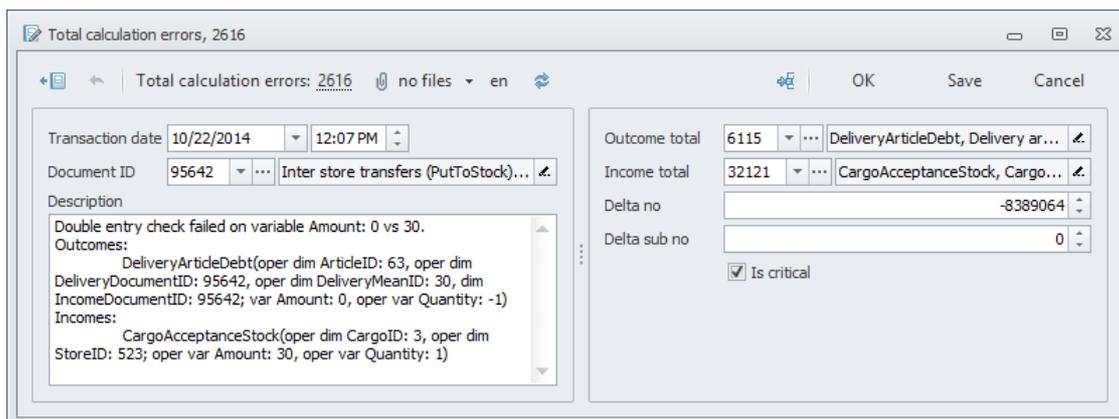
В открывшейся списковой форме "Total calculation errors" приведен перечень всех ошибок:



	Is critical	Transaction date	IncomeTotal.Name	OutcomeTotal.Name	Document
<input type="checkbox"/>	<input checked="" type="checkbox"/>	10/22/2014 12:07:31 PM	CargoAcceptanceStock	DeliveryArticleDebt	Inter store transfers (PutToStock) #95642, 10/22/2014
<input type="checkbox"/>	<input type="checkbox"/>	10/22/2014 12:07:31 PM	CargoAcceptanceStock	DeliveryArticleDebt	Inter store transfers (PutToStock) #95642, 10/22/2014
<input type="checkbox"/>	<input checked="" type="checkbox"/>	10/22/2014 12:07:31 PM	CargoAcceptanceStock	DeliveryArticleDebt	Inter store transfers (PutToStock) #95642, 10/22/2014
<input type="checkbox"/>	<input type="checkbox"/>	10/22/2014 12:07:31 PM	CargoAcceptanceStock	DeliveryArticleDebt	Inter store transfers (PutToStock) #95642, 10/22/2014

Записи справочника можно фильтровать по *Описанию* ошибки (*Description*).

Для каждой ошибки в системе хранятся:



- *Transaction date* – дата выполнения транзакции;
- *Document* – документ, по которому выполнялась транзакция;
- *Description* – описание ошибки;
- *Outcome total* – итог, с которого выполняется проводка;
- *Income total* – итог, на который выполняется проводка;
- *Delta no* – номер проводки;
- *Delta sub no* – номер рассчитанной проводки после расслоения;
- *Is critical* – установленный флаг свидетельствует о критичности ошибки, например, нарушении правила двойной записи.

Сброс кеша справочников



Функция сброса кеша справочников *Flush dictionary cache* посылает на сервер приложений команду, о необходимости сбросить кеш всех справочников на локальных компьютерах пользователей. Сервер в свою очередь рассылает эту команду всем подключенным в данный момент клиентам. Клиенты, получившие команду, сбрасывают кеш данных справочников.

Группа инструментов *WebService*

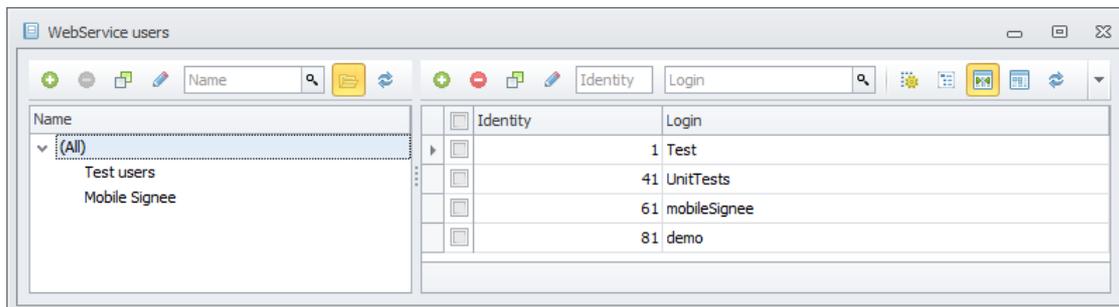
Доступ к web сервисам ограничен и предоставляется только авторизованным пользователям web сервисов.

Выдача прав на выполнение web сервисов реализована посредством ролей по аналогии с обычными пользователями системы. Пользователю web сервиса может быть назначено несколько ролей web сервисов. Роль, в свою очередь, может давать доступ к нескольким web сервисам. В итоге пользователь получает доступ ко всем web сервисам всех выбранных ролей.

