ULTIMATE SOLID®

2016 R2

Administrator guide

ver. 5.7.0

www.ultimatesolid.com

Table of Contents:

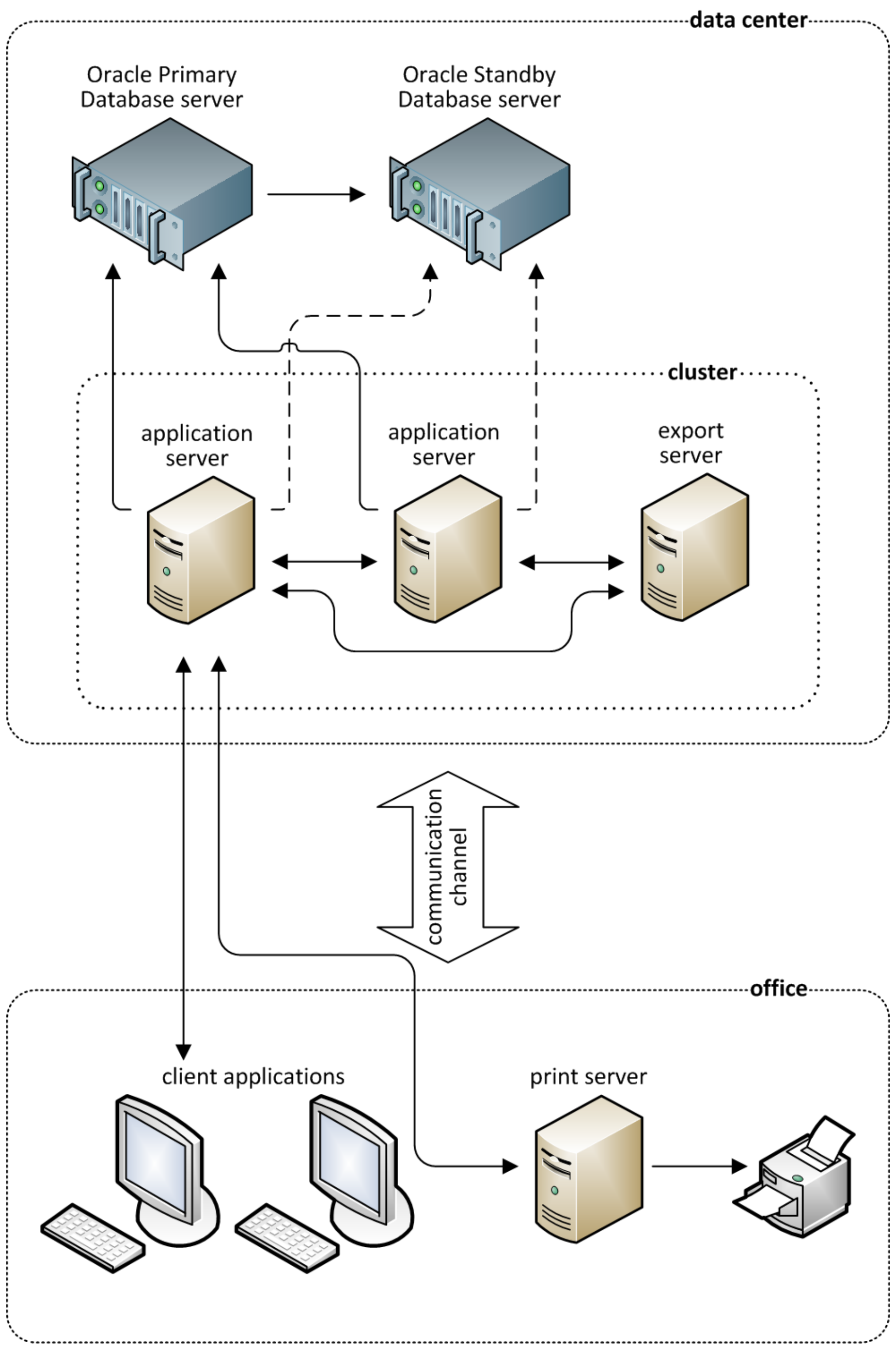
Introduction

Ultimate Solid Architecture

Ultimate Solid is a three–tier architecture software suite (wikipedia) which includes:

- database server (Oracle 11gR2 Enterprise Edition or Oracle 12c Enterprise Edition);
- application server;
- print server;
- client applications.

The general model of interaction is presented in the diagram below:



As a rule, and that is highly recommended to, the **database server** is located in the data center. To increase hardware fault tolerance and mitigate the risk of data loss, as well as to share the load, it is recommended to install a standby server/server cluster.

Based on similar considerations of fault tolerance and performance, the **applications server** which carries out processing of business logic and directly exchanges data flows with database server, can also be scaled into a cluster. Due to extremely high intensity of data exchange between the application server and the database server, they should be located in close proximity to each other, at least within the same local network.

As opposed to the application server and the database server, the **print server is located** at the same place with the printing devices and staff, thus reducing the load on the communications due to transferring smaller volumes of data.

With help of screen logic of **client applications,** the user can view, enter and edit data.

Composition logic of Ultimate Solid - based solutions

Operated by the customer intellectual managing system on a platformUltimate Solid, consists of two physically and logically different entities: platform and configuration.

PlatformUltimate Solid-**invariant system kernel** hereinafter the terms*platform*andkernelare used interchangeably), which remains invariable for any subject field and business logic. In other words, for each installation of the system kernel, it is also a platform - absolutely the same as for any other.

The platform ensures the functioning of the entire software system as a whole and provides the following tools and mechanisms:

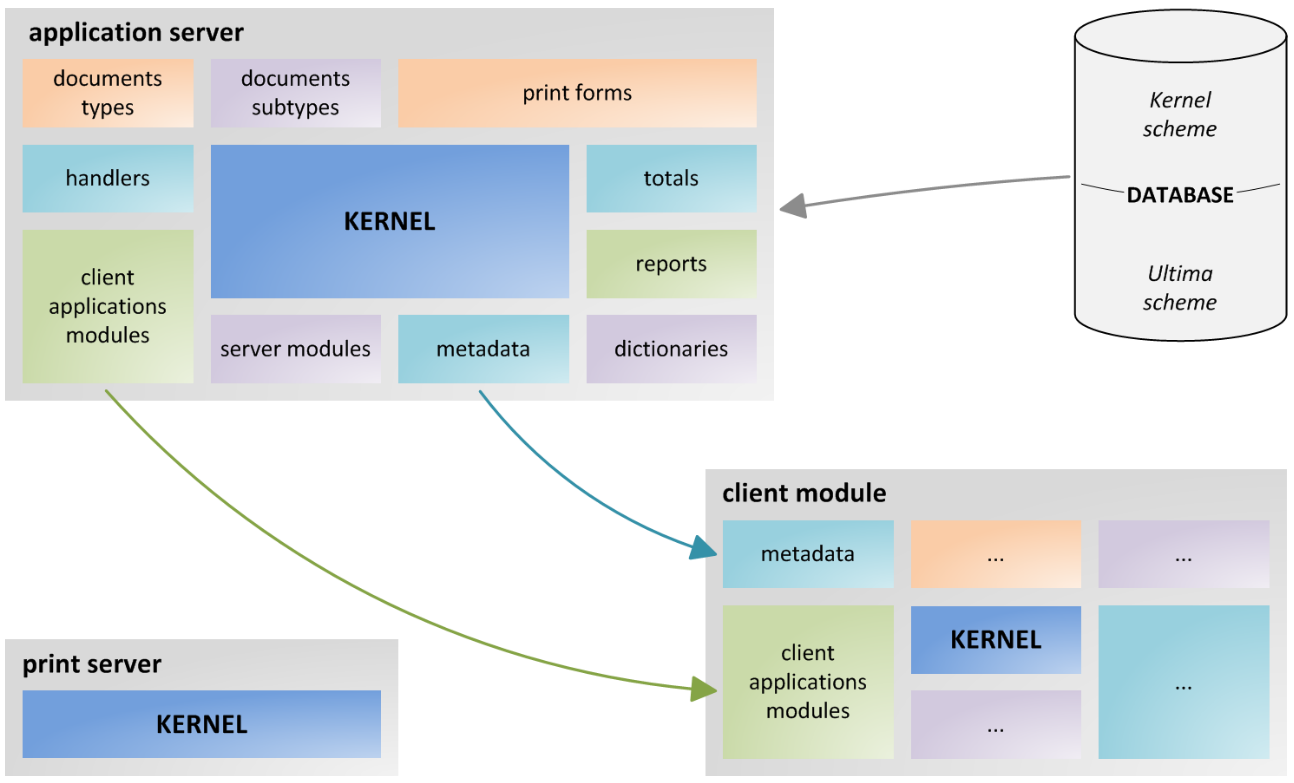
- Users storage , their authorization and entitlement inspection;
- communication between applications;
- access to the DBMS (database management system);
- management of configuration changes (versioning mechanism);
- conversion of “raw data” from DB to system objects;
- integration with other systems – SOAP, JSON, REST, XML, etc.;
- development environment and configurationchanges.

Configuration(for example, basic industry configuration Ultimate e-Trade and Ultimate e-Service) – **description of the subject area**. In essence, it is implementation of business logic and its information binding. A specific solution that works for a specific customer, even if it was built on the basis of the basic configuration, in 100% of cases has its own distinctive features, which are formed in the process of establishing the terms of reference, the process of implementation and operation of the embedded solutions.

Configuration - a set of meta data describing the business objects:

- documents;
- reference manuals;
- connection between reference manuals and documents;
- results;
- handlers, describing the logic of interaction between business objects;
- server modules;
- client modules;
- user rights.

The general model is presented on the chart:



The configuration is stored entirely in the database. As DBMS Oracle 11gR2 Enterprise Edition or Oracle 12c Enterprise Edition are used, which contain two schemes:

- *Kernel*– static, from this scheme the configuration is loaded when the application server starts;
- *Ultima*– application-oriented, in it tables are created and data of application area are stored.

The kernel is unavailable to be changed by the application-oriented developer, however it provides an interface and tools to implement business logic. The business logic performed on the application server is implemented in the form of classes on C# inherited from the corresponding classes of a platform (further such classes will be called scripts). The screen logic is carried out on the client application and is implemented in the form of modules on any .NET compatible language.

Administrator

Enter topic text here.

Hardware system

Database server

Oracle 11gR2 Enterprise Edition or Oracle 12c Enterprise Edition is used in Ultimate Solid. The administrator has to possess necessary skills of its administration. To reduce the risk of data loss and reduce possible downtime in the event of an accident on the main database server it is recommended to install one backup (StandBy) server at least. If the reserve servers are open for reading, they can be used to reduce the load on the main server. The database stores all the basic information. Any files on the application server or client devices or refer to the platform, or they are configuration files of platform applications, or they are either cache for data at the DB. Only the application server communicates with the server of the database, other parts of system do not require and should not have access to the DB.

Platform update

The platform update process implies the consecutive execution of the following actions:
1. create a database working copy;
2. using the working copy, carry out all the procedures (update and others) necessary to overcome changes that violate backward compatibility (described in details further in this section);
3. test the company's basic processes within the updated copy of the database;
4. proceed to update the main database only if no problems occurred on the previous stages;
5. stop working in the database (turn off all the application servers; for security reasons, you can change users' passwords to ensure an exclusive access);
6. backup the database;
7. carry out the update procedure and other associated actions; test the critical functions;
8. if problems occurred, backtrack the changes and recover the database from the backup copy; let the users work with the old version of the database. To solve the problems, use the copy (!) of the database and repeat the procedure starting at step 1 again.

The update procedure begins with the database update. It is performed by means of the utility (console-based application) ***VersionUpgrade.exe***, which is included into the Ultimate Solid distribution kit.

The utility settings are stored in ***VersionUpgrade.exe.config*** XML file (when first run, you can copy or rename the file *VersionUpgrade.exe.config.sample*). The settings that can be changed by the administrator are stored in *userSettings* section:

```
<userSettings>
       <Ultima.VersionUpgrade.Settings>
              <setting name="AppServerID" serializeAs="String">
                     <value>1</value>
              </setting>
              <setting name="UserID" serializeAs="String">
                     <value>1</value>
              f</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>
              f</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>
              f</setting>
       </Ultima.Server.Properties.Settings>
</userSettings>
```

AppServerID – application server ID.

UserID – ID of the user, in whose name the data are being changed.

The user *UserID* should possess the rights of developer (the role possesses the *Developer* right).

ConnectionString – string containing settings for the access to the database server for the user *kernel*.

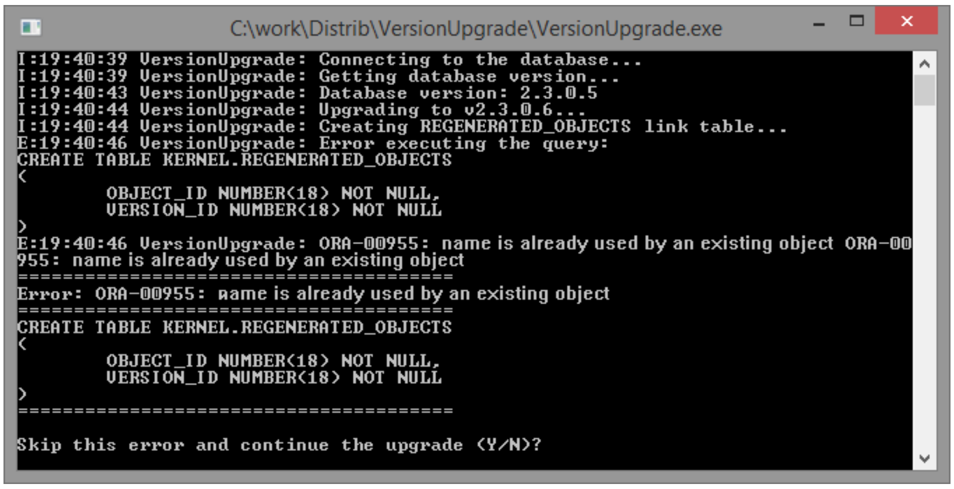
SysConnectionString — connection string meant for a user with the admin rights, who can grant the user *kernel* new rights.

To update the database, one needs to:

- edit the configuration file *VersionUpgrade.exe.config*:
  - specify values for the parameters *HOST*, *SERVICE_NAME* and *PASSWORD* in the sections *ConnectionString* and *SysConnectionString* ;
  - make sure that the application server specified in the section *AppServerID* is working on the configuration branch marked with the tag *Default*. It is inadmissible to start the database update procedure on any other tag (!);
- launch the utility *VersionUpgrade.exe*.

The database update process is carried out incrementally. In other words, if the update is from version 2.3.0.6 to version 2.3.0.9, the process will run through all the steps: first, to version 2.3.0.7, then 2.3.0.8, and at last 2.3.0.9.

The log of the update process is located in the console, in which the utility was launched. If the database was cloned from an intermediate release, there may occur noncritical errors, e. g., an attempt to create an already existing table. Every time such error occurs the update utility provides the user an opportunity to choose whether to abort or continue the update process. If the query text gives to understand that the error can be skipped, the process should be continued. If any doubt exists, consult the system department:



```
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: name is already used by an existing object ORA-00
955: name is already used by an existing object
=======================================
Error: ORA-00955: name is already used by an existing object
=======================================
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)?
```

After the database is updated, proceed to update the platform:

1. backup platform's current binary files;
2. overwrite the platform files on the application servers with the copied binary files; If necessary (e. g., if the first (major) version of the platform was modified), recompile client modules and metadata;
3. run the new application server and the client application. When starting, the server or the client application may report on inability to load handlers or client modules; if so, disregard such reports;
4. execute Developer -> Compile -> Regenerate metadata classes on each branch used. One should not (!) do this on ordinary non-branch tags;
5. execute Developer -> Compile -> Compile & Reload metadata on each branch used;
6. recompile the client modules with the new metadata version;
7. restart the server and the client application;
8. update is complete.

After that, it is needed to overcome changes (breaking changes) that violate backward compatibility. The detailed information is provided in the corresponding section of the developer's documentation.

Licensing

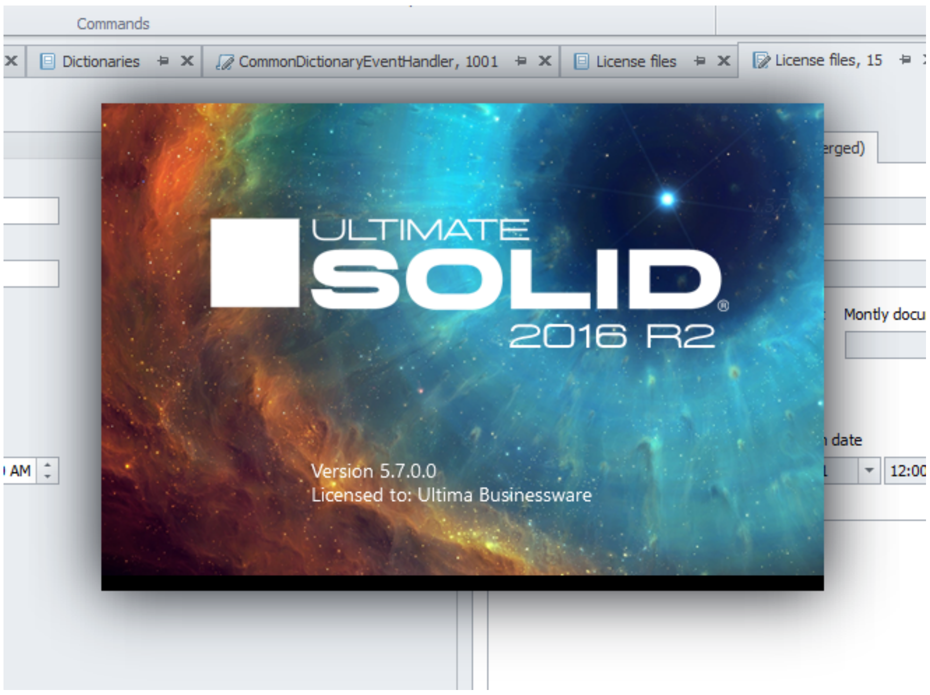
For work with application serverUltimate Solid you should have the license. License is the text with the name and the key of company owner and the parameter list of licensing which was certified by digital signature of affiliate center. The parameters of licensing set the kit of restrictions for the operational characteristics of system:

- Amount of the synchronous user sessions;
- Amount of the synchronous sessions of developers;
- The limit for the amount of created documents (in re-count for the user);
- The list of the web service accounts, which can have the unlimited number of sessions;
- The date of license grant;
- The date of beginning of licensing period (often coincides with the date of license grant);
- The expiry date of licensing period (only for the temporary license).

After the purchase of license, it must be downloaded to the dictionary of licenses. For this it rather to put under the license in the work catalog of applications server so as to load automatically at start up. After the load in dictionary, license will be available for the all services of applications which work with the same database.

Companies, which got the license for using platform, get the unique key identifier for marking all files of license, which were granted further for this company. The key of company is determined by license, which is marked in dictionary as the main. In every moment of time it can be active the several files of license. The number of users and developers, which are listed in all acting licenses herewith they are summed, the lists of unlimited web-users are united, the limit of document number of user are selected maximal. The second load of the same license files in dictionary doesn't bring to the increase of number of permitted users because duplicates are not counted.

All licenses are active if their company key will coincide with the company key of the main license and if their period of validity won't end. The last issued license give the name to company-licensee so if you want to change the name, it will rather to get the license with new name of company (the key of company stays the same). If the configuration requires the increase of characteristics outward the given restrictions in license file (for example, with increasing of users which work in the same time), it will be necessary to purchase and load the new license. You can control the loaded licenses in the dictionary of license files. The name of company-owner is displayed on the window “about the program”, at the bottom:



The control of document limits and terminal session

During operation the system measures all parameters, which are limited by license, automatically: The number of concurrent sessions of users and created documents.

The sessions of users can be of two types: Native connections and web-connections. Native connections consider common users and user-developer singly and they always are recorded by the number. Web-connection let two variants of accounting: Possessional (quantitative) and unlimited. By default it uses possessional scheme. Unlimited web-connections are available only for users whose records are listed in active license. Unlimited web-connections always are used for the internet-shop, for the integration module with Bitrix and etc.

For the action the unlimited regime it is required to have the login and the password of web-user coincide with specified on execution of license file login and password completely (includes the case). When web-user authorizes, the password of unlimited account is checked by license file, not by dictionary of web-users.

Applications, which use the unlimited scheme, must hide the login and the password of access (for example, by encryption of configuration file) and guarantee the inability of their change.

Applications, which use the possessional scheme, must contact with server at least 2 times in 1 minute. The connection would active if the last access had been no more than 2,5 minutes ago.

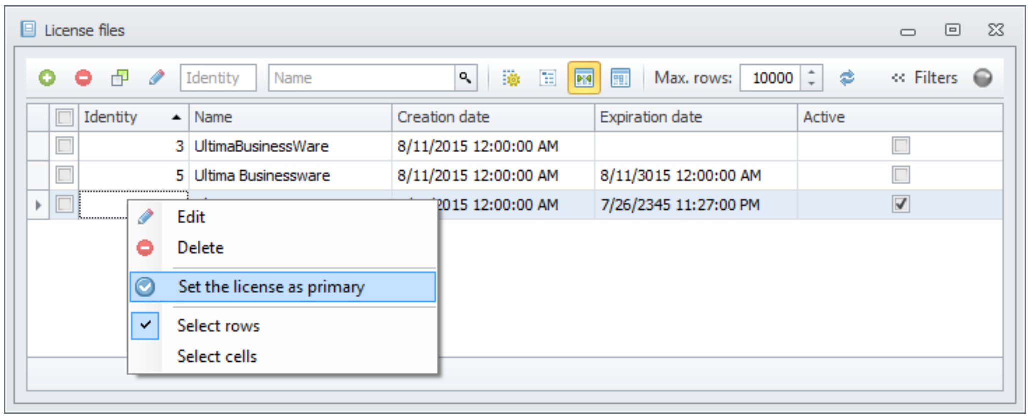
The control of number of synchronous sessions happens at the entrance by user in system or at the recovery of inactive session (for example, when employee starts to work after the lunch break). In case the limit is exceed, the sign-in to the system will be limited till any session will get free.

Developers sessions are separated from users (developer is the user had Developer allowance). If the number of developers, which are working simultaneously, had exceed the established by license limit, developer would can use the unoccupied session of common user. Inverse is wrong.

Number accounting of the created documents works in the following way: It accounted the last 90 days of system work, which were broke down into three periods for 30 days. The number of created documents is calculated in every period. If the limit to the number of documents had exceeded in the first two periods, the system would give a warning. If the limit is exceeded in the all three periods, the system will block the created of new documents. The re-count of the number of created documents happens at the start of server and then once a day.

The dictionary of license files

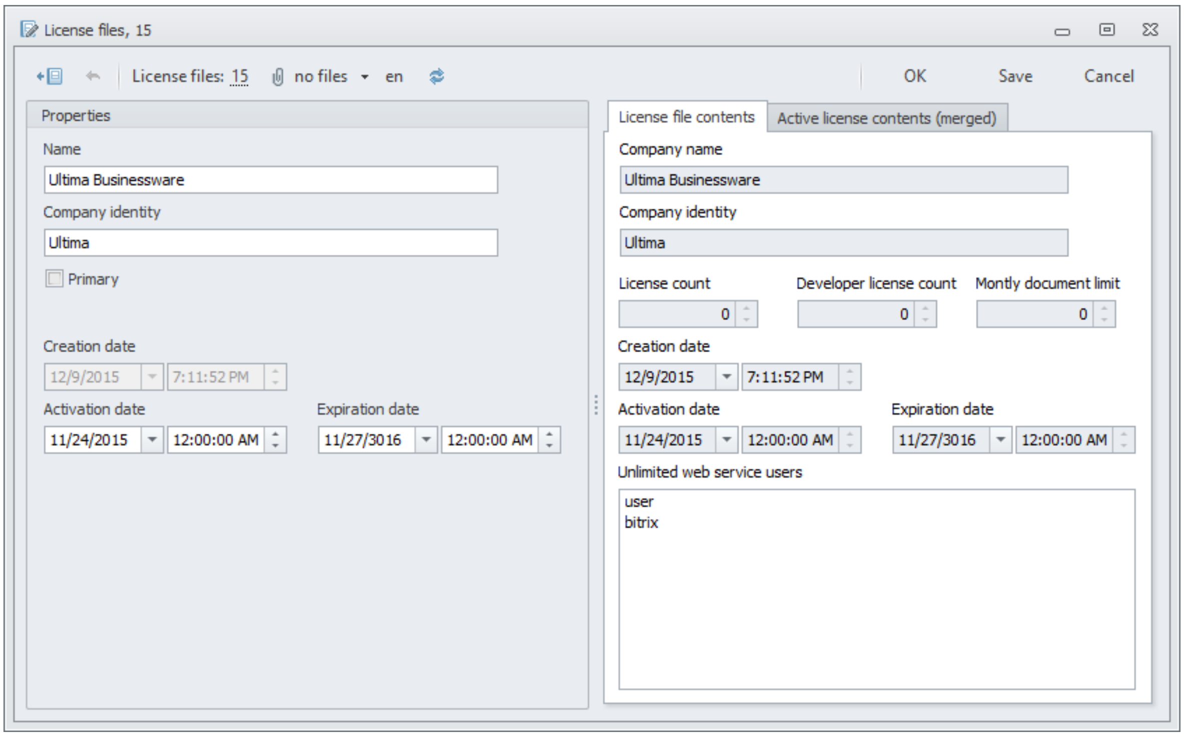
In the license dictionary you can see the downloaded licenses, download the new licenses and choose the primary.


License files

Identity | Name | Max. rows: 10000 | Filters

Identity	Name	Creation date	Expiration date	Active
3	UltimaBusinessWare	8/11/2015 12:00:00 AM		☐
5	Ultima Businessware	8/11/2015 12:00:00 AM	8/11/3015 12:00:00 AM	☐
		2015 12:00:00 AM	7/26/2345 11:27:00 PM	✓

The previous primary license loses the status when you mark the primary license by dint of the command of shortcut menu. The license dictionary let see the content of licenses, the date of issue, the parameters and the lists of unlimited web-users, but it is not let change the parameters of licensing.


License files, 15

License files: 15 | no files | en | OK | Save | Cancel

Properties

Name: Ultima Businessware

Company identity: Ultima

☐ Primary

Creation date: 12/9/2015 7:11:52 PM

Activation date: 11/24/2015 12:00:00 AM

Expiration date: 11/27/3016 12:00:00 AM

License file contents | Active license contents (merged)

Company name: Ultima Businessware

Company identity: Ultima

License count: 0

Developer license count: 0

Montly document limit: 0

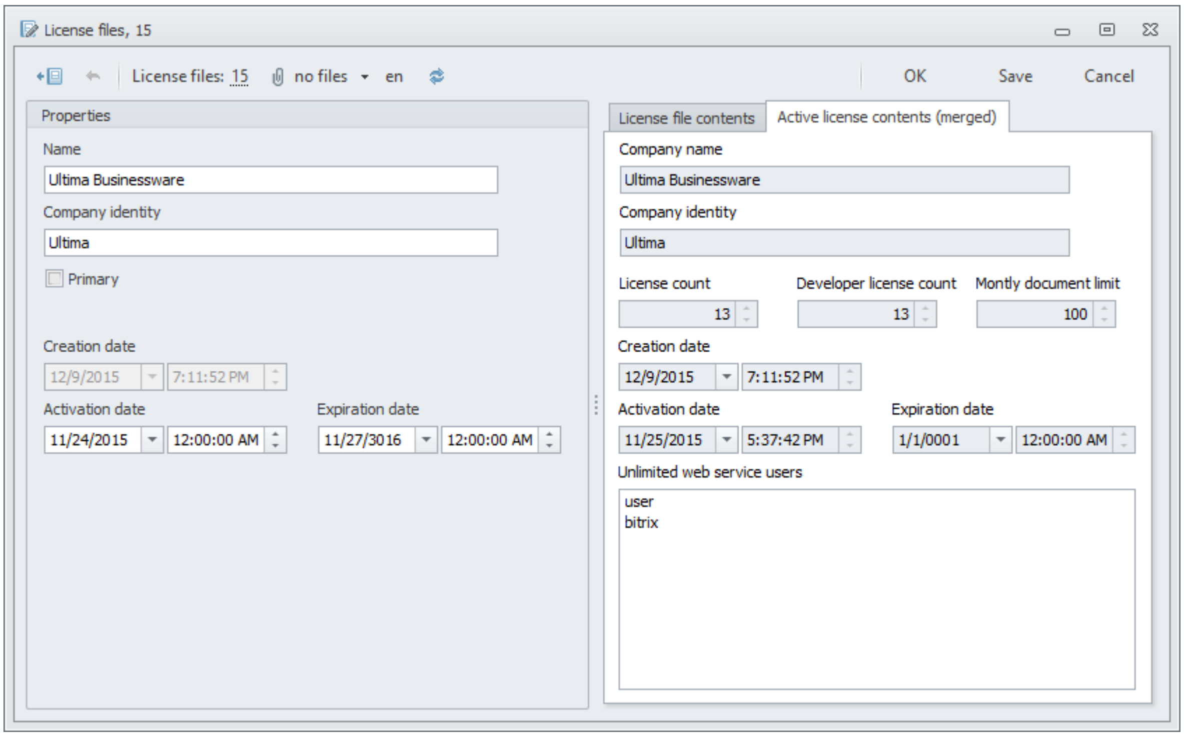
Creation date: 12/9/2015 7:11:52 PM

Activation date: 11/24/2015 12:00:00 AM

Expiration date: 11/27/3016 12:00:00 AM

Unlimited web service users:
user
bitrix

in the tab “Active license contents (merged)” you can see the current restrictions of united license, which include all active licenses of company at the moment.


License files, 15

License files: 15 | no files | en | OK | Save | Cancel

Properties

Name: Ultima Businessware

Company identity: Ultima

Primary

Creation date: 12/9/2015 7:11:52 PM

Activation date: 11/24/2015 12:00:00 AM

Expiration date: 11/27/3016 12:00:00 AM

License file contents | Active license contents (merged)

Company name: Ultima Businessware

Company identity: Ultima

License count: 13

Developer license count: 13

Montly document limit: 100

Creation date: 12/9/2015 7:11:52 PM

Activation date: 11/25/2015 5:37:42 PM

Expiration date: 1/1/0001 12:00:00 AM

Unlimited web service users:
user
bitrix

Application server

To balance the load and reduce downtime in case of hardware failure, the application server can be installed on several servers.

The application server can be launched as **a console–based application** – ***ConsoleServer.exe*** (used predominantly by developers) or install it as a **service**, running the same file with the install key (with the administrator permissions) – UltimaService.exe install.

To launch the server as a console-based application, use the administrator permissions.

The following command line options are supported: ConsoleServer.exe [**key**] [-option:value] [-switch]:

- **run** – run as a console-based application (used by default when launching the application without key);
- **help** – show help;
- **install** – install service:
  - -username – user name for running the service;
  - -password – user's password;
  - -instance – name of the service instance if installed more than once;
  - --autostart – the service must be started automatically (default option);
  - --disabled – the service must be installed with disabled status;
  - --manual – the service must be started manually;
  - --delayed – the service must be started automatically (with a delay);
  - --localservice – start the service with the account of the local service;
  - --networkservice – launch the service with the permissions of a network service;
  - --interactive – the service will address the user during installation for the authorizations of the service;
  - --sudo – inquiries to UAC, if run on Vista/W7/2008;

  - -servicename – name of service, which must be used during installation. By default, the service name is ***UltimaService***;
  - -description – description of service, which must be used during installation;
  - -displayname – the displayed service name, which must be used during installation;
- **start** – start the service, if not already started:
  - -instance – instance of the service to be started;
- **stop** – stop the service, if started:
  - -instance – instance of the service to be stopped;
- **uninstall** – uninstall the service;
  - -servicename – name of the service;
  - -instance – name of the service instance if installed more than once;
  - --sudo – inquiries to UAC, if run on Vista/W7/2008;
- characteristics affecting the start of the application server as a console-based app:
  - -AppServerID:N – code of N application server (integer-valued, no default value);
  - -TcpPort:N — TCP port Number (integer-valued, no default value);
  - -DuplexProtocol:true/false — use either duplex or simplex protocol (true by default);
  - -UseManagedDataProvider:true/false — use managed data provider (true by default);
  - --SkipLoadingMetadata — skip loading metadata (if damaged).

If the service has been renamed (servicename) or a service instance name has been specified when installing, these parameters will need to be provided for each action within the service – start, stop, or removing.

Virtually all settings of the application server (and cluster) are stored on the database server. The configuration of the cluster in the minimum volume includes the list of servers. Only the options of launching the server module are customized on the application server itself. They are stored in an XML file (standard for any .NET application) ***ConsoleServer.exe.config***.

The settings which can be changed by the administrator are stored in *userSettings* section:

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

ServiceHostName – application server name; the name will be displayed in the list of services.

TcpPort – a port used by other servers and client apps to obtain access to the given application server.

***UseManagedDataProvider* —** use either managed version of ODP.NET (Oracle Data Provider Managed Driver) or a general version, which requires installation of Oracle Client. Using the managed driver is more preferable, as it requires no administration.

ConnectionString – string containing database server access settings.

When using an embedded driver, the following options are available:

- *Data Source* or *Server* or *Host* – name of the server of the Oracle database which is connected to;
- *User ID* or *User* – Oracle account;
- *Password* – password for Oracle account;
- *Service Name* – name of the Oracle database instance;
- *SID* – unique name of the Oracle database instance;
- *Statement Cache Size* – enables or disables caching of operators. The value defines the maximum number of operators which can be cached for connection. By default, the value of this attribute is *0* (off). Caching of operators starts when the parameter value is more than *0*. It cannot be more than MAX_OPEN_CURSORS parameter in Oracle database;
- *Connection Timeout* – the timeout (in seconds) when attempting to connect before completing the attempt and generating an error. Value *0* means absence of limit. The default value is *15* (sec);
- *Pooling* – if *true*, connection pooling is used. The connection pooling helps reduce database access time. The default value is *true*;
- *Min Pool Size* – minimum number of connections in the pool. The default value is *0*;
- *Max Pool Size* – the maximum number of connections in the pool affect performance. The default value is *100*;
- *Connection Lifetime* – life time of the connection. When the connection returns to the pool, its creation time is compared with the current time, and the connection will be closed if such interval (in seconds) exceeds the value set in *Connection Lifetime*. The default value is 0;
- *Default Command Timeout* – the timeout (in seconds) when attempting to execute a command before completing the attempt and generating an error. Value 0 means absence of limit;
- *Validate Connection* – determines if to check the connections received from the pool;
- *Enlist* – toggles on (if *true*) or off automatic attachment to a promotable transaction in the existing transactions;

- *Promotable Transaction* – indicates if the transaction is local (if *local*) or promotable (if *promotable*) throughout its lifetime.

When using the unmanaged version of ODP.NET, the following parameters can be specified:

- *Home* – the value of Oracle Home which will be used;
- *Oci Session Pooling* – if *true*, makes available the functions of the OCI session pool (OCI Session Pooling);
- *Oci Session Pool Allow Waiting* – if *true*, new connections will wait until the old ones are closed, if the value set by *Oci Session Pool Max Size* is reached; if *false*, new connections will be declined;
- *Oci Session Pool Connection Lifetime* – determines the life time of the connection in seconds. Before returning the connection back to the to pool, its life time is verified. If the life time of the connection exceeds the value of that property, the connection is closed and deleted. If the value of this parameter is 0, the life time of the connection is never verified;
- *Oci Session Pool Increment* – allows the applications to set the following increment for sessions, which will be launched, if the current number of sessions is lower than *Oci Session Pool Max Size*. Permissible values: 1 and greater;
- *Oci Session Pool Max Size* – determines the maximum number of sessions, which can be opened in the sessions pool. As soon as such value is reached, no new sessions will be opened. Permissible values: 1 and greater;
- *Oci Session Pool Min Size* – sets the minimum number of sessions in the session pool. Such number of sessions will be started initially. After that, the sessions will be opened only as necessary;
- *Oci Session Pool Password* – determines the password for proxy user if installed;
- *Oci Session Pool User Id* – determines the login for proxy user if installed. Available only when the parameter of *OCI Session Pooling* is set to *true*.

DuplexProtocol – use the built–in channel of the driver for the connection with client applications or the standard .NET Remoting; has the following values:

- *true* – duplex channel is used (recommended value);
- *false* – standard .NET Remoting channel is used.

AssembliesFolderName – name of the folder, where server modules are loaded.

AppServerID – identifier of the applications server, coincides with the identifier of the above application server in Ultimate Solid system (for details, see Servers and clusters).

ScriptCacheFolderName – name of the folder, where scripts are cached.

LicenseFileName – name of the license file;

TaskStatSyncInterval – interval (in ms) of polling the DB server by task scheduler;

TaskExecutorLogin – name of the user (of Ultimate Solid system), under which the task will be executed;

TaskExecutorPassword – password for the login specified;

TaskSchedulerActive – if the scheduler is running (*True* – yes, *False* – no) on this application server. The tasks must be executed only on one application server (cluster) of the system.

Setting up the computer intended for functioning of the application server requires the following actions:

- install a Windows family operating system on the computer;
- The OS user, who launches the application server, must possess admin rights.
- install .NET Framework platform, version 4.5 or greater;
- copy the AppServer distribution package;
- edit the configuration file *UltimaService.exe.config*;
- Additional setting of the application server are carried out in the client app's administrator module Ultimate Solid and described in Servers and clusters section.

Several services of the application server may be installed on one server. Each service must have its own settings file *.exe.config. The settings file shares the same name with the application exe. file, therefore a number of exe. files are needed. The services may be installed in separate folders, or just copy the file ConsoleServer.exe within the same directory. Each server file will have its corresponding configuration file:

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

It is needed to specify separate names of instances for each service of the server. For that purpose, the instance option of the command line is used:

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

To start, stop or uninstall the service, it is necessary to specify the instance:

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

By default, the service name is ***UltimaService***; The name can be changed by servicename option of the command line. This option can be combined with the "instance" parameter:

```
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
```

Transaction timeout

During long (more than ten minutes) operations the server can give the following error message:

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

This is due to the fact that the transaction (the classes from namespace *System.Transactions*, appeared in .NET 2.0) by default have the maximum timeout equal 10 minutes. When starting the application server checks the system settings and gives a warning about this:

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

To solve the problem, it is necessary to edit the file *machine.config*, which can be found in the folder:

```
%WINDIR%\Microsoft.NET\Framework64\%version number%\Config
```

Set-up, remove the limit for the maximum duration of a transaction, looks as follows:

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

Print server

It accept the documents for the print from applications servers which are situated in the data center.. As a rule, the server of print is situated in office, where it is used the client applications – it let substantially reduce the load on channels of connections at the expense of transmission of much less amount of data.

The using of internal mechanisms of print let provide the guaranteed delivery of print form after the printer, the centralized journal entry (who, when, what, how many and where published) and other functions.

The server, on which it is loaded the print server, and net, in which it situated, must be tuned for establish the connection on print server from applications server and establish the connection for the everyone of the applications server from print server.

The applications server caches the patters of print forms for the reducing the burden on the data channel. If it is the active configuration development, with which the patterns often change, you will be able to watch a significant increase of the cache size. Cache may be deleted at any time, it will start to fill one more time.