Пользователи web сервисов



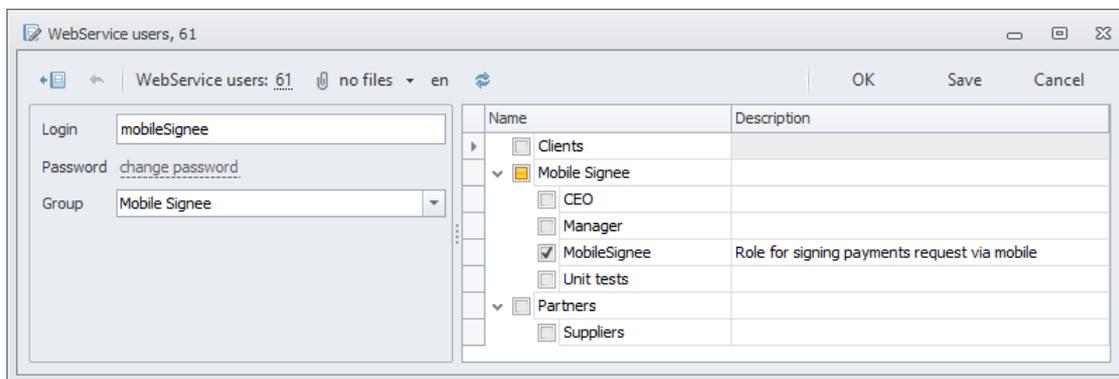
Список всех пользователей web сервисов можно найти в справочнике "WebService users":



Окно справочника разделено на две части: слева отображается дерево групп пользователей, справа – список пользователей выбранной слева группы.

Группы пользователей можно фильтровать по *Названию* группы (*Name*), а пользователей по *Логину* (*Login*).

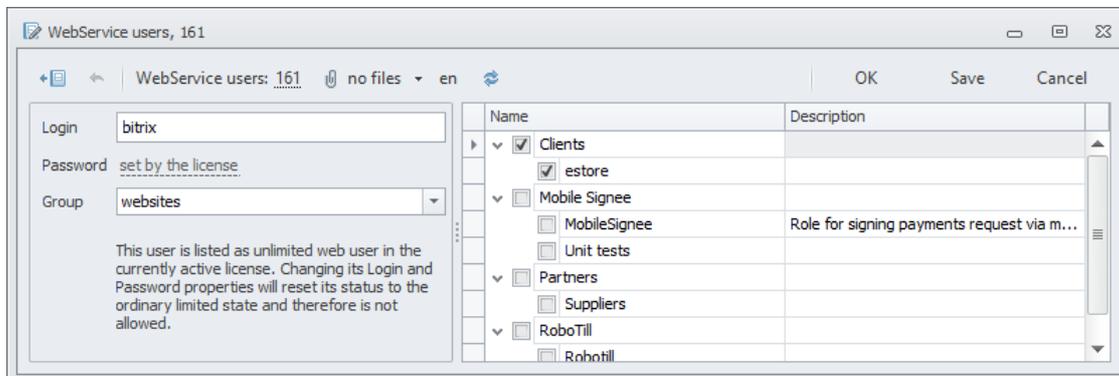
Форма редактирования пользователей web сервисов разделена на две части:



Слева перечислены свойства пользователя:

- *Login* – логин;
- *Password* – ссылка *change password*, по нажатию на которую открывается форма смены пароля;
- *Group* – группа, в которую входит пользователь.

Для безлимитных пользователей веб-сервисов пароль задается в файле лицензии, поэтому смена пароля для них запрещена:



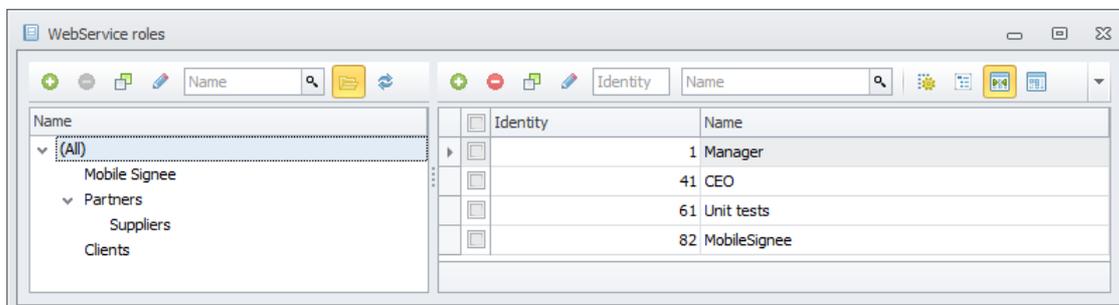
Справа выведено дерево всех ролей web сервисов, в котором флагами отмечены роли, назначенные данному пользователю. Будучи назначенной пользователю, роль дает ему доступ ко всем своим web сервисам. Одному пользователю может быть назначено несколько ролей.