The print server supports the print of the document package. The document package will be printed on the one printer in the order, no one of other documents won't be printed when the document package is printed (on condition that no one hasn't the access to printer besides the print server).

The print server may be launched as the service and as the console application.

Print server setting ***PrintServer.exe*** in file ***PrintServer.exe.config*** in the section *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=”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="UseTemplatesCache" serializeAs="String">
                        <value>LocalRepository\TasksRepositories</value>
                </setting>
                <setting name=”IemplatesRespositoryPath" 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 – address of applications server. This server operates for receiving the patterns, saving of statistic and etc. Any known good server will do.

PrintServerID –print server ID in system.

AdditionalThreadsCount – stream available amount for tasks. It is not recommended, the number of additional streams exceeds the number of virtual computer core, which print server works on.

AppServerLogin – – login for connecting to the application server. It is recommended to use the initially logged in system user of *PrintServer*.

AppServerPassword – – password for connecting to the application server.

PrintWorkerApplicationPath – path to application Print Worker.

TcpPort – the port the print server works on.

TaskBuildingTimeOut – is the maximum time of waiting for building the print form (the rendering) in seconds.

PagePrintingTimeOut –is the time in seconds for the printing of one page. It is recommended to establish a little more time for printing of one page by the slowest printer on the network.

RetryCount – is the number of the attempt to print tasks in case of error. After reaching the number of failed attempts, which are specified by parameter, the task is deleted from call.

UseTemplatesCache – caching the templates of printed forms, has the following values:

- *true* – the templates are cached locally and when used repeatedly, the application server is not accessed. Caching helps reduce traffic between servers and reduce the load on the application server.

When the print form inside the configuration version is changed, the cache is not reset. Such function can be used only with a fixed version of the configuration.

- *false* – templates are loaded every time from the application server. Such possibility is necessary for developers, or if the cluster (application server) works the unfixed version of the configuration changes into which can be made in real time.

TasksRepositoriesPath – path to the folder with print job cash.

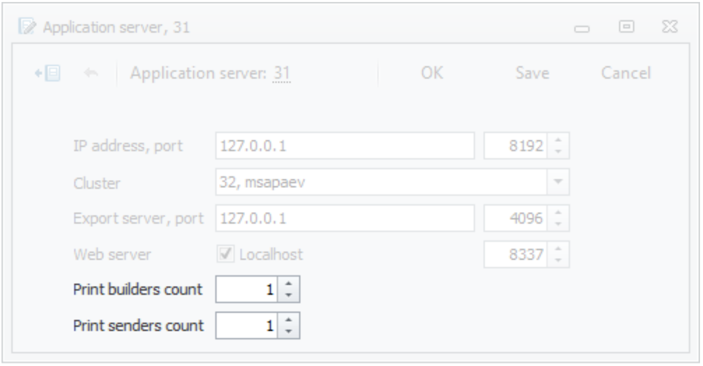
TemplatesRepositoryPath – path to the folder with plates pattern cash.

Setting up the computer meant for functioning of the printing server requires the following actions:

- install a Windows family operating system on the computer;
- switch off dispatcher service of print order (Print Spooler);
- install .NET Framework platform having version at least 4.5;
- install and connect the printers to the print server;
- wireless printers, used in single-rank net, also are need to connect to the print server;
- copy the distributive PrintServe to the print server;
- In dictionary Printers applicationUltimate Solid add the print server, stated IP address of setting computer and port, on which the PrintServer will operates ;
- add in this dictionary for created print server, the printers which installed in the same way in this computer. Printer system name *System name* must coincide with the name of printer in the operating system of setting computer;
- Issue to users permissionto print on created printers ;
- To edit the configuration file of the print server *PrintServer.exe.config*, stated the print server ID (ID by created print server in the system), port, on which the print server will work (similarly indicated previously in the system) and other parameters;
- To make sure that in the configuration fail *ConsoleServer.exe.config* at least one apps server of cluster as the activated task scheduler (tasks are sent on print by system task):

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

- Also in configuration at least one of cluster application servers must be stated for one stream for operating (*Print builders count*) and dispatching (*Print senders count*) print tasks:


Application server, 31

Application server: 31 | OK | Save | Cancel

IP address, port	127.0.0.1	8192
Cluster	32, msapaev	
Export server, port	127.0.0.1	4096
Web server	✓ Localhost	8337
Print builders count	1	
Print senders count	1	

Printing process described in details in the chapter print process description.

Export server

The export server is used to convert print forms to files.

The address of the export server is specified in the cluster configuration. In addition, the address can be specified in the configuration of a certain application server in case it is necessary for such application server to use a separate export server (for the application server which forms part of the cluster, they will use the export server specified in the configuration of that application server).

The export server uses system-dependent components and can be executed only in the operating system where .NET FrameWork platform is installed. It is recommended to locate the application server and the export server in the same (local) network to reduce the load on the communications channel, because a considerable volume of data will be transferred between them.

The export server can be installed as **a service** – ***UltimaExportServer.exe*** or **a console application** – ***ExportConsoleServer.exe***. The parameters of their configurations are identical and are stored in the relevant files:

- 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="UseTemplatesCache" serializeAs="String">
			<value>True</value>
		</setting>
	</Ultima.Exports.ExportSettings>
</userSettings>
```

TcpPort – the port the print server works on.

AppServerUrl – URL of the application server for loading templates of print forms. Any known good server will do..

AppServerLogin – login for connecting to the application server. It is recommended to use the initially logged in system user of *ExportServer*.

AppServerPassword – password for connecting to the application server.

UseTemplatesCache – caching the templates of printed forms, has the following values:

- *true* – the templates are cached locally and when used repeatedly, the application server is not accessed. Caching helps reduce traffic between servers and reduce the load on the application server.

  When the print form inside the configuration version is changed, the cache is not reset. Such function can be used only with a fixed version of the configuration.

- *false* – templates are loaded every time from the application server. Such possibility is necessary for developers, or if the cluster (application server) works the unfixed version of the configuration changes into which can be made in real time.

Setting up the computer meant for functioning of the export server requires the following actions:

- install a Windows family operating system on the computer;
- install .NET Framework platform having version at least 4.5;
- copy the ExportServer distribution package;
- edit the export server configuration file *ExportServer.exe.config*, specifying the port on which it will be working and other parameters.
- In the Clustersdictionary of the Ultimate Solid application specify the server of export to the configuration of cluster(s) and/or application server(s) specifying the IP address of the setup computer and the port specified before in the configuration *ExportServer.exe.config*.

Client application

The settings of the client application ***ClientLoader.exe*** are made in the file ***ClientLoader.exe.config*** of the section *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="ThreadingAlerts" serializeAs="String">
                     <value>True</value>
              </setting>
              <setting name="ScreenDpiCheckEnabled" serializeAs="String">
                     <value>True</value>
              </setting>
              <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 — selects a user authorization method when the application is launched; has the following values:

- *interactive* — graphical interface to enter login/password;
- *automatic* — fully automatic logging in.

Login — login for the automatic logging in to the application.

Password — password for the automatic logging in.

AutomaticServerSelection — determines the possibility of selection of an application server when starting the application:

- *true* — user is not allowed to select an application server on his own;
- *false* — user is allowed to select an application server on his own.

ModuleCacheFolderName — folder for storing client application modules.

MapCacheFolder — folder for storing the map cache.

ClientApplicationID — client application code (the list of client applications is stored in the database). The code of the standard modular client application Windows Forms is 1. Examples of other applications: delivery notes print terminal, release monitor, etc. When designing client modules as usual, there is no need to update this setting.

UpdateModules — loading of client application modules; has the following values:

- *true* — when a client application is launched, its modules are loaded from the application server and saved into a folder set by the parameter *ModuleCacheFolderName*;
- *false* — modules are loaded locally from the folder set by the parameter *ModuleCacheFolderName.*

DuplexProtocol — link to an application server, which must be the same as for the application server; has the following values:

- *true* — duplex channel is used (recommended);
- *false* — standard .NET Remoting channel is used.

ThreadingAlerts — produce alerts, when synchronous remote method invocations are executed in a client application. The setting is useful, in the first place, for application programmers and users, who are working with the test configuration:

- *true* — alerts produced;
- *false* — no alerts produced.

ScreenDpiCheckEnabled —check DPI of the screen when running an application.

- *true* — check system settings and stop operations if unsupported DPI settings detected;
- *false* — skip the check.

StartupCommands — list of commands to be run when starting a client application (GUIDs for module commands separated by commas). As a command run at the system start, may serve, e. g., a most frequently used document register (or a number of such registers), the welcome screen, or a tip of the day.

ServerAddresses — list of application servers that the client application can work with. Specified by the array *ArrayOfServerAddress*, which includes a single or several elements *ServerAddress* with the following parameters:

- *Host* – application server IP address;
- *Port* – application server port;
- *DuplexProtocol* – link to an application server similar to the described above;
- *Quality* – weight determining the probability that the client module is connected exactly with this application server, if there is more than one application server in the array *ArrayOfServerAddress*. The probability is calculated as a ratio of the weight of the given server to the total weight of all servers. For example, there are three application servers in the array; the *Quality* of the first two servers is equal to "1", the third's is to "2". In this case, the client module, when connected to the application server, will choose one of the first two servers with a 25% chance of happening; the last one will have a 50% probability of being chosen.

Setting up the computer meant for the client application requires the following actions:

- to install a Windows operating system;
- to install a .NET Framework platform, version 4.6 or greater;
- to copy the Client distribution package;
- edit the configuration file *ClientLoader.exe.config*.
- it is strongly recommended to run the client application via the *ClientUpdater.exe* update utility.

Update of the client application

The client application is upgrading using utility ***ClientUpdater.exe*** , which forms part of Ultimate Solid distributive and is in the same folder with the client application.

The settings of the utility as stored in ***ClientUpdater.exe.config*** XML file (when the utility is launched for the first time, you can copy or rename the file *ClientUpdater.exe.config.default*). The settings which can be changed by the administrator are stored in *appSettings* section:

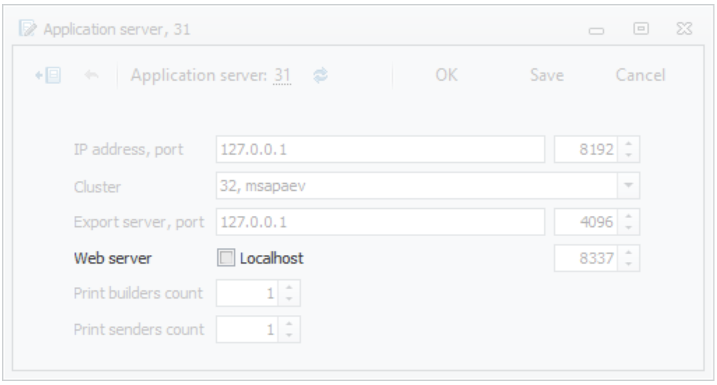
```
<appSettings>
       <add key="UpdateServiceUrl" value="http://localhost:8337" />
       <add key="WindowShowingDelay" value="1000" />
</appSettings>
```

UpdateServiceUrl – address of the application server where the updates are uploaded. The web-server must be necessarily setup on the application server. The port of the web server shall be specified as the port.

WindowShowingDelay – the time of showing the upgrade utility in milliseconds in case there is no need to upgrade the client application (absence of upgrades).

For the utility update operate of the client application it is necessary:

- The following is necessary for the work of the client application upgrade utility:
  - on the application server which acts as an upgrade server: By default it is running locally. That it was available externally it is necessary to clear *Localhost* flag in the application server settings:

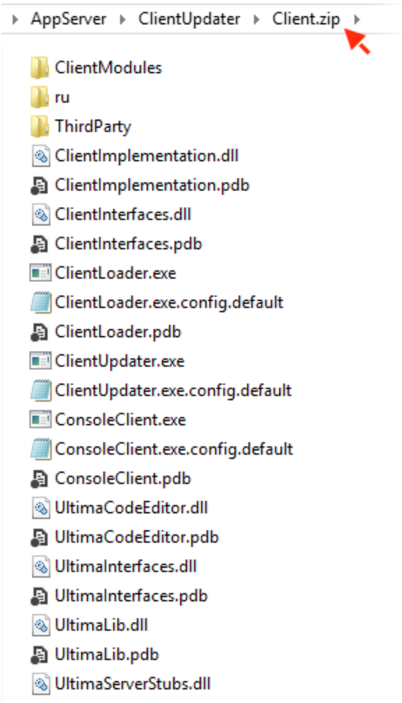


Application server, 31

Application server: 31 | OK | Save | Cancel

IP address, port	127.0.0.1	8192
Cluster	32, msapaev	
Export server, port	127.0.0.1	4096
Web server	☐ Localhost	8337
Print builders count	1	
Print senders count	1	

  - if the application server is starting as the console application, it shall be started as root;
  - archive with the actual distribution kit of the client application shall be placed in the folder: *AppServer/ ClientUpdater/Client.zip*;
  - the archive *Client.zip* must contain only the contents of *Client* folder of the client application, but not the *Client folder itself*;
  - the archive with the distribution package of the upgrade should not contain the configuration file of the client application *ClientLoader.exe.config*. Otherwise, the user configuration file will be rewritten during the upgrade process;
- on the client's computer, it is necessary to edit the configuration file of the upgrade utility by *ClientUpdater.exe.config*, setting the correct address of the upgrade server *UpdateServiceUrl* (the port of the web server shall be specified as the port).

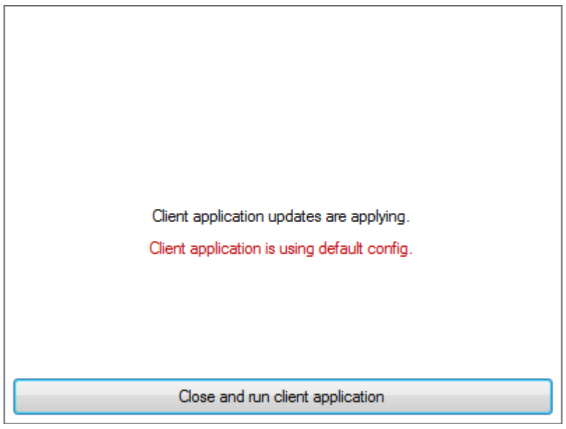


```
AppServer ▸ ClientUpdater ▸ Client.zip ▸

ClientModules
ru
ThirdParty
ClientImplementation.dll
ClientImplementation.pdb
ClientInterfaces.dll
ClientInterfaces.pdb
ClientLoader.exe
ClientLoader.exe.config.default
ClientLoader.pdb
ClientUpdater.exe
ClientUpdater.exe.config.default
ConsoleClient.exe
ConsoleClient.exe.config.default
ConsoleClient.pdb
UltimaCodeEditor.dll
UltimaCodeEditor.pdb
UltimaInterfaces.dll
UltimaInterfaces.pdb
UltimaLib.dll
UltimaLib.pdb
UltimaServerStubs.dll
```

When *ClientUpdater.exe utility is started:*

- the current version of the client application is verified;
- in case upgrade is not required, the client application is launched automatically upon expiration for the delay time set by the parameter *WindowShowingDelay* of the configuration file;

- in case upgrade is required:
  - the upgrades of the client application are successively loaded and applied;
  - in case of successful upgrade, the client application starts automatically;
  - in case there is no configuration file in the client application folder, the default configuration file is used, whereof a message in the utility form is displayed;


Client application updates are applying.
Client application is using default config.

Logging

System Ultimate Solid supports two ways of logging:

- the main option - host-centric logging based on Serilog library that supports a large range of storage of different granularity: from text files to document-database. Log analyzebuilt-in the system works with MongoDB 3.0 database that allows to write quite complex queries for document properties including regular expressions and search taking into account the Russian morphology. If desired, however, you can use any Serilog storage supported by the library including NLog using your own analysis tools;
- an additional option - logging on the NLog platform that supports a set of log output methods - from the file and the console wherein the application is running, to an event log and e-mail.

Each method has its advantages and defects.

So Serilog offers:

- structured storage of events (in the form of text and other properties);
- a large set of databases, formats and tools available for analysis;

  but:
- it can not overload settings from the configuration file on the fly;
- it does not allow multiple processes to write to a log file.

At the same time NLog:

- It allows multiple processes to write to a log file. It is actual for PrintWorker application; ;

  but it has some defects:
- there are own log files on each server for which analysis it is necessary to have access to each of them;
- log files still need to be found (this and the preceding paragraph are relevant more for an application developer);
- the files are quite large and there is no always a program at hand that can open them;
- when there is a lot of files it is difficult to determine which of them are relevant to the current time;
- as the text is not structured, search comes often across excess coincidence;
- search with regular expressions is not available in every program;
- sometimes the logs are stop at the most interesting place.

This Serilog can use NLog as a storage that completely eliminates its defects. In this case the need to write a single file from multiple processes is easily resolved by events redirecting in NLog.

Choice of logging method - only Serilog based on MongoDB or its storage, or Seroilog with NLog - is still for the system manager/the application developer.

Serilog

Serilog libraries are included in the Ultimate Solid distribution package and do not require additional installation.

The MongoDB 3.0 database supported by the log analyzer embedded in the system needs to be initially started and adjusted:

- create an empty folder for the database and execute the command:

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

- when running the database on Wired Tiger engine, you will need another empty folder and an additional parameter *--storageEngine*:

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

- example of configuration of an operative event base:

```
// restrict event collection volume down to 4 Gb
db.runCommand({"convertToCapped": "log", size: 4*1024*1024*1024});

// index desired properties
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});
```

- since SQL queries are saved in the log, it may contain secret data. Due to this, authorization is needed. To write logs, a new user is created, who has the privileges only to write data; his password needs not to be hidden from the public eye. To read logs, new users are created by the number of programmers:

```
// 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" }
    ]
  }
);
```

- after the users setup is completed, restart the MongoDB using the *--auth* key to activate authorization:

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

The detailed information on the MongoDB storage configuration can be found on the project's website www.mongodb.org. Below are a few notes:

- database file may grow fairly rapidly;
- MongoDB 3.0 supports Wired Tiger storage format;
- volumes of collections can be restricted (new events will overwrite the oldest ones);
- properties, which are of interest in terms of filtering, should be indexed;
- restrictions adjustments and indexing can be carried out on the run;
- when transiting to the packed format, restart the MongoDB server
- Since the Mongo's integrated framework is not very handy, you may wish to pay attention to Robomongo (doesn't support Wired Tiger for the present) or MongoChef.

The logging settings are performed for each separate application in configuration files. In the example below, the logging is set up to go to a file, the console and the MongoDB storage:

```
<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 supports the *MinimumLevel* conception. When the minimum level is specified, events below the level are not logged. Below are the event levels in order of increasing in criticality level:

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

If no minimal level specified, all events of *Information* level and higher are logged.

For more details on Serilog, see serilog.net.

Nlog

NLog platform libraries are included in the Ultimate Solid distribution package and do not require additional installation.

The logging settings are performed for each separate application in configuration files. However, for correct logging purposes, it is needed to also configure Serilog within the applications' configuration files; Serilog settings are marked with the tag *appSettings*. In the example below, Serilog is writing logs into NLog and 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>
```

The detailed information on NLog adjustments can be found on the platform's site ⇨ nlog-project.org. Below are a number of typical examples of configuration.

NLog logging settings for application server (*ConsoleServer.exe.config*). The log output is in the console and files. Some fast growing log files are overwritten and archived (archived if > 100 Mb, last 10 archives stored, files numbering per counter):

```
<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}
      ${exception:format=tostring:maxInnerExceptionLevel=10}"
      maxArchiveFiles="10" archiveNumbering="Sequence" archiveAboveSize="100000000"
      archiveFileName="${basedir}/ConsoleServer.Archive{####}.log" />
    <target name="ExceptionLog" xsi:type="File" fileName="${basedir}/Exceptions.log"
      layout="${newline}${date:format=yyyy\.MM\.dd\ HH\:mm\:ss}:
      ${exception:format=tostring,data:inner=tostring,data:maxInnerExceptionLevel=10}"
      maxArchiveFiles="10" archiveNumbering="Sequence" archiveAboveSize="100000000"
      archiveFileName="${basedir}/Exceptions.Archive{####}.log" />
    <target name="SqlLog" xsi:type="File" fileName="${basedir}/SqlStatements.log"
      layout="${newline}${date:format=yyyy\.MM\.dd\ HH\:mm\:ss}: ${message}"
      maxArchiveFiles="10" archiveNumbering="Sequence" archiveAboveSize="100000000"
      archiveFileName="${basedir}/SqlStatements.Archive{####}.log" />
    <target name="ServerCallLog" xsi:type="File" fileName="${basedir}/ServerCalls.log"
      layout="${date:format=yyyy\.MM\.dd\ HH\:mm\:ss}: ${message}"
      maxArchiveFiles="10" archiveNumbering="Sequence" archiveAboveSize="100000000"
      archiveFileName="${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>
```

NLog logging setting for the PrintWorker application (*PrintWorkerApp.exe.config*) of a print server. Logging to a single log file with multiple processes (*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>
```

Print process description

Let us research the chain, which preact the print task.

1. The user start the print function on the applications server. Printing may be:

- *Synchronous (common)* – in this case:
  - is performed the handler, which at the first onset computes and create the object with data for the print form;
  - the client waits for the work end of this handler;
- *Asynchronous (delayed)* – in this case the data will be formed later.

If the user brings about the print function directly into the screen form, for example, printing the document from his editing form or dictionary record of list form, the print always will be synchronous. In case, if the print is brought about by script or handler, the way selection will stay for application developer.

The applied developer can use the asynchronous print if he is sure that in fact of deferred data forming for printing, the user will get the necessary result anyway. For example, you can print the document asynchronous of the goods shipment on the storage when he order status changes to “paid”.

2 The applications server saves the task in the database on the server Oracle.

3 All apps servers, which have the permission for this by appropriate settings, refers to the database server and sends received from him tasks on (Print Service).

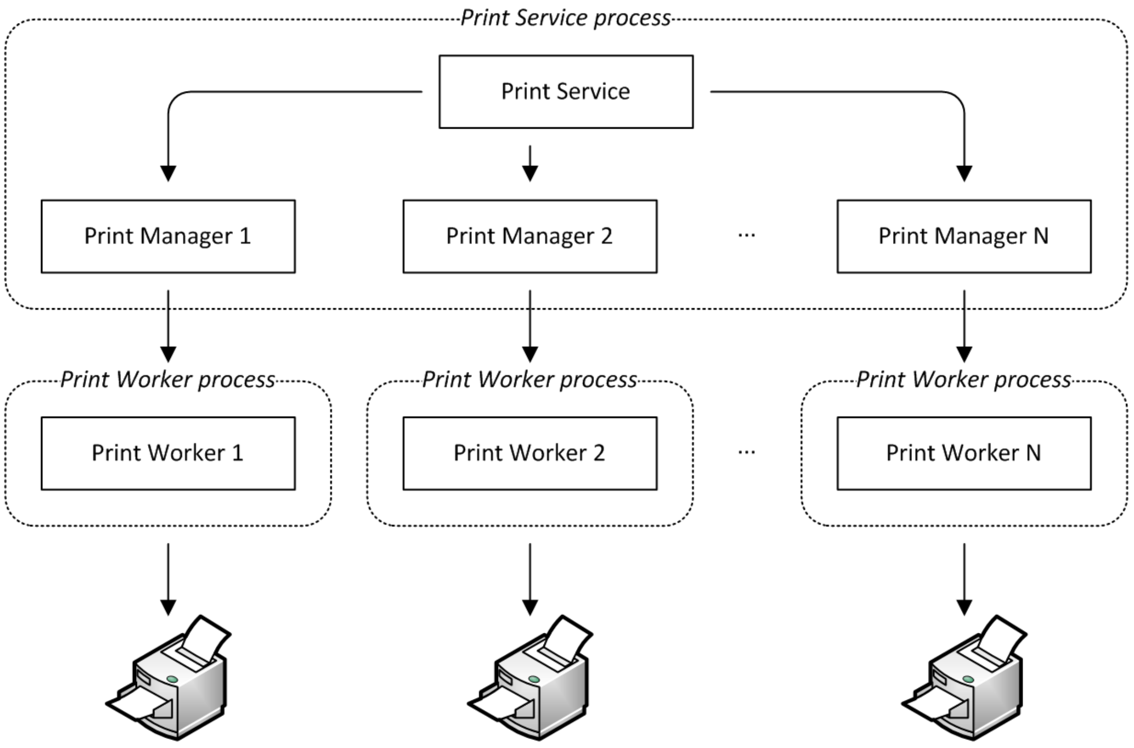
Oracle groups all the entering for the print tasks for the printers, on which they were sent, and, ignoring turn of their arrival, distributes tasks for appealing to it applications servers for each printers However in order for the each concrete printer ,the tasks are distributed in the order of their supplies - firstly the first incoming task is send, after this the second and etc. The fragmentation the common print order the orders for printers realized so as the printers not to stay without the work. It can be happened in case when the big number of tasks are flagged on the on loaded printer successively, for example hundred, and after this one hundred and first task are sent to the other idle printer, which must wait the sending of previous hundred tasks in case of common call.

The tasks can be print by packages. It was realized for cases, when you need print some print forms for the one printer so as not to break the given print sequence by other wedge tasks. The task package will be always printed in the predestinated sequence independently of the supply order for the print of tasks package.

If the data for the print hadn't been calculated at once (the case of asynchronous print), the applications server would determine it before dispatch on the Print Service. The database server gives this tasks for applications servers which have the permission of the calculate operation by settings.

In case of successful dispatch of task on the Print Service, it will be deleted from the database.

4. Print Service accepts the tasks from applications server and distributes it of Print Manager.



The separate Print Manager is launched for each printer, which works with print tasks only for this printer. If the Print Service doesn't found the Print Manager for any printer (the task is sent on the printer firstly from the start moment of Print Service), it will create the new Printer Manager.

All print managers and Print Service work within the framework of one service on the print server.

Everyone of Print Manager:

- It keeps in the file own call of tasks for the print locally. It let to recover the print tasks in case of crashing and restarting of print service;
- It sends the tasks on the printer through the separate special process Print Worker;
- The tasks are kept in the order until Print Worker doesn't print it.

Print Worker is the separate process, which:

- has its own call of the tasks, which were sent for the print in the order of this supply on print Worker - firstly it is sent the first incoming task, after this the second and etc;
- creates (renders) the print form;

- sends the created form on the printers;
- works with driver of printer directly.

Each of the task is rendered before the print. For this action it is required a certain amount of time, in the process of this printer is inactive. So as the printer not to stand idle possibly, the tasks rendering may be in separate additional stream. The main stream would get the ordering task and renders it, if this hadn't been done by additional stream, after this sends for the print.

Print Service controls by underlining the additional streams proportionally of the Print Worker processes loads. The number of additional streams, which are controlled by Print Service, is determined by *AdditionalThreadsCount* parameter in file of its configuration.

The time-out is given on the rendering process, which size is determined in the configuration file Print Server by parameter *TaskBuildingTimeOut*. On expiry of this time-out the print task enters an invalid state and is deleted from the calls Print Worker and Print Manager. The notification about the inability of task print is sent for the user, which sent this task.

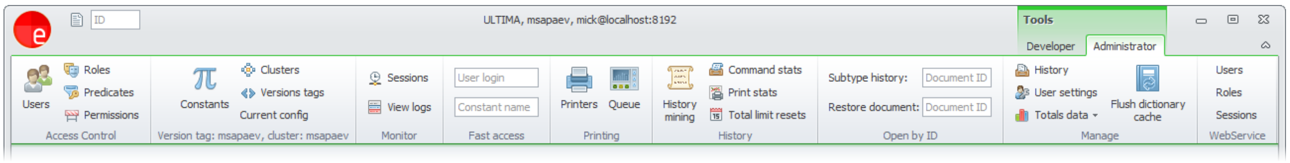
Also the time-out is set on the print process directly: The time-out size of the print of one page is determined in the configuration file Print Service by *PagePrintingTimeOut*. The time-out size of task print is proportional to the page number in task and is equal of product [the number of task pages]*[the number of print copies]*[the time-out size]. After the lapse of time-out, if the print hadn't been ended, Print Service would restart the appropriate process of Print Worker.

The meaning of the Print Worker underlining in the separate process is in the ability of its restart, when the printer driver is suspended, without a loss of print task and restart of all print service. In the case of Print Worker restart, Print Manager sends to Print Worker the tasks order again.

Functionality of the administrator module

To receive the access to the Administrator module the user has to have the right of *Administrator*.

All the main tools of system administrators Ultimate Solid are organized at the tab "Administrator":



Tools are divided into groups according to their mission:

- *Access control* – tools of work with users and their rights;
- tools of control of system configuration;
- *Monitor* – tools of control of system work;
- *Fast access* – tools of fast access to the objects;
- *Printing* – tools of work with the printing;
- *History* – tools of work with history of actions and logs;
- *Open by ID* – tools of fast access to demanded operations on objects;
- *Manage* – tools of management of settings;
- *WebService* – tools of work with web services.

Besides, in the menu the tool settings of the main menu (UI) of client application is available.

Access control tools

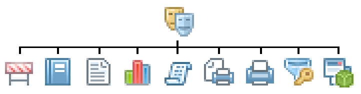
Any manipulations with data in system Ultimate Solid are made under any user with a final set of the rights. The rights define whether the user can make this or that action or get access to any object.

Ideology of system of the rights is that any action or a possibility of access to the object needs initially to be allowed, i.e. – that is not authorized is forbidden.

The following objects of the rights are differed in the system:
The rights, formulated and checked by the developer. These rights are checked at the level of the executed code. *Login under another name* or *Administrator permission* can be so;
Permissions to access to objects of metadata:
- dictionaries;
- documents;
- totals;
- user commands, commands on the dictionaries and documents;
- printers;

Predicates, allowing to limit access for the user at the level of separate lines;
Client modules.

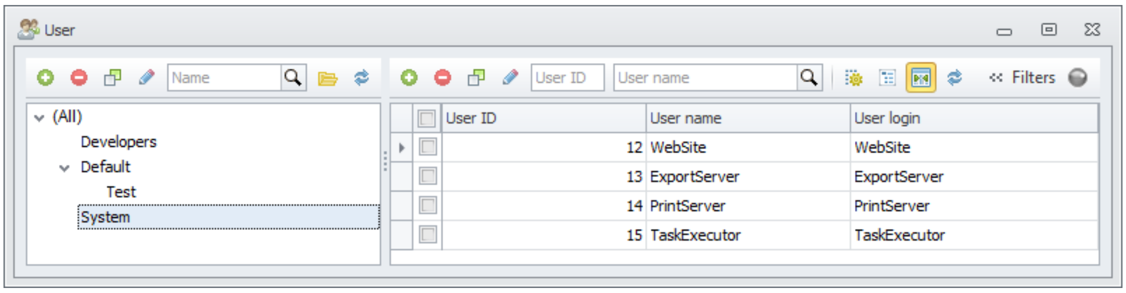
Roles combine all the above-mentioned objects of the rights:



Being assigned to the user, the role gives him the corresponding rights. Only one role can be assigned for one user. The role, in turn, can consist of other roles. Also usually same role is appointed to users with identical functionality.

Users

The list of all system users can be found in the User dictionary:



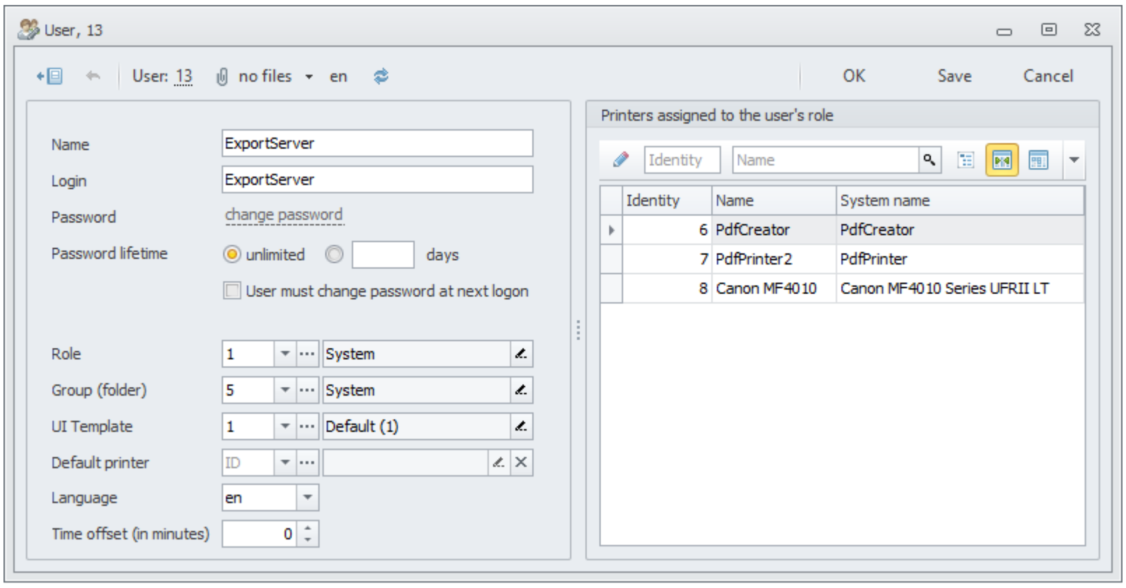
	User ID	User name	User login
	12	WebSite	WebSite
	13	ExportServer	ExportServer
	14	PrintServer	PrintServer
	15	TaskExecutor	TaskExecutor

The dictionary window is divided into two parts: on the left, there is a tree of user groups, on the right – the list of users of the group selected on the left.

The *System* group is used for *system users* under which the service requests to the database are executed without immediate participation of the company staff, for example, *the print server* and *export serverwork, tasks are started*.

User groups can be filtered by Group *Name(Name*) and users - by *name* (*User name*).

Each user has the following properties:



User, 13

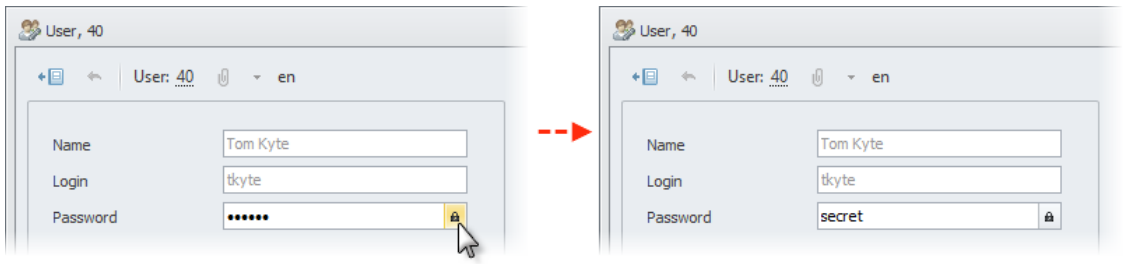
User: 13 no files en OK Save Cancel

Name	ExportServer
Login	ExportServer
Password	change password
Password lifetime	unlimited / days
	User must change password at next logon
Role	1 System
Group (folder)	5 System
UI Template	1 Default (1)
Default printer	ID
Language	en
Time offset (in minutes)	0

Printers assigned to the user's role

Identity	Name	System name
6	PdfCreator	PdfCreator
7	PdfPrinter2	PdfPrinter
8	Canon MF4010	Canon MF4010 Series UFRII LT

- *Name* – user name;
- *Login* – login;
- *Password* – a link to change a password. For the user created before, the password can be changed by clicking on the *change password* link. For a new user password must be entered. The entered password can be shown:



User, 40 — User: 40 en — Name: Tom Kyte; Login: tkyte; Password: ••••••

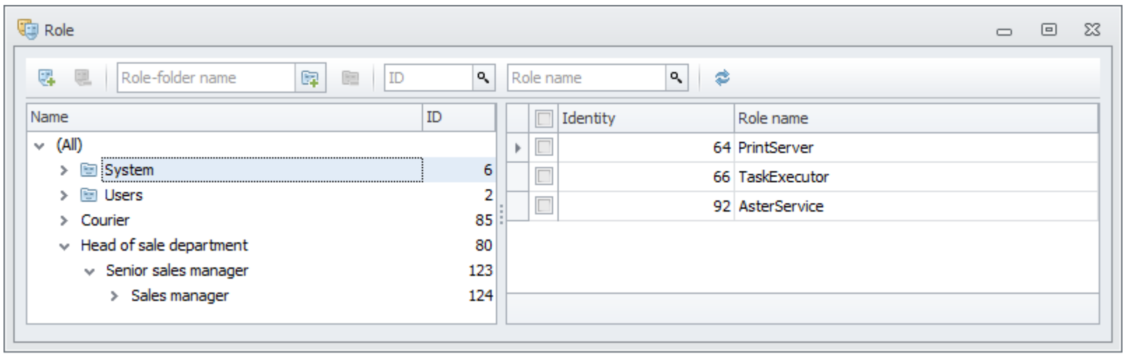
User, 40 — User: 40 en — Name: Tom Kyte; Login: tkyte; Password: secret

- *Password lifetime* – life time of the password:
  - *unlimited* – not time-limited;
  - *days* – limited by the number of days entered in that field. Upon expiration of the above term in case of next logon to the system, the user will be prompted to change the password.
  - When choosing the limitation of the password lifetime, additional field *Valid until* which contains the date till which the user password is valid, in case its effective time is limited;
- *User must change password at next login* – the user to which this flag is selected will have to change the password during next logon. After the password is changed, the flag will be deselected automatically;
- *Role* – role of the user determining his permissions;
- *Group (folder)* – the group to which the user belongs. For the created user it is set automatically in accordance with the group selected in the filter of the list form of the dictionary at the time of creation;
- *UI Template* – user interface;
- *Default printer* – default printer;
- *Language* – user language. Russian language is default for the created user;
- *Time offset (in minute)* – time offset of the user in minutes compared to the main time zone used by the company. Used for accounting the difference in time between users working in different time zones. The main time zone can be the time zone of the head office or UTC+0 (GMT+0). For the users working in the main time zone, the time shift is 0.
  For example, the main time zone in the company is Moscow (UTC+4). Then the time for the users

working in Kaliningrad (UTC+3) will be "-60”, and for Yekaterinburg users (UTC+6) – "120";
- *Printers assigned to the user's role* – the list displays the printers access to which is provided by the role of the Role user.

Roles

The roles are organized in the dictionary Roles in a tree structure where each role can have several parents and children:


Role

Role-folder name | ID | Role name

Name	ID
(All)	
System	6
Users	2
Courier	85
Head of sale department	80
Senior sales manager	123
Sales manager	124

Identity	Role name
64	PrintServer
66	TaskExecutor
92	AsterService

The dictionary window is divided into two parts: the tree of roles on the left and first level child roles on the left.

In the tree on the left, the roles are sorted in alphabetical order, and the folders always precede the ordinary roles.

Dictionary records can be searched by *Role name*. The search is carried out by occurrence of the searched text in the name of the role when clicking the button of the control element or pressing Enter. In case of successful search, the cursor is set on the first record meeting the conditions, in case of unsuccessful, remains on the previous one. Another click on the search button will result in finding the next record meeting the conditions of searching the entry and setting the cursor on it.