Список назначенных ролей можно редактировать установкой или снятием флагов. Выбор группы в дереве приводит к выбору всех входящих в нее ролей.

Роли web сервисов



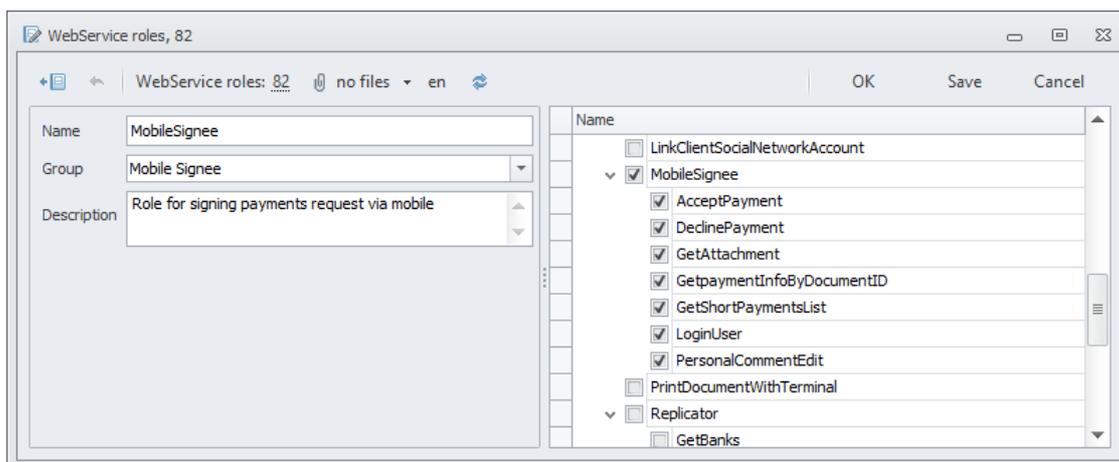
Список всех ролей web сервисов можно также найти в справочнике "WebService roles":



Окно справочника разделено на две части: слева отображается дерево групп ролей, справа – список ролей выбранной слева группы.

Группы ролей и сами роли можно фильтровать по *Названию (Name)*.

Форма редактирования ролей web сервисов разделена на две части:



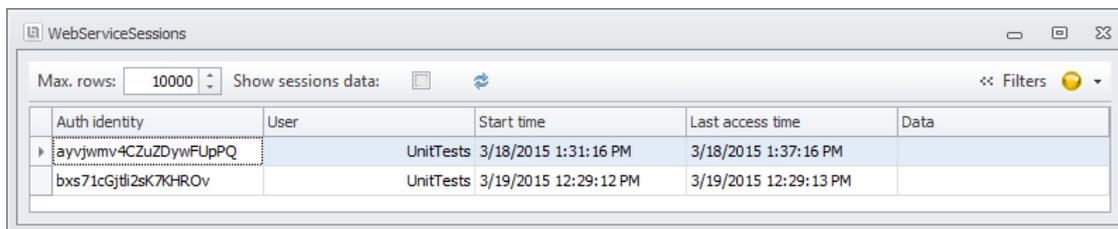
Слева перечислены свойства пользователя:

- *Name* – название роли;
- *Group* – группа, в которую входит роль;
- *Description* – описание роли.

Справа выведено дерево всех web сервисов, в котором флагами отмечены те из них, доступ к которым разрешен обладателю роли. Список доступных web сервисов можно редактировать установкой или снятием флагов. Выбор группы в дереве приводит к выбору всех входящих в нее web сервисов.

Сессии web сервисов

Список всех сессий web сервисов можно найти в форме "WebServiceSessions":



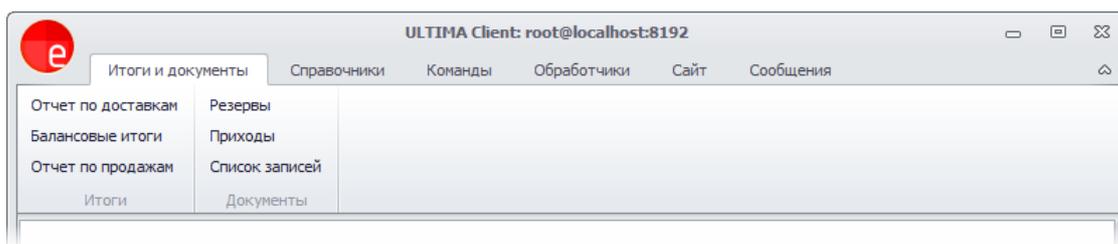
Auth identity	User	Start time	Last access time	Data
ayvjwmv4CzuZDywFUpPQ	UnitTests	3/18/2015 1:31:16 PM	3/18/2015 1:37:16 PM	
bxs71cGjtl2sK7KHROv	UnitTests	3/19/2015 12:29:12 PM	3/19/2015 12:29:13 PM	

- *Auth identity* – идентификатор сессии;
- *User* – логин пользователя web сервиса;
- *Start time* – время открытия сессии;
- *Last access time* – время последнего обращения;
- *Data* – данные сессии, выводятся с установленным в панели инструментов формы флагом *Show sessions data*.