The role can also be a folder. Such roles are designated by the icon before its name. The folder roles do not have any permissions. They are designed exclusively for convenience of organizing the structure of roles and are used, for example, for grouping them. The functionality of the folder roles is restricted to a number of prohibitions:
- the folder role does not enable you to grant access to any of the object of permissions;
- a folder-role cannot be assigned to the user;
- a folder-role cannot be made a child for other role if not the folder either.

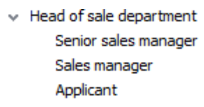
The functionality of the dictionary of roles allows to perform the following actions:
- create new roles. The new role is created at the uppermost level regardless of what role is selected on the left in the tree;
- delete the selected roles which are not folders;
- create folder roles with the name entered in the field Folder role name, without opening the form to create a new role. The new folder role is created at the uppermost level regardless of what folder role is selected on the left in the tree;
- delete selected folder roles.

Each role can have an unlimited number of children and can be a child for an unlimited number of parents.

It is impossible to define the parent role for the role, only vice versa. That’s why the newly created role always appears on the upper level of the tree.

If the created new role (let’s name it “A”) must be a child for any other role (for example, “B"), it is necessary to save role "A", then open role "B" in the editor and bind role "A" to it as child.

Tree-form dictionary of roles also has important functional load: Parent role takes over all permissions of children role whatever level of nesting. For example, the roles determining the functionality of subsidiaries, being the child of the role of the head of the department, give him an opportunity to do all operations available for them:

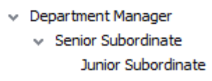


Therefore, the permissions can be explicitly allowed for the role or derived by it from the child role (derived).

The permissions derived from the children can be also allowed for the parent role in an express form, so that the parent role does not lose the required right in case of its exclusion from the child role.

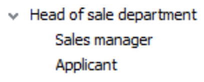
However, there may occur a situation where a child role has certain permissions, and the parent role should not have such permissions. For example, the role of the man has the right to ship products from the storehouse. At the same time, it is a child of the role of the storehouse coordinator. But the storehouse coordinator is not an accountable officer and does not have the right to do such shipment. The parent role of the storehouse coordinator can be deprived of such functionality by revoking the relevant permission they derived from the child role.

Such revoked permission will not be derived by the parent role. For example,granting permission for the *Junior Subordinate role*, automatically grants the ones to the parent roles – the – *Senior Subordinate* and the *Department Manager*.



Revoking such permissions for the role of the *Senior* Subordinate, we thus revoke them as well for the *Department Manager* (unless such permissions were received by the *Department Manager* also from other child roles).

If the role of the parent role has two same-level child roles, one of which is allowed some permissions and the other - vice versa (revoked), such right will be derived by the parent role. That is, allow has higher priority than revoke.



Summarizing the above, the access to the object of permissions in the form of editing the role will be set with the help of flags:

- Derived – access is derived from the child role. You can’t deselect that;
- Allow – allowed. This flag can be also selected when the Derived flag is selected. It is mutually exclusive with the Revoke flag;
- Revoke – revoked. Selecting this flag revokes the derived access to the object of permissions for the current role and for its parents. Accordingly, selecting it makes sense only when the Derived flag is selected. It is mutually exclusive with the Allow flag.

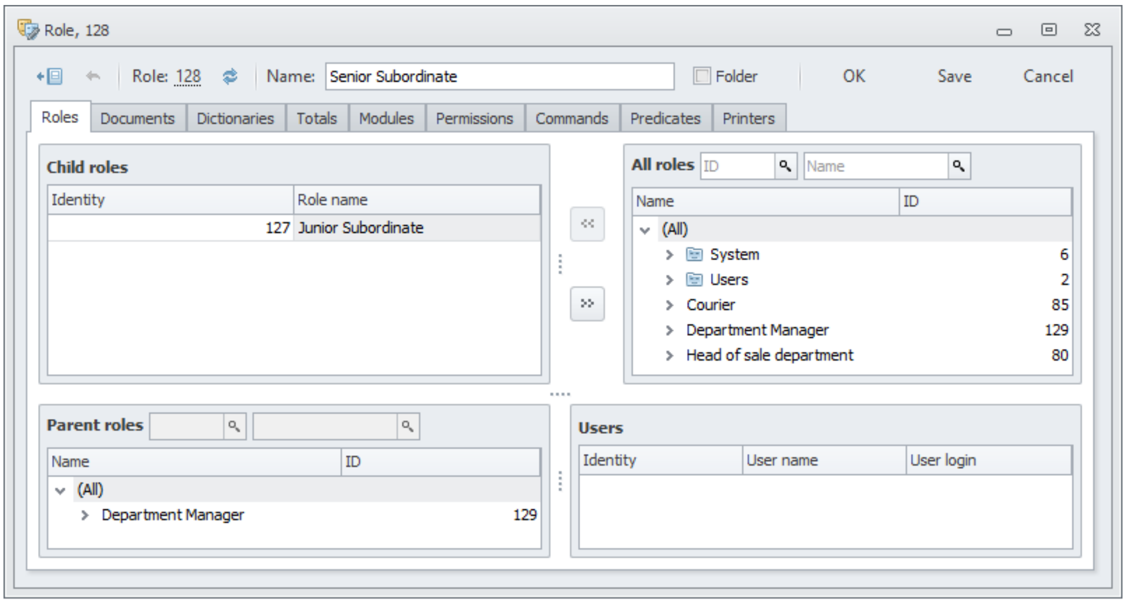
In addition, for a number of permissions objects access can be granted not on the object itself, but on performance of four types of operations with the object (CRUD):

- Read – reading the objects and opening a list form of the objects of such type;
- Create – create a new object;
- Update – edit object;
- Delete – delete object.

Having mastered the mechanism of deriving permissions, we can go to description of the role edit form.

Role editing form

All properties of a role except those added to the toolbar of the role *Name* form and *Folder* flag indicating that the role is a folder are grouped by semantic tabs:


Role, 128

Role: 128 | Name: Senior Subordinate | ☐ Folder | OK | Save | Cancel

Tabs: Roles | Documents | Dictionaries | Totals | Modules | Permissions | Commands | Predicates | Printers

Child roles

Identity	Role name
127	Junior Subordinate

All roles

Name	ID
(All)	
System	6
Users	2
Courier	85
Department Manager	129
Head of sale department	80

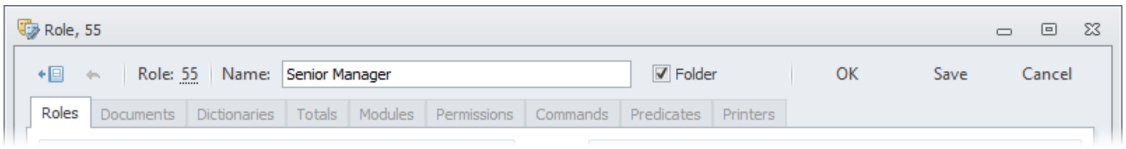
Parent roles

Name	ID
(All)	
Department Manager	129

Users

Identity	User name	User login

For the folder role, only Roles tab is available:


Role, 55

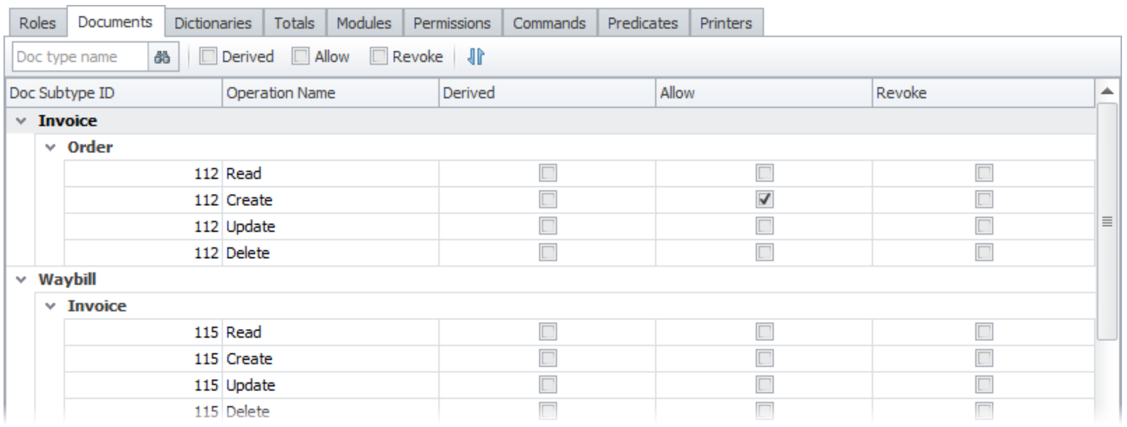
Role: 55 | Name: Senior Manager | ☑ Folder | OK | Save | Cancel

Tabs: Roles | Documents | Dictionaries | Totals | Modules | Permissions | Commands | Predicates | Printers

In the Roles, tab one can change the list of roles which are children for the edited role, and view the structure of parent roles and the list of users whose permissions is affected by the edited role:

- the area “Child roles” contains the roles - children of the edited role. They can be deleted by selecting and clicking the button ">";
- “All roles” area contains a tree of all existing roles. They can be added as children to the edited one by selecting and clicking the button "<" (the roles in the tree are always sorted alphabetically; the folders always precede the ordinary roles);
- "Parent roles" area displays a tree of all parent roles, for which the edited role is a child (the roles in the tree are sorted in alphabetical order and the folders always precede usual roles);
- “Users” area displays all users, to whom the edited role or the roles from the “Parent roles” list are assigned. Thus, you can see the whole list of users, which will be affected by the changes of the role being edited.

In the “Documents" tab, an access to document subtypes is set:


Roles | Documents | Dictionaries | Totals | Modules | Permissions | Commands | Predicates | Printers

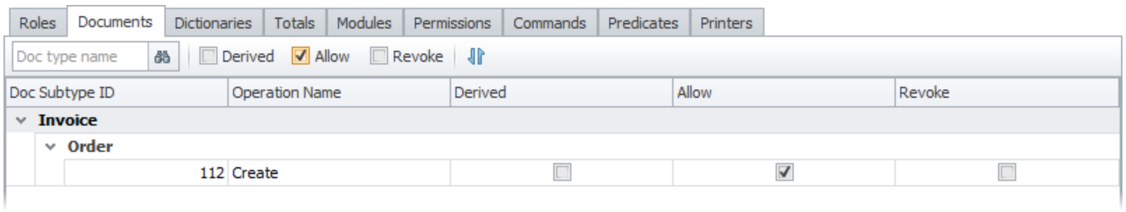
Doc type name ☐ Derived ☐ Allow ☐ Revoke

Doc Subtype ID	Operation Name	Derived	Allow	Revoke
Invoice				
Order				
112	Read	☐	☐	☐
112	Create	☐	☑	☐
112	Update	☐	☐	☐
112	Delete	☐	☐	☐
Waybill				
Invoice				
115	Read	☐	☐	☐
115	Create	☐	☐	☐
115	Update	☐	☐	☐
115	Delete	☐	☐	☐

Four types of operations (CRUD), which are grouped for this subtype, can be executed in documents of each subtype. The subtypes, in their turn, are grouped by type of documents.

The “Documents” tab has its own toolbar, which allows to do the following:

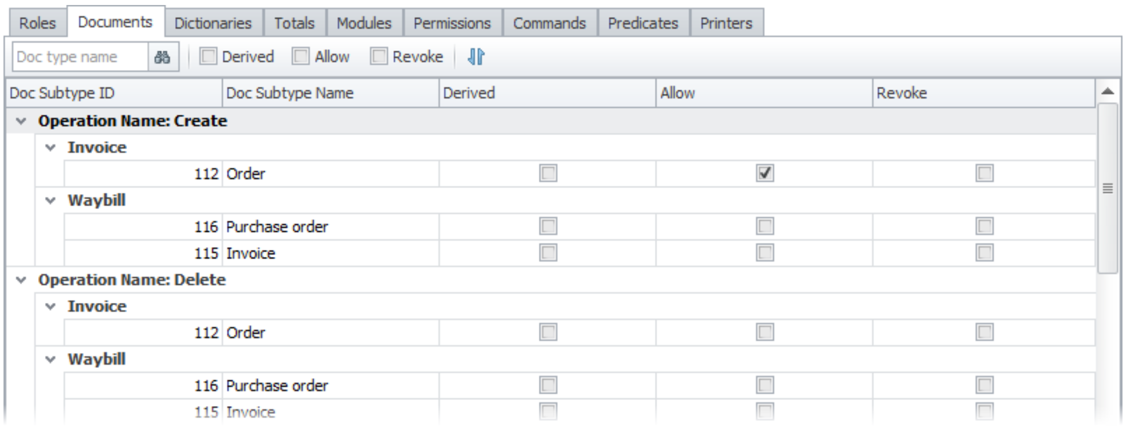
- filter the displayed types of documents in accordance with the text entered into the field “Doc type name”. For filtering, enter the value in the text field and click . You may reset the filter by cleaning a text field and making a repeated search;
- filter by *allowed*, *derived* and *canceled* operations with document subtypes by selecting the flags “Derived”, "Allow" and "Revoke" respectively. When selecting one or more flags of the filter in the toolbar, the content of the tab will be filtered in such a way so that only the operations with the relevant flags remain displayed:


Roles | Documents | Dictionaries | Totals | Modules | Permissions | Commands | Predicates | Printers

Doc type name ☐ Derived ☑ Allow ☐ Revoke

Doc Subtype ID	Operation Name	Derived	Allow	Revoke
Invoice				
Order				
112	Create	☐	☑	☐

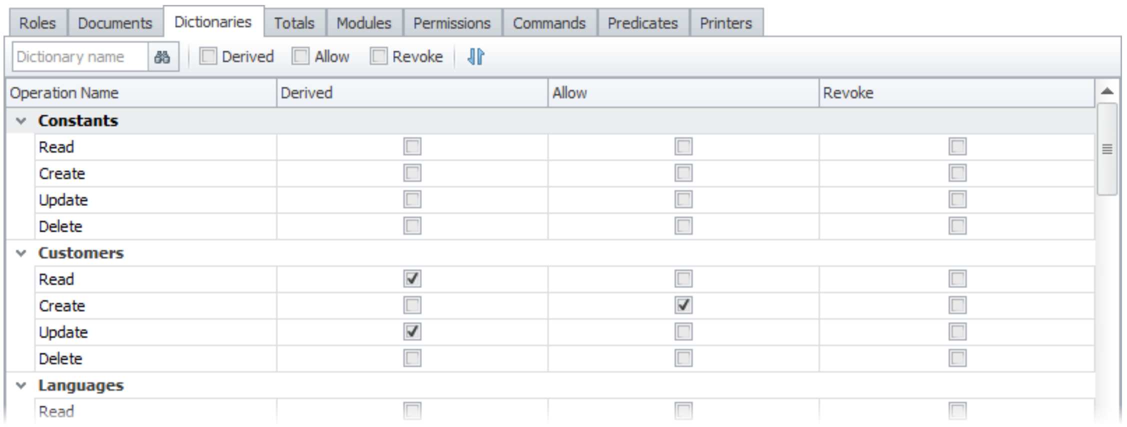
  Two selected flags of the filter operate in logical conjunction. That is, the satisfactory conditions of filtering will be operations with two simultaneously selected flags.
- additional grouping of the content by type of operation by clicking . In this case, Create-type operations go first, also grouped by type and subtype of document, then Delete, Read and Update. The repeated click on the removes additional grouping:


Roles | Documents | Dictionaries | Totals | Modules | Permissions | Commands | Predicates | Printers

Doc type name ☐ Derived ☐ Allow ☐ Revoke

Doc Subtype ID	Doc Subtype Name	Derived	Allow	Revoke
Operation Name: Create				
Invoice				
112	Order	☐	☑	☐
Waybill				
116	Purchase order	☐	☐	☐
115	Invoice	☐	☐	☐
Operation Name: Delete				
Invoice				
112	Order	☐	☐	☐
Waybill				
116	Purchase order	☐	☐	☐
115	Invoice	☐	☐	☐

The "Dictionaries" tab establishes access to dictionaries:


Roles | Documents | Dictionaries | Totals | Modules | Permissions | Commands | Predicates | Printers

Dictionary name | Derived | Allow | Revoke

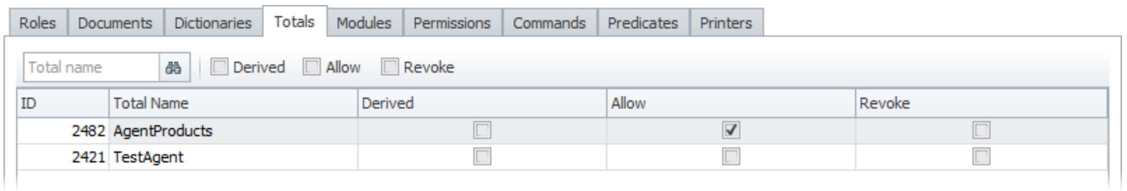
Operation Name	Derived	Allow	Revoke
Constants			
Read	☐	☐	☐
Create	☐	☐	☐
Update	☐	☐	☐
Delete	☐	☐	☐
Customers			
Read	✓	☐	☐
Create	☐	✓	☐
Update	✓	☐	☐
Delete	☐	☐	☐
Languages			
Read	☐	☐	☐

For each dictionary, access is provided for making operations of four types (CRUD) which are grouped by this dictionary.

The “Dictionaries” tab has its own toolbar which allows to do the following:

- filter the displayed dictionaries in accordance with the text entered into the field “Dictionary name”;
- filter by *allowed*, *derived*and *canceled* operations with dictionaries by selecting the flags “Derived”, "Allow" and "Revoke" respectively (two selected flags of the filter operate in logical conjunction);
- additional grouping of the content by type of operation by clicking . Clicking again the button will remove the additional grouping.

In “Totals" tab, the access to the totals is set:


Roles | Documents | Dictionaries | Totals | Modules | Permissions | Commands | Predicates | Printers

Total name | Derived | Allow | Revoke

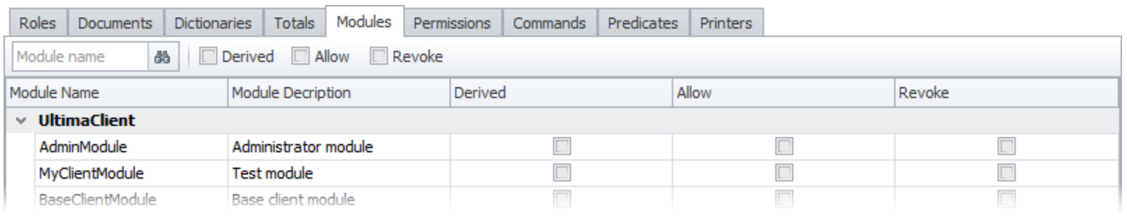
ID	Total Name	Derived	Allow	Revoke
2482	AgentProducts	☐	✓	☐
2421	TestAgent	☐	☐	☐

Getting access to the totals helps the user build reports on such total using the internal reporting system.

The “Total” tab has its own toolbar which allows to do the following:

- filter the displayed totals in accordance with the text entered into the “Total name” field;
- filter by *allowed*, *derived* and *canceled* totals by selecting the flags “Derived”, "Allow" and "Revoke" respectively (two selected flags of the filter operate in logical conjunction).