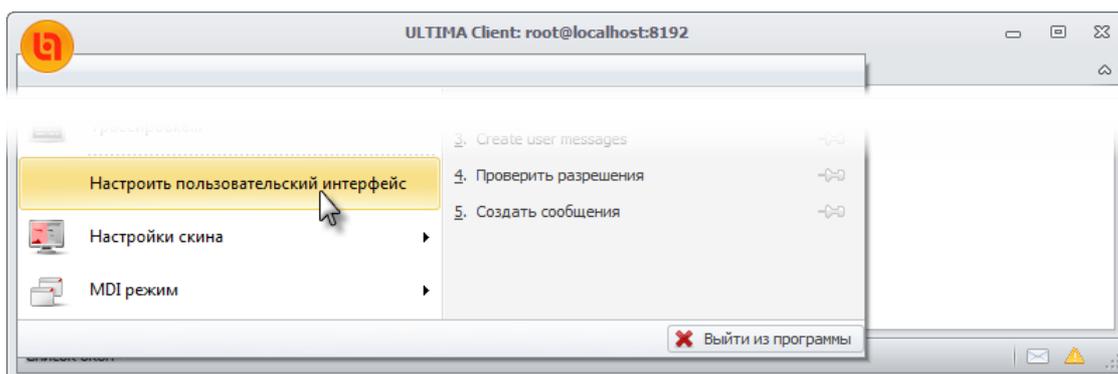
Время жизни сессий задается в [настройках кластера](#).

Настройка пользовательского интерфейса

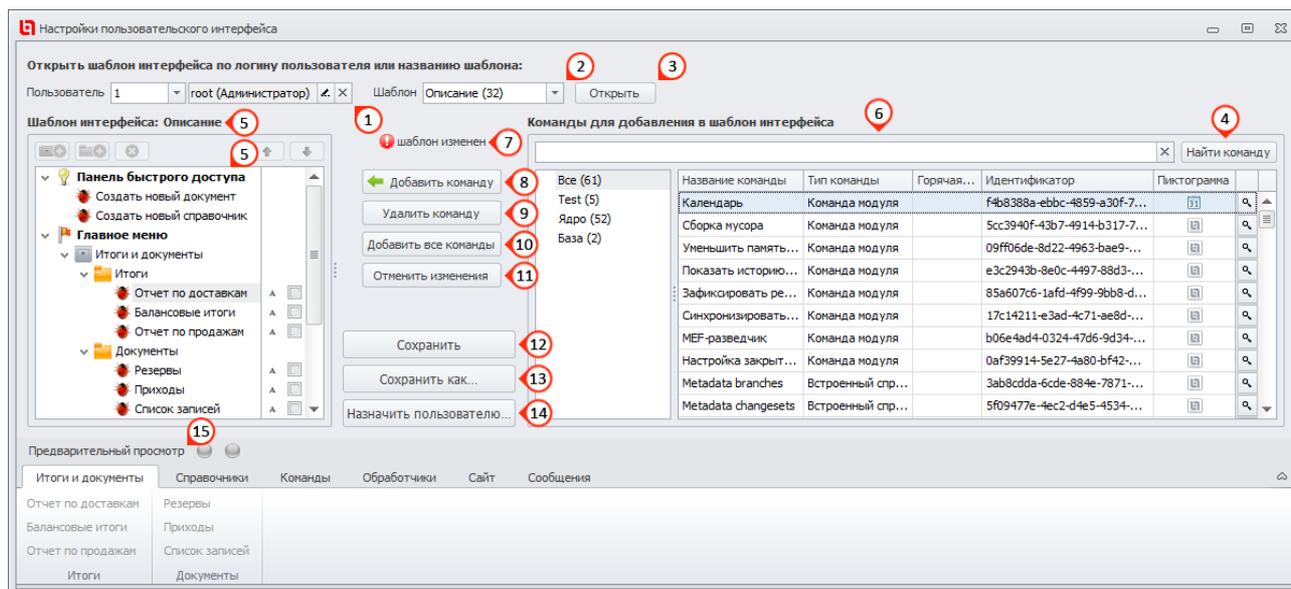
Внешний вид главного меню главной формы клиентского приложения можно настраивать:



Инструмент настройки вызывается через пункт меню "Настроить пользовательский интерфейс" или сочетанием горячих клавиш **Ctrl + Alt + C**:



При открытии формы настройки пользовательского интерфейса загружается шаблон интерфейса текущего пользователя:



При настройке пользовательского интерфейса доступны следующие опции:

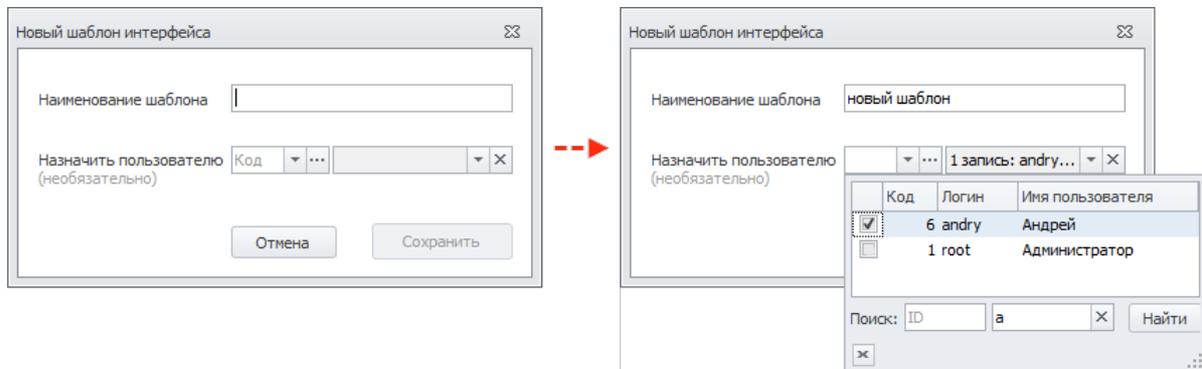
- (1) раскрывающийся список, с помощью которого выбирается пользователь, интерфейс которого необходимо настроить. По умолчанию выбран пользователь, под которым произведен вход в систему. В списке (2) автоматически выбирается привязанный к выбранному пользователю шаблон интерфейса;
- (2) раскрывающийся список, с помощью которого выбирается уже существующий шаблон интерфейса, который необходимо отредактировать.



Название шаблона интерфейса не уникально и может повторяться. В этом случае различить их можно по идентификационному номеру.

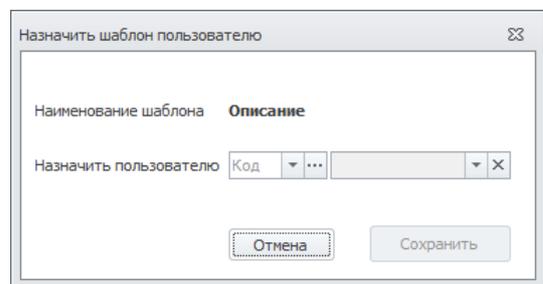
- (3) кнопка "Открыть" загружает выбранный шаблон интерфейса;
- (4) кнопка "Найти команду" ищет в списке команду по названию;
- (5) структура панели быстрого доступа и главного меню загруженного шаблона интерфейса в виде дерева закладок, групп и команд. Название загруженного шаблона дублируется в конце заголовка;
- (6) набор доступных для добавления в шаблон интерфейса команд;
- (7) при внесении в загруженный шаблон любых изменений появляется индикатор "шаблон изменен";
- (8) кнопка "Добавить команду" добавляет выбранную в списке доступных команду в загруженный шаблон интерфейса;
- (9) кнопка "Удалить команду" удаляет из интерфейса выбранную в его наборе команду, группу или закладку со всем содержимым;
- (10) кнопка "Добавить все команды" добавляет все выбранные в списке доступных команды (согласно фильтру) в загруженный шаблон интерфейса;
- (11) кнопка "Отменить изменения" возвращает загруженный шаблон в исходное состояние на момент последнего сохранения или открытия, если сохранение не производилось;
- (12) кнопка "Сохранить" сохраняет внесенные в шаблон интерфейса изменения;

- (13) кнопка "Сохранить как..." сохраняет интерфейс со всеми внесенными изменениями в новый шаблон.
 При ее нажатии открывается окно нового шаблона интерфейса. В поле "Название шаблона" вводится название нового шаблона. В раскрывающемся списке "Назначить пользователю" может быть опционально выбран один и более пользователей, которым будет по сохранении интерфейса назначен данный новый шаблон:



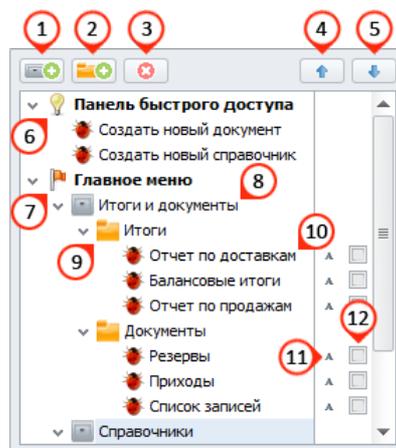
При сохранении новый шаблон будет открыт в секции (5). В редактируемый ранее интерфейс (если он не сохранялся) изменения внесены не будут.

- (14) кнопка "Назначить пользователю..." позволяет назначить загруженный шаблон интерфейса одному или более пользователям.
 При ее нажатии открывается окно назначения шаблона пользователю. В раскрывающемся списке "Назначить пользователю" можно выбрать одного и более пользователей, которым будет назначен данный шаблон:



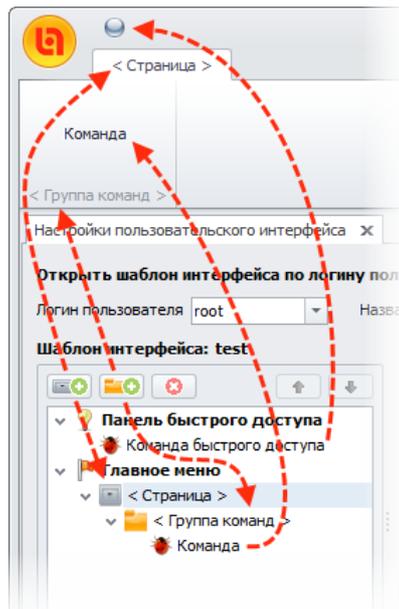
- (15) предварительный просмотр создаваемого главного меню, где отображаются в режиме реального времени все вносимые изменения. В нем можно переключаться между закладками, сворачивать и разворачивать меню, недоступно только выполнение команд. При выборе в "Шаблон интерфейса" (5) закладки, или относящейся к ней группы или команды, она будет открыта и в области предварительного просмотра (15).

В окне "Шаблон интерфейса: ..." отображается набор команд "Панели быстрого доступа" (6) и в древовидной иерархии структура "Главного меню" (7):

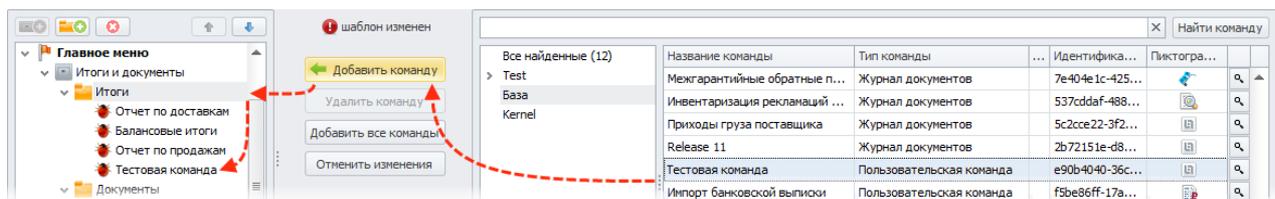


В верхней части секции расположен набор команд:

- кнопка (1) добавляет новую закладку (8) в главное меню (7),
 кнопка (2) добавляет новую группу (9) в выбранную в "Наборе команд интерфейса" закладку (8):

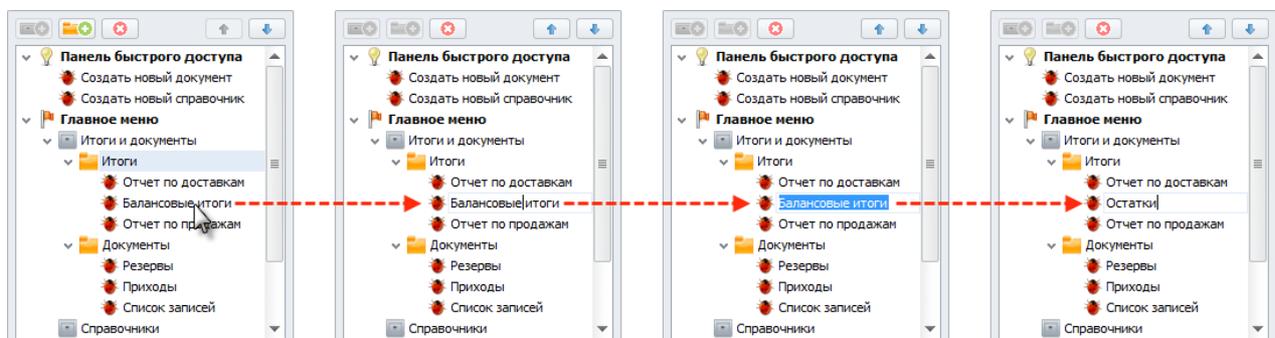


- если в "Наборе команд интерфейса" выбрана команда (10), группа (9) или "Панель быстрого доступа" (6) становится активной кнопка "Добавить команду" справа от набора команд. При ее нажатии команда, выбранная в окне доступных для добавления в интерфейс, попадет в конец списка команд группы (если в "Наборе команд интерфейса" была выбрана группа или входящая в ее состав команда) или в конец списка команд "Панели быстрого доступа" (если она или входящая в ее состав команда были выбраны в "Наборе команд интерфейса"):



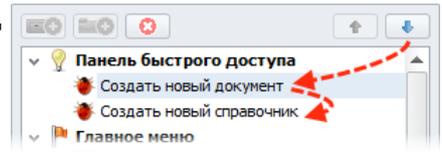
Также команду можно добавить в шаблон интерфейса двойным нажатием левой кнопки мыши.