In “Modules" tab, the access to the client modules is set:


Roles | Documents | Dictionaries | Totals | Modules | Permissions | Commands | Predicates | Printers

Module name | Derived | Allow | Revoke

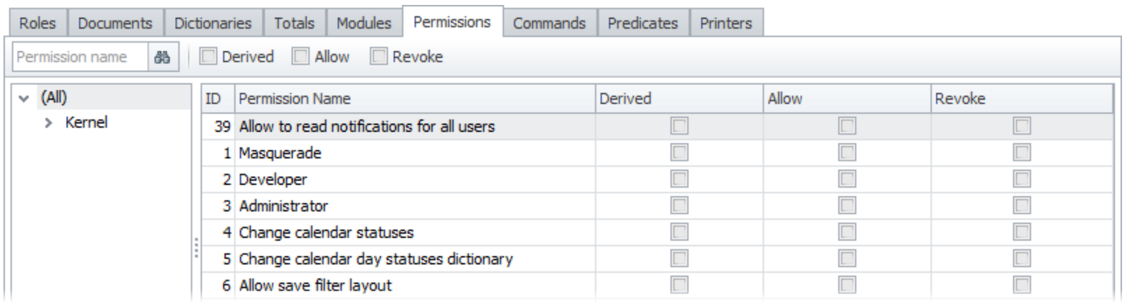
Module Name	Module Decription	Derived	Allow	Revoke
UltimaClient				
AdminModule	Administrator module	☐	☐	☐
MyClientModule	Test module	☐	☐	☐
BaseClientModule	Base client module	☐	☐	☐

The user whose role can access to the client module can load and launch it.

The “Modules” tab has its own toolbar which allows to do the following:

- filter of the displayed client modules in accordance with the text entered into the “Module name” field;
- filter by *allowed*, *derived* and *canceled* client modules by selecting the flags “Derived”, "Allow" and "Revoke" accordingly (two selected flags of the filter operate in logical conjunction).

In the “Permissions” tab, the access to the permissions formulated and verified by the developer is set:


Roles | Documents | Dictionaries | Totals | Modules | Permissions | Commands | Predicates | Printers

Permission name | Derived | Allow | Revoke

(All)
Kernel

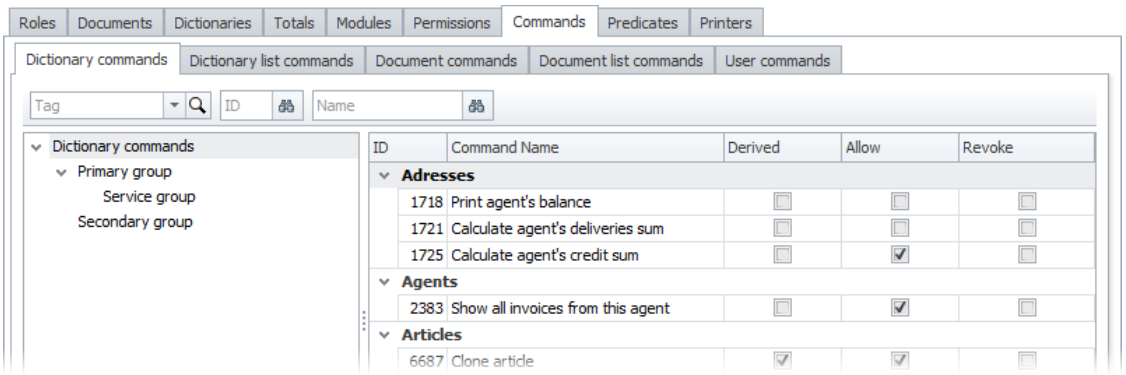
ID	Permission Name	Derived	Allow	Revoke
39	Allow to read notifications for all users	☐	☐	☐
1	Masquerade	☐	☐	☐
2	Developer	☐	☐	☐
3	Administrator	☐	☐	☐
4	Change calendar statuses	☐	☐	☐
5	Change calendar day statuses dictionary	☐	☐	☐
6	Allow save filter layout	☐	☐	☐

The permissions tab is divided into two parts: to the left, a tree of the permissions group is displayed, and to the right, the permissions of the group selected on the left (including the permissions of all child groups).

The “Permissions” tab has its own toolbar which allows to do the following:

- filter of the displayed rights in accordance with in accordance with the text entered into the field “Permissions name”;
- filter by *allowed*, *derived* and *canceled* permissions by selecting the flags “Derived”, "Allow" and "Revoke" respectively (two selected flags of the filter operate in logical conjunction).

In the “Commands" tab, the access to execution of commands is set:


Roles | Documents | Dictionaries | Totals | Modules | Permissions | Commands | Predicates | Printers

Dictionary commands | Dictionary list commands | Document commands | Document list commands | User commands

Tag | ID | Name

Dictionary commands
Primary group
Service group
Secondary group

ID	Command Name	Derived	Allow	Revoke
Adresses				
1718	Print agent's balance	☐	☐	☐
1721	Calculate agent's deliveries sum	☐	☐	☐
1725	Calculate agent's credit sum	☐	✓	☐
Agents				
2383	Show all invoices from this agent	☐	✓	☐
Articles				
6687	Clone article	✓	✓	☐

All commands are divided into five types with help of tabs:

- Dictionary commands – commands executed over dictionary record;
- Dictionary list commands – commands executed over several dictionary records;
- Document commands – commands executed over a document;
- Document list commands – commands executed over several documents;
- User commands – user commands.

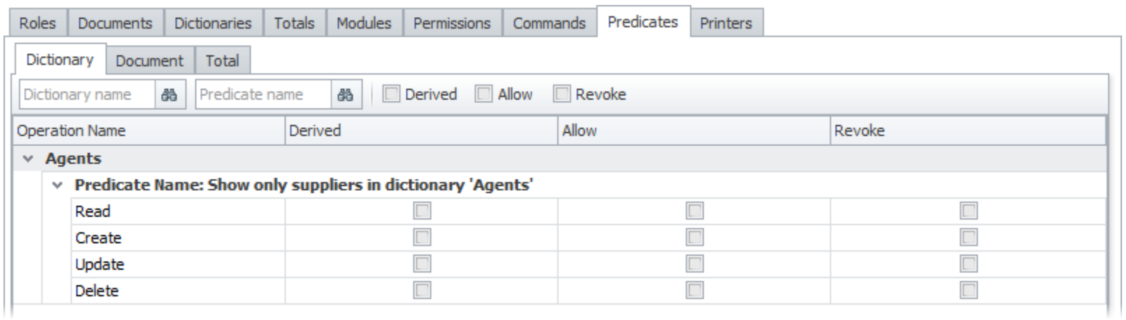
Each commands tab is divided vertically into two parts: to the left, a tree of the commands group is displayed, and to the right, the commands of the group selected on the left (including the commands of all child groups).

In the list in the tab "Dictionary commands" and "Dictionary list commands", the commands are grouped by dictionaries, over records of which they are executed. In the list in the tab “Document commands”, the commands are grouped by subtypes of documents which in their turn are grouped by types of documents. In the list in the “Document list commands" tab, the commands are grouped by types of documents.

Commands of each type can be filtered by tag, identifier and name in accordance with the text entered in the fields "Tags", "ID" and "Name". For filtering, enter the value in the text field and click . The filter can be reset by clearing the text field and repeated search.

To execute a command, the user must have only the permissions to launch it. In such case the command will be performed correctly even if it performs the actions unavailable to the user in ordinary mode of operation, for example, adds a record to the dictionary the user does not have access to at all.

The “Predicates” tab lists the predicates which can be applied to the given role:



Operation Name	Derived	Allow	Revoke
Agents			
Predicate Name: Show only suppliers in dictionary 'Agents'			
Read	☐	☐	☐
Create	☐	☐	☐
Update	☐	☐	☐
Delete	☐	☐	☐

All predicates are split by tabs into three groups by objects to which they apply:

- Dictionary – dictionaries;
- Documents – documents;
- Totals – totals.

For *totals* , the predicates apply directly to the object (at the time of building by the user of report), for *dictionaries* and *documents* – to each operation (Read, Create, Update, Delete) separately. The sense of applying predicates to each type of operations separately is that it might be necessary not to limit the functionality of actions over the objects for the user, but only part of it. For example, the user needs to see the whole list of dictionary records of agents, but create, edit and delete only part of it – natural persons. In such case the predicate applies to all types of operations, except Read.

In the list in the “Dictionary" tab, the predicates are grouped by dictionaries to which they are assigned, and in the list in the "Documents" tab - by types of documents.

Since the predicate restricts access to an object, the role where the predicate is applied must also have access to such object. For example, if a user needs to provide a limited access to the dictionary of agents, it is necessary for its role:

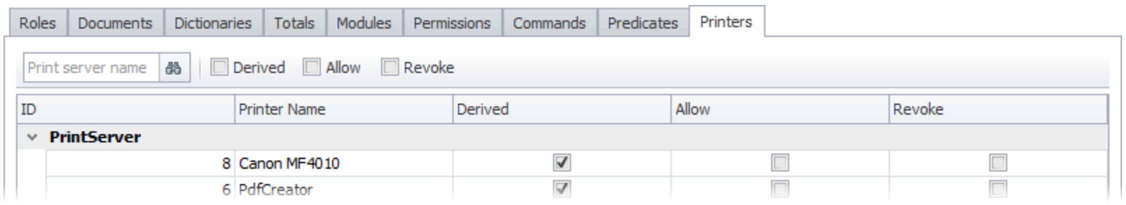
- provide access to the dictionary of agents;
- apply the predicate restricting such access.

In such case the operation which is restricted by the predicate must also be allowed. For example, application of the predicate imposing restrictions on the operation of deleting records from the list of agents, for the role which will be granted access only to read, add or modify the entries of such dictionary will be senseless, because in such case there is an attempt to limit the access to the operation which still is not allowed.

Each of the subtabs by type of predicate "Dictionary", "Documents" and "Totals" in the “Predicates” tab has its own toolbar which allows to do the following:

- filter the displayed predicates by type of object to which they apply in accordance with the text entered in the field "Dictionary name" (for dictionaries, "Doc type name" for document types, "Total name" for totals);
- filter the displayed predicates in accordance with the text entered in the field "Predicate name";
- filter by *allowed*, *derived* and *canceled* predicates by selecting the flags “Derived”, "Allow" and "Revoke" accordingly (two selected flags of the filter operate in logical conjunction).

The “Printers" tab, the access to the totals is set:



ID	Printer Name	Derived	Allow	Revoke
PrintServer				
8	Canon MF4010	✓	☐	☐
6	PdfCreator	✓	☐	☐

The printers available to the user are displayed in the standard print form of Ultimate Solidsystem, while unavailable are invisible.

In the list in the “Printers” tab, the printers are grouped by print servers.

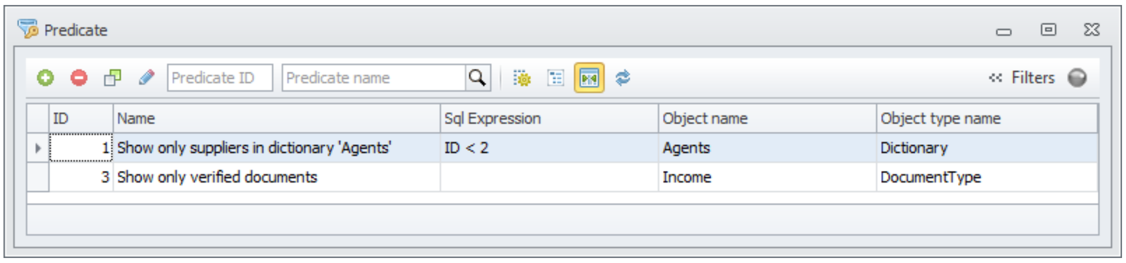
The “Printers” tab has its own toolbar which allows to do the following:

- filter the displayed printers in accordance with the text entered in the field "Printer name";
- filter by *allowed*, *inherited* and *canceled* printers by selecting the flags “Derived”, "Allow" and "Revoke" accordingly (two selected flags of the filter operate in logical conjunction).

Predicates

Predicates are objects of the permissions which allow to limit the user access at the level of separate lines.

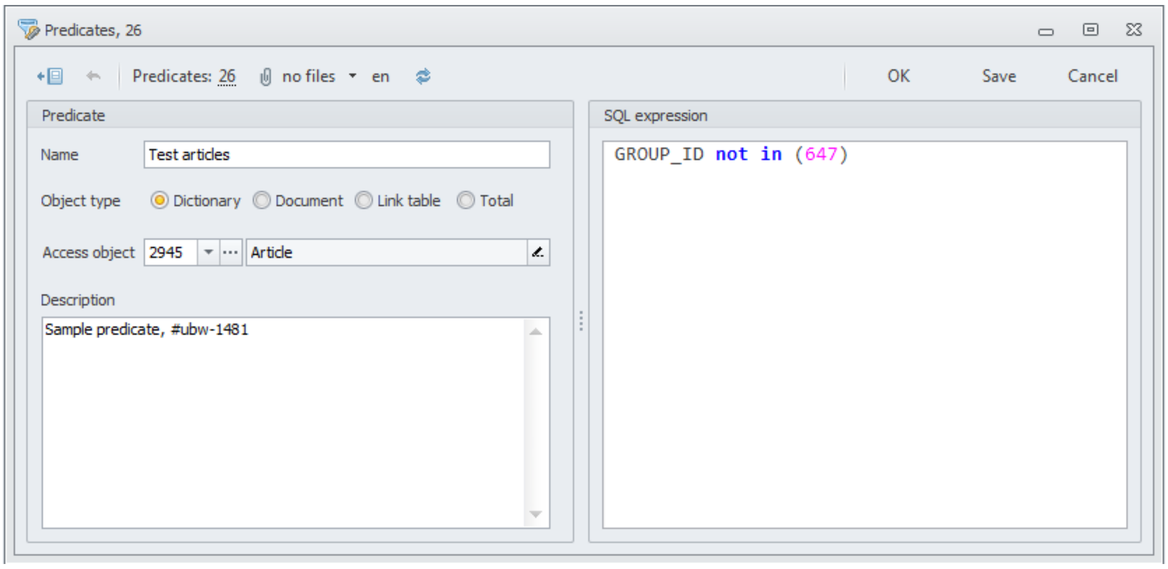
Predicates are organized in the dictionary Predicate:



ID	Name	Sql Expression	Object name	Object type name
1	Show only suppliers in dictionary 'Agents'	ID < 2	Agents	Dictionary
3	Show only verified documents		Income	DocumentType

Dictionary records can be filtered according to the *Predicate name* (*Predicate name*).

For a new predicate it is necessary to specify:



Name	Test articles
Object type	[✓ Dictionary] Document Link table Total
Access object	2945 Article
Description	Sample predicate, #ubw-1481

```
GROUP_ID not in (647)
```

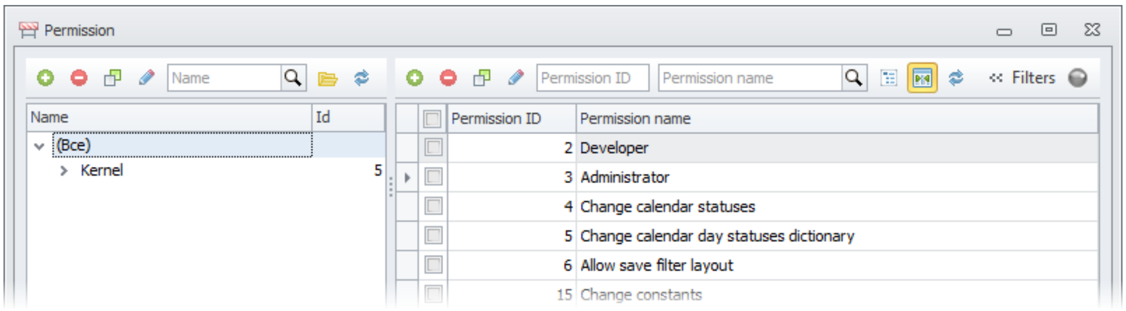
- *Name* – predicate name;
- *Access object* object type – *Dictionary*, *Document* or *Total* (dictionary, document or result) – and

directly object to which the predicate is applied;

- *SQL expression* – actually the expression in SQL language. This expression will be substituted at implementation of any reference to the corresponding table;
- *Description* – description of predicate action.

Permissions

Permissions, formulated and checked by the developer at the level of the executed code, are organized in the dictionary Permission:



Permission

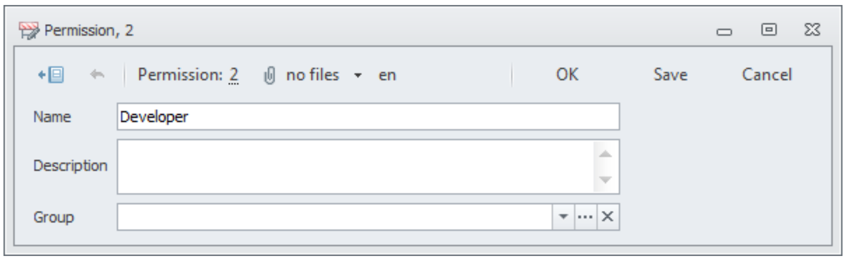
Name	Id
(Bce)	
Kernel	5

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

Dictionary of the permissions is divided into two parts: on the left the tree of permissions groups is displayed, on the right – the list of permissions of chosen from the left group.

Dictionary records can be filtered according to *Permission name* (*Permission name*).

For the new permission it is necessary to specify:



Permission, 2

Permission: 2 no files en OK Save Cancel

Name	Developer
Description	
Group	

- *Name* – permission name;
- *Description* – description of the action/functionality provided by the permission;
- *Group* – group to which the permission belongs.

Addition of records into the dictionary of permissions makes sense only if the developer also writes the corresponding code that checks these permissions.

Permissions to access system DB tables

The system database Ultimate Solid consists of two circuits: *kernel scheme* and *application scheme*. The metadata describing the business logic and the business objects of the system are stored in the *kernel scheme*. In *the application scheme* tables are created and data of the application area (dictionaries, link tables, table parts of the documents, total, etc).

If the access to database*application scheme* tables is determined by granting permissions to perform CRUD operations (Create, Read, Update, Delete) immediately over objects (dictionaries, document subtypes, etc.), then for the tables of the database *kernel scheme* – system tables – such access is determined by granting the relevant *permissions* (*Permissions*). In addition, access is closed for all users to a number of service tables of the *kernel scheme.*

Access to system tables follows is subject to the following rules.

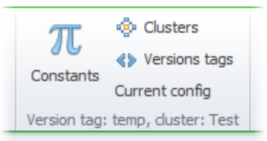
1. Any user, regardless of his role and presence of any *permissions* can read (perform Read operation) data from all tables.

   Also, any user may perform all operations (CRUD) with the following tables:
   - ATTACHMENTS – enclosures attached to metadata objects;
   - ATTACHMENT_TYPES – types of enclosures;
   - NOTIFICATIONS – notifications;
   - PARENT_DOCUMENTS – relations parent–child between documents;
   - SPELL_CHK_WORDS – the vocabulary added by the users during spell check.
2. The user having the **Administrator** *permission* can (apart from reading all tables) to do the following additional operations for the following tables:
   - operations 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;
   - operations UPDATE, DELETE:
     - PRINT_PACK_QUEUE;
     - PRINT_QUEUE;
     - PRINT_QUEUE_LOCKS;
     - PRINT_TASK_QUEUE;
   - operation DELETE:
     - USER_SETTINGS;
   - operation UPDATE:
     - TASKS (only for property EXEC TIME).
3. The user having the **Developer** *permission* can perform all operations (CRUD) over all tables except the following:
   - only operations READ, UPDATE, DELETE:
     - PRINT_PACK_QUEUE;
     - PRINT_QUEUE;
     - PRINT_QUEUE_LOCKS;
     - PRINT_TASK_QUEUE;
   - only operations READ, DELETE:
     - USER_SETTINGS;

- only operation READ:
  - GENDERS;
  - LAYOUT_TYPES;
  - OBJECT_TYPES;
  - PARAM_MODES;
  - PRINT_PACKAGE_STATES;
  - PRINT_TASK_STATES;
  - PROPERTY_TYPES;
  - SCRIPT_TYPES;
  - SPELL_CHK_DICT_TYPES.

4. The user having the **Calendar management** *permission* can perform all operations (CRUD) with the following tables:
   - CALENDAR_DAY_STATUSES;
   - CALENDAR_STATUSES.
5. The user having the *permission* to **Edit constants** has an opportunity to perform (CRUD) with the following tables:
   - CONSTANT_GROUPS;
   - CONSTANTS.
6. The user having the *permission* to Edit exception translations can perform CRU operations with the following tables:
   - EXCEPTION_TRANSLATIONS;
   - EXCEPTION_TRANSL_TEXTS.
7. The user having the *permission* to **Edit metadata translations** can perform CRU operations with METADATA_TRANSLATIONS table and all operations (CRUD) with SPELL_CHK_DICTIONARIES tabl.
8. The user having the *permission* to **Edit metadata user comments** can perform CRU operations with METADATA_USER_COMMENTS table.
9. The user having the *permission* to **Manage print queues** can perform CRU operations with the following tables:
   - PRINT_PACK_QUEUE;
   - PRINT_QUEUE;
   - PRINT_QUEUE_LOCKS;
   - PRINT_TASK_QUEUE.
10. The user having the *permission* to **Edit UI templates** has an opportunity to perform all operations (CRUD) with the UI_TEMPLATES table.

System configuration tools



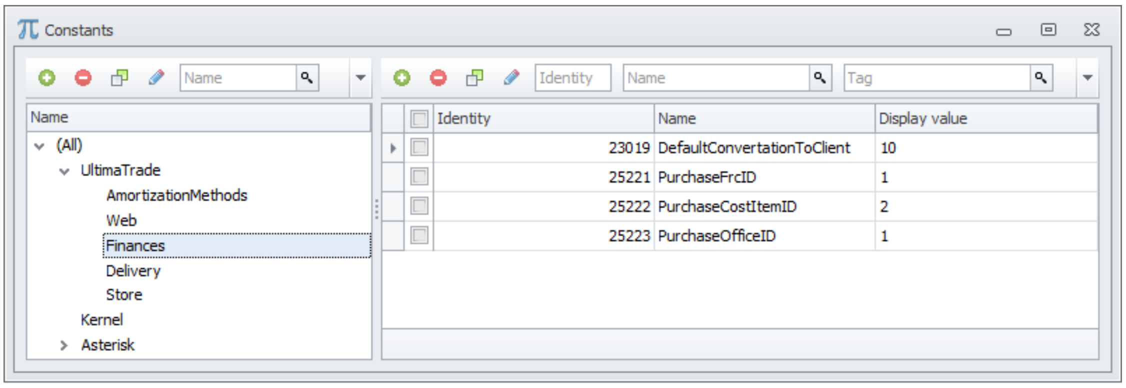
Tools of group besides dictionaries constants (*Constants*), clusters (*Clusters*) and versions tags (*Versions tags*) contain the following functionality:

- *Current config* – configuration of the cluster with which the started client works. By clicking the button the form of editing of the configuration cluster will be opened;
- *Version tag* – version tag, from which the application server is running, with which the client application works. In the given example it is a tag *temp*;
- *cluster* – cluster name, in which the application server is included, with which the client application works. In the given example it is a cluster *Test*.

Constants

Constants are meant for storing variables used in the software code which can vary over time. For example, this can be the date of closing the reporting period, direct e-mail text, etc.

You can view the existing and create new constants in the Constants dictionary:



Constants

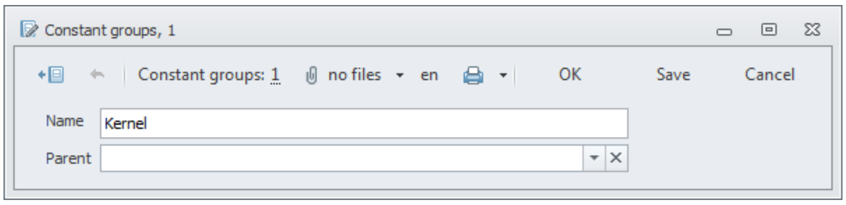
Name
- (All)
  - UltimaTrade
    - AmortizationMethods
    - Web
    - Finances
    - Delivery
    - Store
  - Kernel
  - Asterisk

Identity	Name	Display value
23019	DefaultConvertationToClient	10
25221	PurchaseFrcID	1
25222	PurchaseCostItemID	2
25223	PurchaseOfficeID	1

The dictionary of constants is divided into two parts: on the left, the tree of the constants groups is displayed, and on the right – the list of constants of the group selected on the left.

The groups of constants can be filtered by *Name* of the group (*Name*), and constants – by Constant *Name(Name*) and *Tags* (*Tag*).

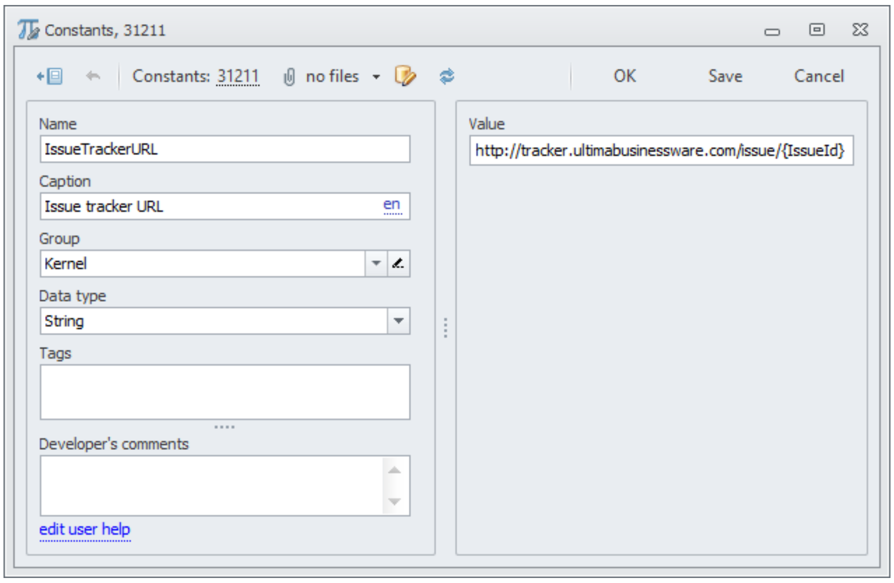
The form of editing the group of constants allows to set its *Name* (*Name*) and select the *Parent* in the tree (*Parent*). In case the parent is not selected, the group will be located in the uppermost level of the tree:



Constant groups, 1

Name	Kernel
Parent	

The constant edit form allows to set the following properties:



Constants, 31211

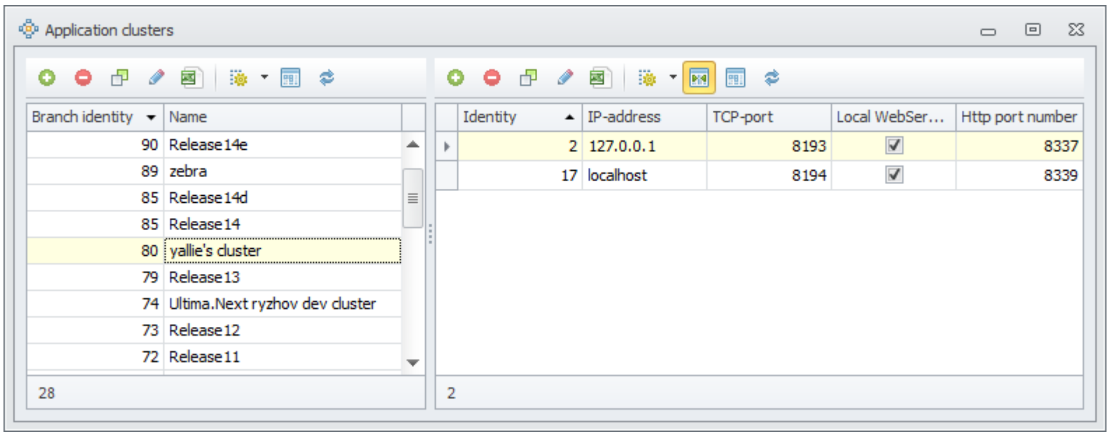
Name	IssueTrackerURL
Caption	Issue tracker URL
Group	Kernel
Data type	String
Tags	
Developer's comments	
Value	http://tracker.ultimabusinessware.com/issue/{IssueId}

- *Name* – name of the constant used in scripts;
- *Caption* — the localized name of the constant;

- *Group* – a group that the constant belongs to. When you create a new constant as property value the *Constants group marked in the directory is selected*;
- *Data type* – type of the constant. Constant based on type can have the following values:
  - *Integer number* – an integer number, 64-bit;
  - *Decimal number* – a fractional number, 96-bit;
  - *Text* – text not more than 2Kb;
  - *String* – string having size not more than 2Kb;
  - *HTML text* – text having unlimited size;
  - *Date with time* - date and time (is broadcast on time zones);
  - *Date* – data (without time);
  - *Flag* – used for logical values;
  - *Dictionary record ID* – dictionary record;
  - *Records id list* – list of dictionary records;
- *Tags* – tags used for description of the constant functionality. Used for searching the objects implemented for certain functionality associated with such tag.
  The tag is added by keys Space or Enter. Delete – by button ⊗ after the tag. As the gap is used for the tag input it is possible to replace it with characters "_" or "-" in tags with the name of several words;
- *Developer's comments* – comments of the application developer;
- *edit user help* – comment to the object which the end user can see in the form of a hint which drops down after mouseover. The comment is entered for any language of the system;
- *Value* – value of the constant.

Servers and clusters

Management of content and cluster configuration serves via "Application clusters" form.


Application clusters

Branch identity	Name
90	Release14e
89	zebra
85	Release14d
85	Release14
80	yallie's cluster
79	Release13
74	Ultima.Next ryzhov dev cluster
73	Release12
72	Release11

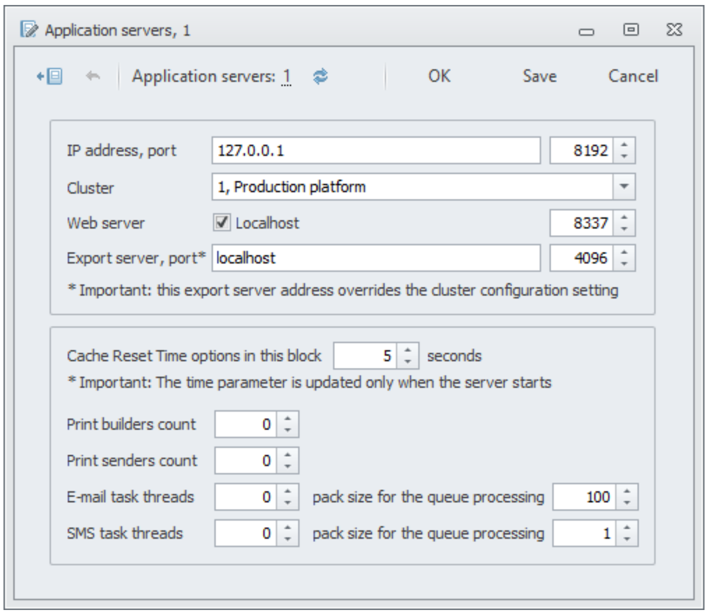
28

Identity	IP-address	TCP-port	Local WebSer...	Http port number
2	127.0.0.1	8193	✓	8337
17	localhost	8194	✓	8339

2

The form is divided into two parts: on the left there is a list of clusters, on the right servers. Clusters list is shown on the left, servers is shown on the right. Selecting the print server on the left filters the list of servers on the right, leaving only those belonging to it.

The edit form of the client application allows setting the following properties:


Application servers, 1

Application servers: 1 | OK | Save | Cancel

IP address, port	127.0.0.1	8192
Cluster	1, Production platform	
Web server	☑ Localhost	8337
Export server, port*	localhost	4096

* Important: this export server address overrides the cluster configuration setting

Cache Reset Time options in this block 5 seconds

* Important: The time parameter is updated only when the server starts

Print builders count	0		
Print senders count	0		
E-mail task threads	0	pack size for the queue processing	100
SMS task threads	0	pack size for the queue processing	1

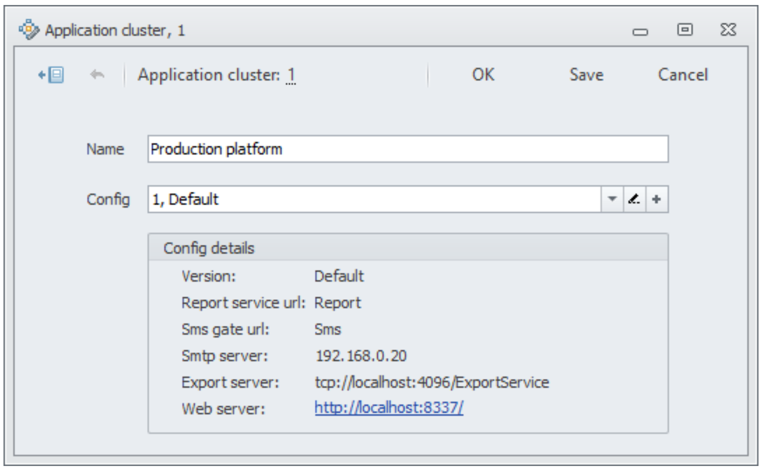
- *IP address, port* – address and port of application server;
- *Cluster* – to select the cluster, which will include the application server;
- *Export server* – the address and the server port of export, which will interact with the application server. If the field is empty, the export server address will be taken from the cluster settings;
- *Web server* – the address and the port of the local web server. With set flag *Localhost* web server will be launched locally.
  If the web server had been deployed to the concrete address and available for everyone, the flag *Localhost*would be removed. The web server is the REST/SOAP service which are used by company site or by sided applicaations for the integration. The server must be tuned in the following way: the appropriate applications (sites, etc.) can connect with the server.

If two applications servers are launched on the one computer and two servers have switching on web servers, you should be sure that web servers will have the different ports to avoid the conflicts.

- *Cache reset time —the lifetime of the cache is loaded the settings of the application server.*
- *Print builders count — the number of threads available for processing* deferred (asynchronous) printing tasks. If valuePrint builders countis equal to 0, the application server will not process the pending print jobs;
- *Print senders count — the number of threads available for sending print jobs to the printing tasks.* If valuePrint senders countvalue is 0, the application server will not send it for printing.
  It is not recommended that the total number of additional stream processing and sending print jobs exceeded the number of virtual computer cores, on which the application server runs.
- *E-mail task threads— number of threads to handle the queue sending e-mail messages.*
- *SMS task threads* — number of threads to handle the queue of SMS messages.
- *Pack size for the queue processing — batch sizes for the processing of the distribution queues.* The batch size determines how many messages will be downloaded for processing at the same time. The smaller the batch size, the more reliable, but slower delivery is. When an error occurs, the list server will send out a batch of messages again, so that some messages can be sent again. To eliminate this situation, set the size of the package to 1.

At least one of the application servers in the cluster must have at least one thread to process pending print, so that those jobs can be processed, and/or one thread for sending print jobs to the print server. Similarly, in the cluster there must be at least one server with enabled tasks of processing queue of e-mail and SMS.

The cluster editing form allows to set the following properties:


Application cluster, 1

Application cluster: 1 — OK — Save — Cancel

Name	Production platform
Config	1, Default

Config details

Version:	Default
Report service url:	Report
Sms gate url:	Sms
Smtp server:	192.168.0.20
Export server:	tcp://localhost:4096/ExportService
Web server:	http://localhost:8337/

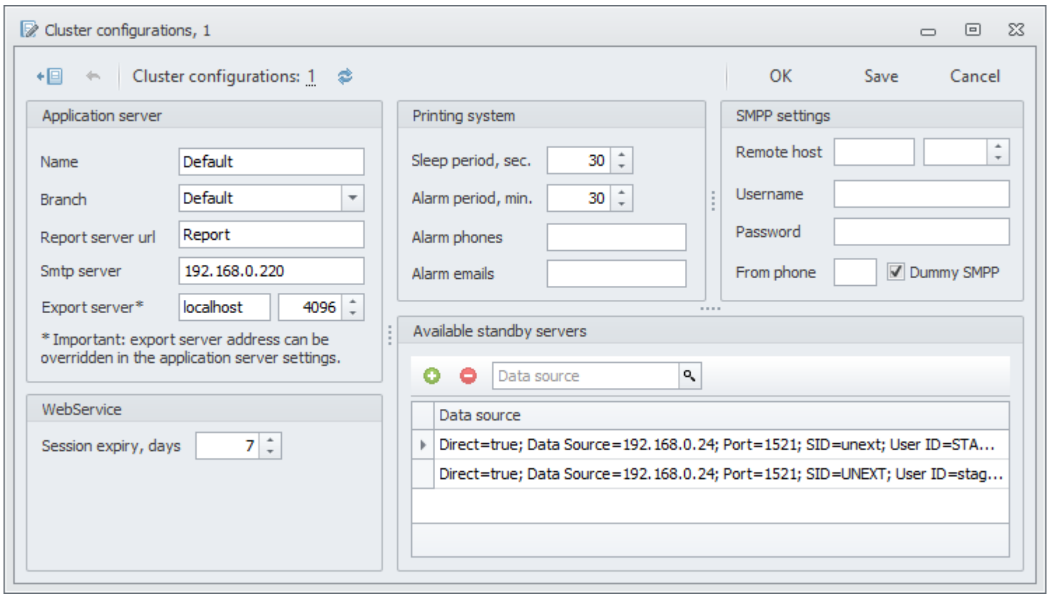
- *Name – cluster name;*
- *Cluster —to select the cluster configuration.* Short description of selected cluster is situated under the configuration– *Config details*. The element of configuration choice control let:

▾ - to select the configuration of the existing;

✎ - to edit the selected configuration

+ - to create new configuration

In cluster configuration editing form the characteristics are grouped:


Cluster configurations, 1

Cluster configurations: 1 — OK — Save — Cancel

Application server

Name	Default
Branch	Default
Report server url	Report
Smtp server	192.168.0.220
Export server*	localhost 4096

* Important: export server address can be overridden in the application server settings.

WebService

Session expiry, days	7

Printing system

Sleep period, sec.	30
Alarm period, min.	30
Alarm phones	
Alarm emails	

SMPP settings

Remote host	
Username	
Password	
From phone	☑ Dummy SMPP

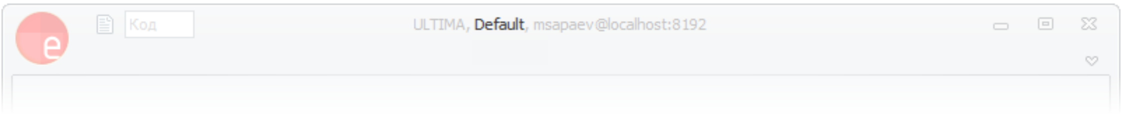
Available standby servers

Data source

Data source
Direct=true; Data Source=192.168.0.24; Port=1521; SID=unext; User ID=STA...
Direct=true; Data Source=192.168.0.24; Port=1521; SID=UNEXT; User ID=stag...

Application server is the settings of print system:

- *Name* — name of driver; This name is displayed in the main form title of application.


Код

ULTIMA, Default, msapaev@localhost:8192

- *Branch— branch configuration, which will work in a cluster of the application server.*

The change of cluster configuration is used by the incoming in this cluster by applications servers hurriedly.

It means that after the new version tag selected in the cluster configuration and saved, the next appeal of client application for the applications server, which is incoming to this cluster, will be the appeal for the new metadata version. For example, the editing document or dictionary record, which is opened in the old version, will be saved in the new version. In some cases it can cause the different conflicts and errors.

It is strongly recommended to carry out such changes at the stopped application server (or, at least, in the period of lowest user activity).

- *Report server url* - the report server.
- *Smtp server* –the address of the mail server (the standard port is 25);
- *Export server — the address and the port of export server.* If you have configured your application server (see above) specified a different server address for exports, then the priority goes to the address, which was specified in the application server.