- выбранную в "Наборе команд интерфейса" команду (10), группу (9) или закладку (8) можно переименовать. Для этого необходимо кликнуть по ней один раз левой кнопкой мыши и ввести новое название. Если нужная строка уже выбрана, можно сразу начать вводить новое название, оно заменит старое. Для сохранения названия следует нажать **Enter**, либо выбрать другой пункт меню. Для отмены переименования (до сохранения изменений) достаточно нажать клавишу **Esc** на клавиатуре:

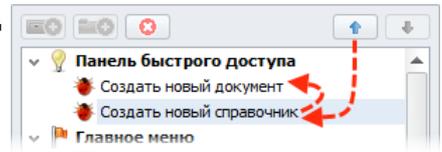


- кнопка (3) удаляет выбранную в "Наборе команд интерфейса" команду (10), группу (9) или закладку (8) со всем их содержимым;

- кнопка (4) перемещает выбранную в "Наборе команд интерфейса" команду (10), группу (9) или закладку (8) со всем их содержимым на уровень вниз;



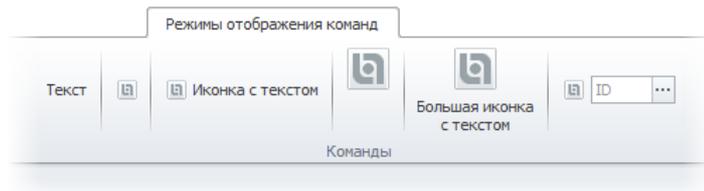
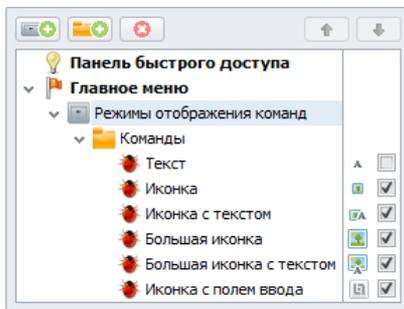
- кнопка (5) перемещает выбранную в "Наборе команд интерфейса" команду (10), группу (9) или закладку (8) со всем их содержимым на уровень вверх;



- иконка (11) напротив каждой команды (10) показывает режим ее отображения в главном меню. Сменить режим отображения можно двойным кликом левой кнопки мыши по иконке (11). Различаются следующие режимы:

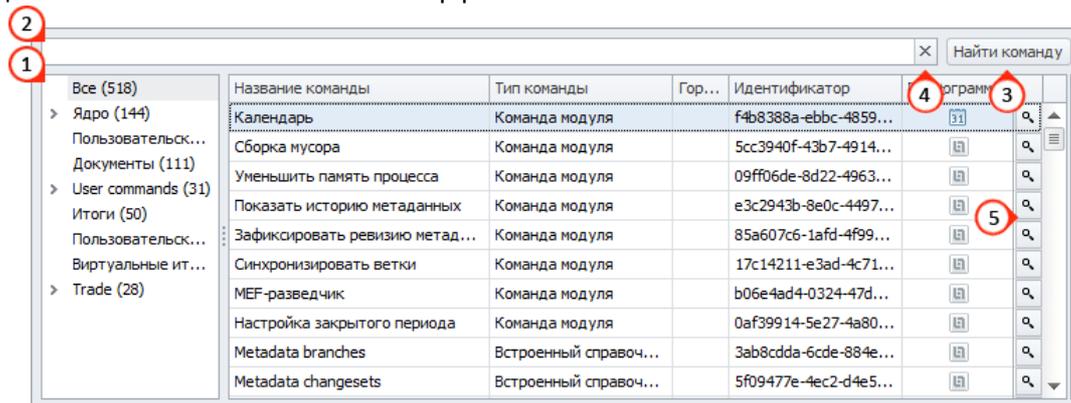
- ▲ – текст;
- 📄 – иконка;
- 📄📄 – иконка с текстом (справа);
- 📄📄📄 – большая иконка;
- 📄📄📄📄 – большая иконка с текстом (снизу);

иконка – маленькая иконка с полем ввода, позволяющим открыть списковую форму объектов или непосредственно объект (запись справочника или документ) по его идентификатору:



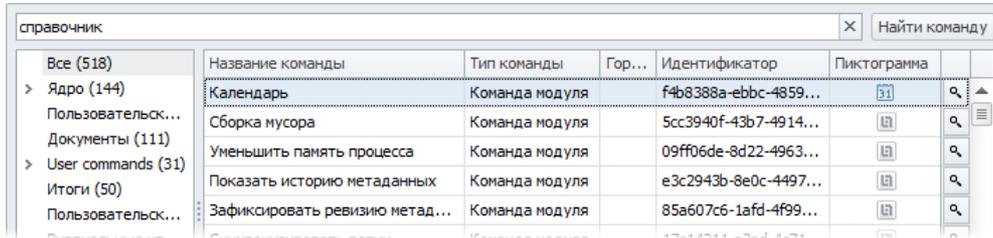
- флаг (12) напротив каждой команды (10) позволяет отделить ее вертикальной чертой от предыдущих. Установка флага для первой команды в группе (9) не принесет результата.

В окне "Команды для добавления в шаблон интерфейса" отображается список команд, которые можно добавить в пользовательский интерфейс:



- команды упорядочены по группам (в скобках отображается количество команд в группе) в древовидной структуре (1). Первый пункт "Все" показывает все доступные команды, выбор остальных групп ограничивает отображаемый список;

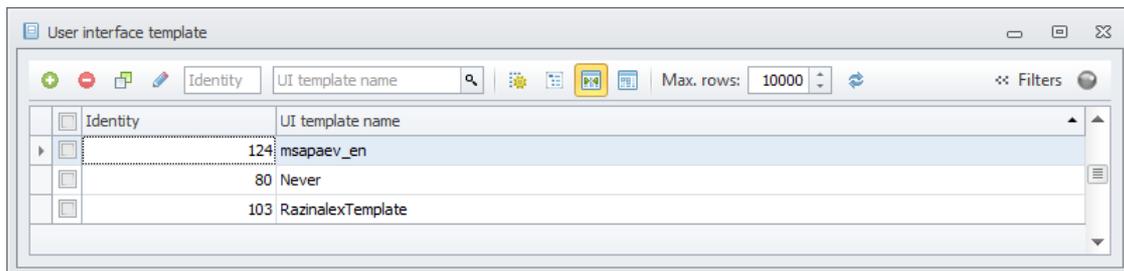
- в строке (2) можно осуществлять фильтрацию команд. Поиск будет вестись только среди содержимого выбранной в структуре (1) группы (при выборе пункта "Все найденные" – по всем командам). Результатом фильтрации будет список команд удовлетворяющих условию поиска. Для осуществления поиска необходимо ввести в строку фильтрации текст и нажать **Enter** или кнопку "Найти команду" (3). Например, введя в строке поиска "справочник", мы ограничим список команд восемью:



- сбросить фильтр можно кнопкой (4);
- с помощью кнопки [Поиск] (5) можно проверить была ли команда уже добавлена в шаблон интерфейса. У соответствующей ей команды в секции "Шаблон интерфейса" будет последовательно подсвечено каждое ее вхождение (при каждом новом нажатии кнопки (5) очередное вхождение), если она уже была добавлена в интерфейс ранее;
- при выборе команды в "Шаблоне интерфейса" (слева) она и ее группа будут выделены и в списке "Команды для добавления в шаблон интерфейса".

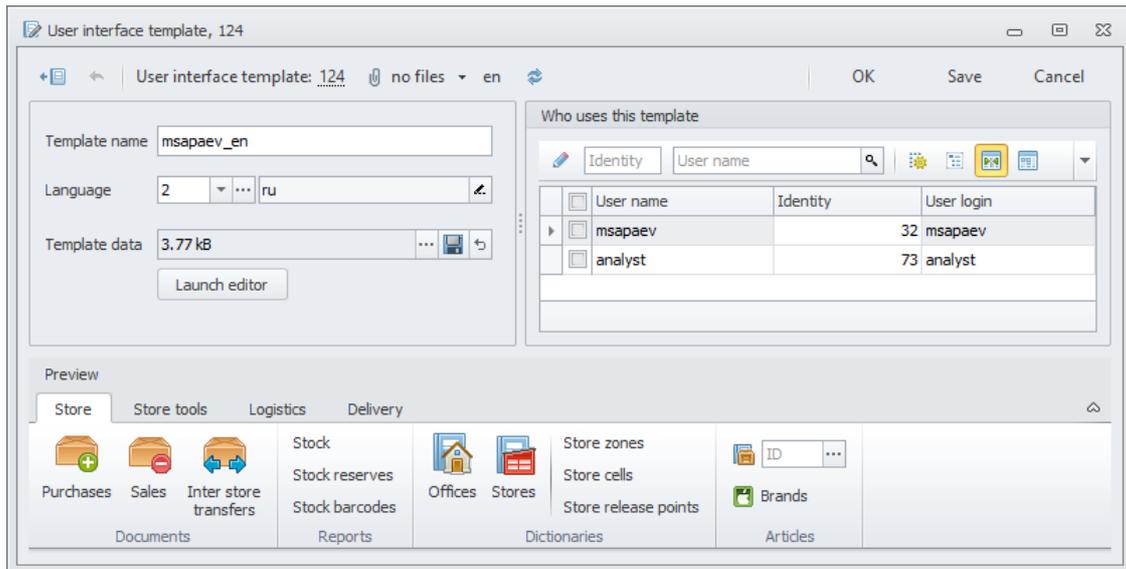
Интерфейсы пользователей

Все шаблоны интерфейсов пользователей хранятся в справочнике User interface templates:



Шаблоны можно фильтровать по *Названию (UI template name)*.

Форма редактирования шаблона позволяет задать следующие свойства:



- *Template name* – название шаблона;
- *Language* – язык, который используется в шаблоне. Выбор языка носит информационный характер и на сам интерфейс никак не влияет;
- *Template data* – непосредственно шаблон интерфейса. Открыть его в [визуальном редакторе](#) можно нажатием на кнопку "Launch editor";
- *Who uses this template* – список пользователей, которым назначен данный интерфейс. Список можно фильтровать по *Имени пользователя (User name)*;
- *Preview* – область предпросмотра шаблона.