Print system – settings of print system:

- *Sleep period, sec.* — the inaction period of print streams on the applications server in seconds. If at the next address to the database the applications server doesn't get the print task to the processing or the dispatch, it will wait a specified period of time before the next appeal. Also in case of the sending failure of task on the print server, for example, because of its inaccessibility, the applications server will wait a specified period of time before the retry of task dispatch.
- *Alarm period, min.*– the dispatch periodicity of alarm notification about print server problems or printers in minutes. The notification will be sent in case of error of task dispatch on the print servers or the directly print effort to the administrators. The delivery of notifications won't be each time in case of error, but accumulatively with the frequency not more than a specified.
- *Alarm phones* –the telephones list of administrators for the delivery of alarm notifications by SMS;
- *Alarm emails* –the list of the mailing address of the administrators for the delivery of alarm notifications by the mails.

SMPP settings – the settings of the SMS portal:

- *Remote host* – the address and port of portal;
- *Username* – the login for the connection to the portal;
- *Password* – the password to the connection to the portal;
- *From phone* – the number which will be used like the number of the sender in the messages;
- *Dumpy SMPP* – the plug to the dispatch check of SMS in the handlers. With the set of the flag, the SMS text is written to the server log instead the dispatch to the portal.

WebService – the settings of web services:

- *Session expiry, days* – the life time of the web service session in days.

Available standby servers – the list of spare Oracle servers, which are available for the cluster servers.

Version tags

In the system Ultimate Solid there is a metadata versioning mechanism which is conveniently represented as a tree.

Each metadata version can be tagged which are divided into:

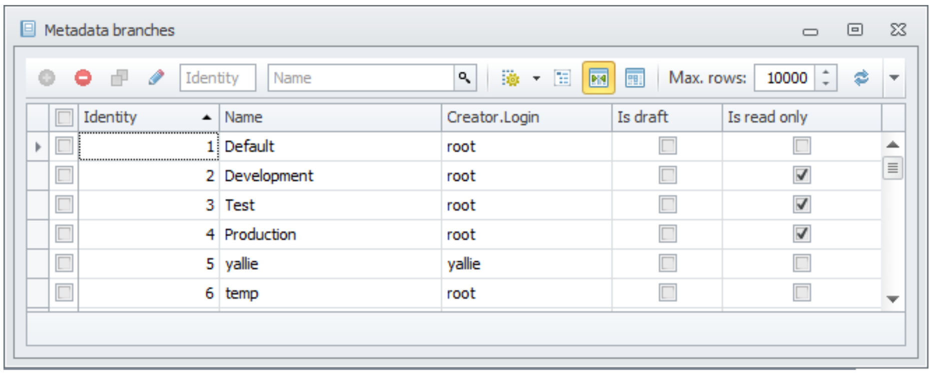
- *branch-tags* – are used to mark the last node of a branch in the version tree. Each version branch (its last node) must be marked by its unique branch-tag. Usually practice is that every application developer has its branch-tag, with which he marks his development branch.
  One version of metadata can be tagged with more than one branch-tag.

Initially the system has got the only *Default* branch-tag with which the main branch of metadata version tree is marked;
- *normal tags* – any metadata version in a branch can be marked with them (except the last). And, one version can be marked with more than one normal tag.

No more than one version of metadata can be marked with any tag. When other version is marked with the same tag it automatically is deselected from previous.

In addition to *Default*branch-tag in system there are two more (normal) tags:
- *Production* – a tag that is used to mark the last stable metadata version intended for operation of basic system users;
- *Debug* – a tag that is used to mark the metadata version for diagnostic, for example, selectively by several users or department of testing.

It is possible to view existing and to create new tags which mark metadata versions in the Version tags dictionary:



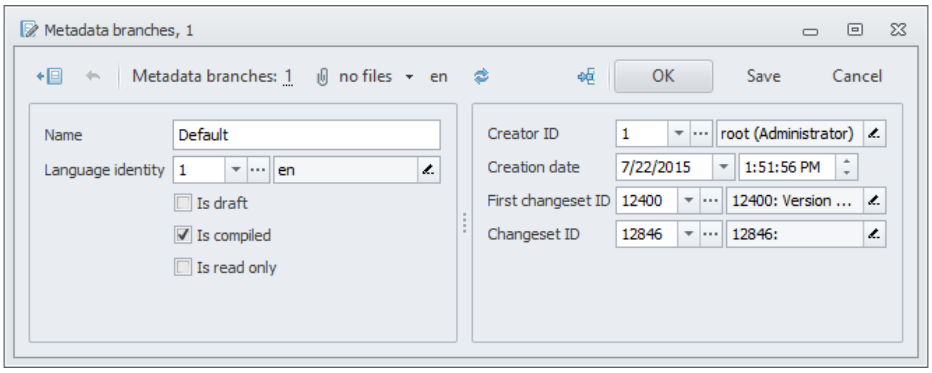
Metadata branches

Max. rows: 10000

Identity	Name	Creator.Login	Is draft	Is read only
1	Default	root	☐	☐
2	Development	root	☐	✓
3	Test	root	☐	✓
4	Production	root	☐	✓
5	yallie	yallie	☐	☐
6	temp	root	☐	☐

The dictionary records can be filtered by Tag *name (Name*).

The tags have the following properties:



Metadata branches, 1

Metadata branches: 1 · no files · en · OK · Save · Cancel

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* – tag name;
- *Version* – metadata version marked with this tag:
  - the version can not be changed for a branch-tag;
  - it is impossible to select an unrecorded version of metadata for a normal tag;
  - it is also impossible to select the version which metadata weren't compiled for a normal tag .

That change of the version for the tag specified in a cluster configuration began to work, it is necessary:
- to restart the application servers entering this cluster;
- optionally, if after restarting of the application servers there are errors, to restart also client applications.

For this reason it is strongly recommended to carry out changes of the version for the *Production* tag in the period of the smallest activity of users.

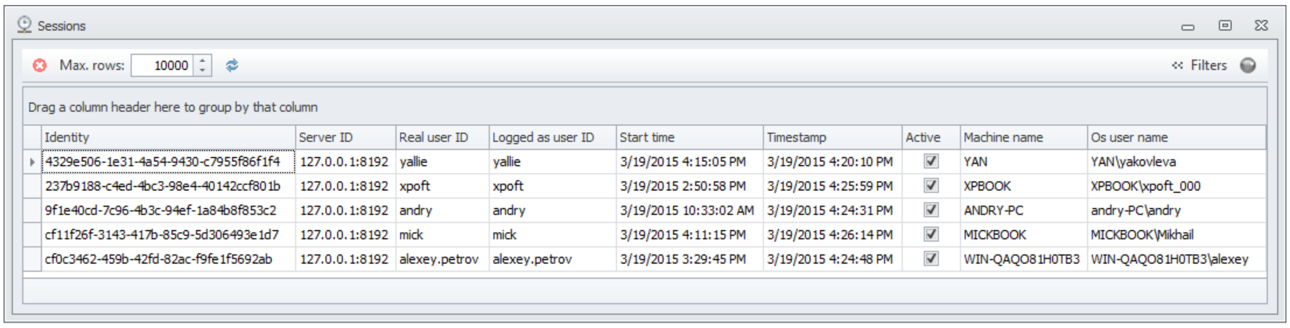
- *Owner user* – the owner user of the tag. The optional property having information character and intended for specifying of the application-oriented developer who carries development under this branch-tag;
- *Draft branch* – it is impossible to push on the *Default branch changes from the metadata branch that is marked by a branch-tag with this flag* .

In the tag dictionary it is possible to create only normal tags. It is only possible to change a name and an owner for branch-tags.

Monitor tools

Sessions

The list of all connections to application servers can be found in the dictionary "Sessions":



Sessions

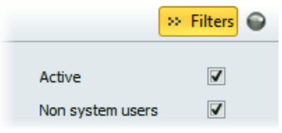
Max. rows: 10000

Filters

Drag a column header here to group by that column

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	yallie	yallie	3/19/2015 4:15:05 PM	3/19/2015 4:20:10 PM	✓	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	✓	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	✓	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	✓	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	✓	WIN-QAQO81H0TB3	WIN-QAQO81H0TB3\alexey

Initially, the Sessions dictionary is opened with a switched on filter which settings limit the displayed list only by active sessions of non-system users.



Filters

Active	✓
Non system users	✓

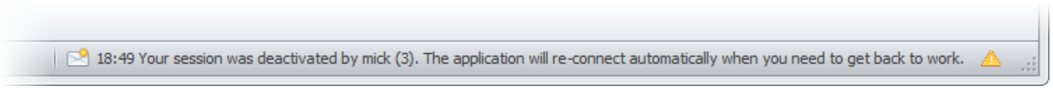
Each session has the following properties:

- *Identity* – session ID;
- *Server ID*– application server that opened the session;
- *Real user ID*– an user login under whose password connection to the application server was carried out;
- *Logged as user ID*– the user login under which connection to the application server was carried out (if the user logged into the client application under login of other user, using for this purpose his login/ password).
  For example, *admin* logs into the client application with his registration data under the *user login* to check settings of his role. Then in the opened session in the property *User who logged in using his own password* there will be the *admin login* and in the property *User who logged in* – the *user login*. In a case when the administrator logs under his name, in both properties there will be the *adminlogin*;
- *Start time* – session start time;
- *Timestamp* – time of the last access to the date base;
- *Active* – a session active flag. Sessions without a flag *Active* – are closed;
- *Machine name* – the user machine name from which the session was started;
- *OS user name* – the user name in the operating system on the machine on which the session was started.

By clicking the button in the toolbar the selected session can be deasserted (to clear the flag *Active*). After that the session is non-active. It helps with two typical situations:

- the application was complete incorrectly and session remained active;
- the user started several program instances and away from keyboard for a long time.

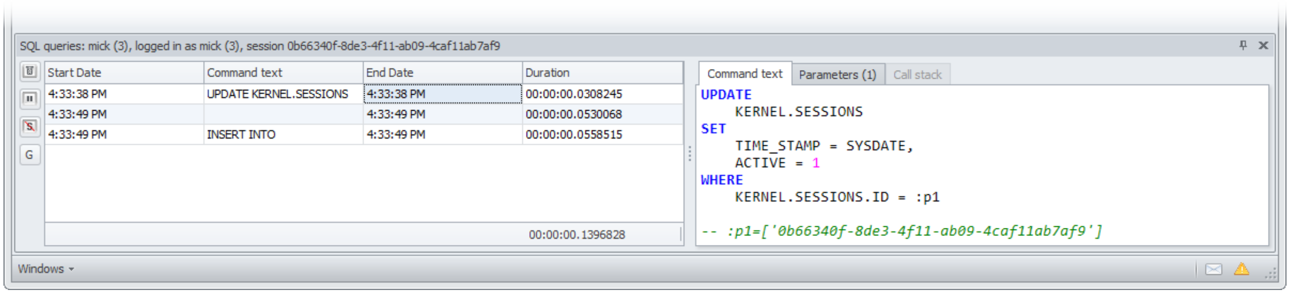
If you deactivate session of a live user, he will receive a message that the current session was suspend by the administrator. When you first need the program will be able automatically to reconnect and to resume your operation as nothing's the matter:



18:49 Your session was deactivated by mick (3). The application will re-connect automatically when you need to get back to work.

Incorrect closing of the client application can lead to the fact that the flag of the session activity (*Active*) won't be cleared. In this case, you can determine the closed sessions by time of the last access to the server application (*Last access time*). The client application reports about the presence (activity) to the application server each five minutes even in the absence of the user activity. Thus, all sessions with significantly large time of the last access can be considered as closed.

SQL query tracer is opened for it in the list by double left click on the session:



SQL queries: mick (3), logged in as mick (3), session 0b66340f-8de3-4f11-ab09-4caf11ab7af9

Start Date	Command text	End Date	Duration
4:33:38 PM	UPDATE KERNEL.SESSIONS	4:33:38 PM	00:00:00.0308245
4:33:49 PM		4:33:49 PM	00:00:00.0530068
4:33:49 PM	INSERT INTO	4:33:49 PM	00:00:00.0558515
			00:00:00.1396828

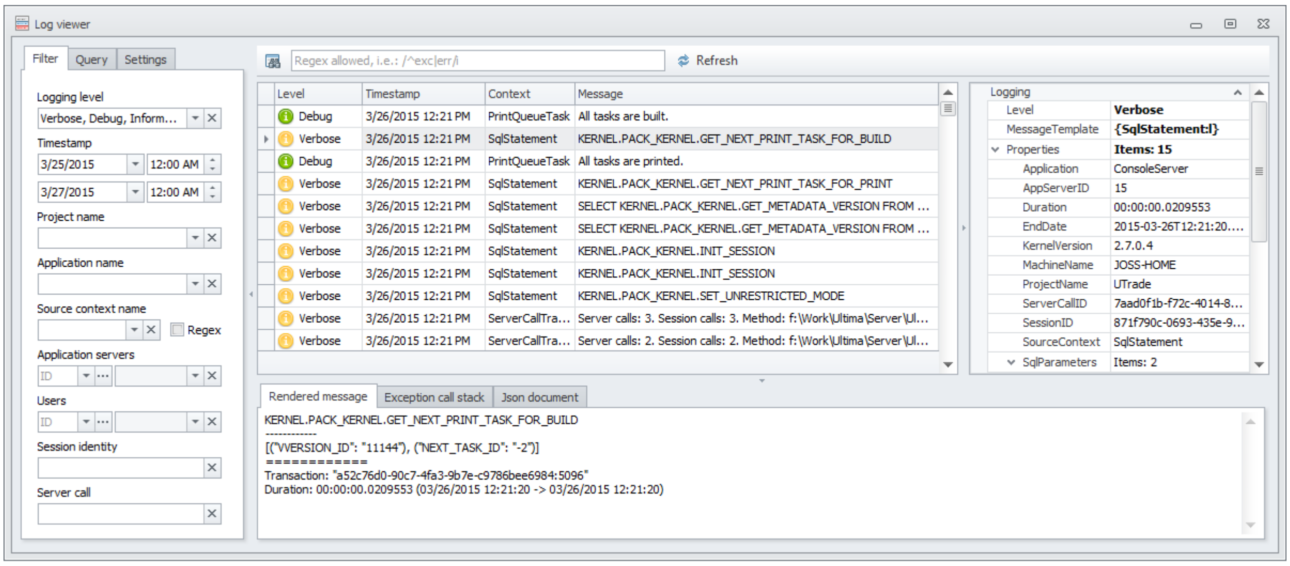
```
UPDATE
    KERNEL.SESSIONS
SET
    TIME_STAMP = SYSDATE,
    ACTIVE = 1
WHERE
    KERNEL.SESSIONS.ID = :p1

-- :p1=['0b66340f-8de3-4f11-ab09-4caf11ab7af9']
```

Soft logs

A Serilog-based structured central logging is implemented in the system Ultimate Solid (http://serilog.net/):

- the system-integrated log analyzer "Log viewer" described below works only with the MongoDB database and implies a specific structure of document-events generated by the Serilog library;
- Mongo DB's start and setup described in section Hardware;
- setup of applications to log using Serilog described in the same place;
- by default, the system logs:
  - all SQL queries (log contains: time, text, parameters, transaction and server call);
  - all exceptions at the time of occurrence – *FirstChanceException*. That is necessary because the application server does not always act on behalf of a client. An exception may occur in tasks, reverse calls and explicitly created streams; such exceptions may cause a server closedown and will not be propagated to the client. To save such information, it is needed to log all exceptions.
  - any *throw* operator. Due to this reason, in a log, one may find exceptions intercepted by something and invisible to the end user;
- the logging of other events needs to be initiated by an application programmer. This can be done via the interfaces *ILogger* and *ILogManager* described in the developer documentation (section "Developer's tools / Scripts / Special managers".

The "View logs" command opens "Log viewer", a form to view events.



Logging level: Verbose, Debug, Inform...
Timestamp: 3/25/2015 12:00 AM – 3/27/2015 12:00 AM
Project name
Application name
Source context name ☐ Regex
Application servers: ID
Users: ID
Session identity
Server call

Regex allowed, i.e.: /^exc|err/i

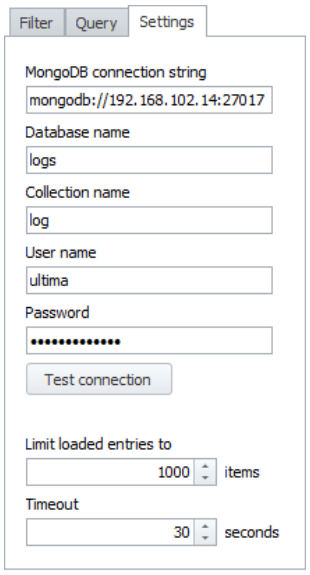
Level	Timestamp	Context	Message
Debug	3/26/2015 12:21 PM	PrintQueueTask	All tasks are built.
Verbose	3/26/2015 12:21 PM	SqlStatement	KERNEL.PACK_KERNEL.GET_NEXT_PRINT_TASK_FOR_BUILD
Debug	3/26/2015 12:21 PM	PrintQueueTask	All tasks are printed.
Verbose	3/26/2015 12:21 PM	SqlStatement	KERNEL.PACK_KERNEL.GET_NEXT_PRINT_TASK_FOR_PRINT
Verbose	3/26/2015 12:21 PM	SqlStatement	SELECT KERNEL.PACK_KERNEL.GET_METADATA_VERSION FROM ...
Verbose	3/26/2015 12:21 PM	SqlStatement	SELECT KERNEL.PACK_KERNEL.GET_METADATA_VERSION FROM ...
Verbose	3/26/2015 12:21 PM	SqlStatement	KERNEL.PACK_KERNEL.INIT_SESSION
Verbose	3/26/2015 12:21 PM	SqlStatement	KERNEL.PACK_KERNEL.INIT_SESSION
Verbose	3/26/2015 12:21 PM	SqlStatement	KERNEL.PACK_KERNEL.SET_UNRESTRICTED_MODE
Verbose	3/26/2015 12:21 PM	ServerCallTra...	Server calls: 3. Session calls: 3. Method: f:\Work\Ultima\Server\Ul...
Verbose	3/26/2015 12:21 PM	ServerCallTra...	Server calls: 2. Session calls: 2. Method: f:\Work\Ultima\Server\Ul...

Logging	
Level	**Verbose**
MessageTemplate	**{SqlStatement:l}**
Properties	**Items: 15**
Application	ConsoleServer
AppServerID	15
Duration	00:00:00.0209553
EndDate	2015-03-26T12:21:20....
KernelVersion	2.7.0.4
MachineName	JOSS-HOME
ProjectName	UTrade
ServerCallID	7aad0f1b-f72c-4014-8...
SessionID	871f790c-0693-435e-9...
SourceContext	SqlStatement
SqlParameters	Items: 2

```
KERNEL.PACK_KERNEL.GET_NEXT_PRINT_TASK_FOR_BUILD
------------
[("VVERSION_ID": "11144"), ("NEXT_TASK_ID": "-2")]
=============
Transaction: "a52c76d0-90c7-4fa3-9b7e-c9786bee6984:5096"
Duration: 00:00:00.0209553 (03/26/2015 12:21:20 -> 03/26/2015 12:21:20)
```

When first run, it is needed to setup a connection to MongoDB in the left part of the form ("Settings" tab):

- *Mongo DB connection string* – database address in the format: *mongodb://127.0.0.1:27017*;
- *User name* and *Password* – each MongoDB user shall have his own account with the privileges to read logs;
- *Limit loaded entries to* – limitation of events to be shown in the form;
- *Timeout* – database response wait period in seconds.



MongoDB connection string: mongodb://192.168.102.14:27017
Database name: logs
Collection name: log
User name: ultima
Password: •••••••••••••
Limit loaded entries to: 1000 items
Timeout: 30 seconds

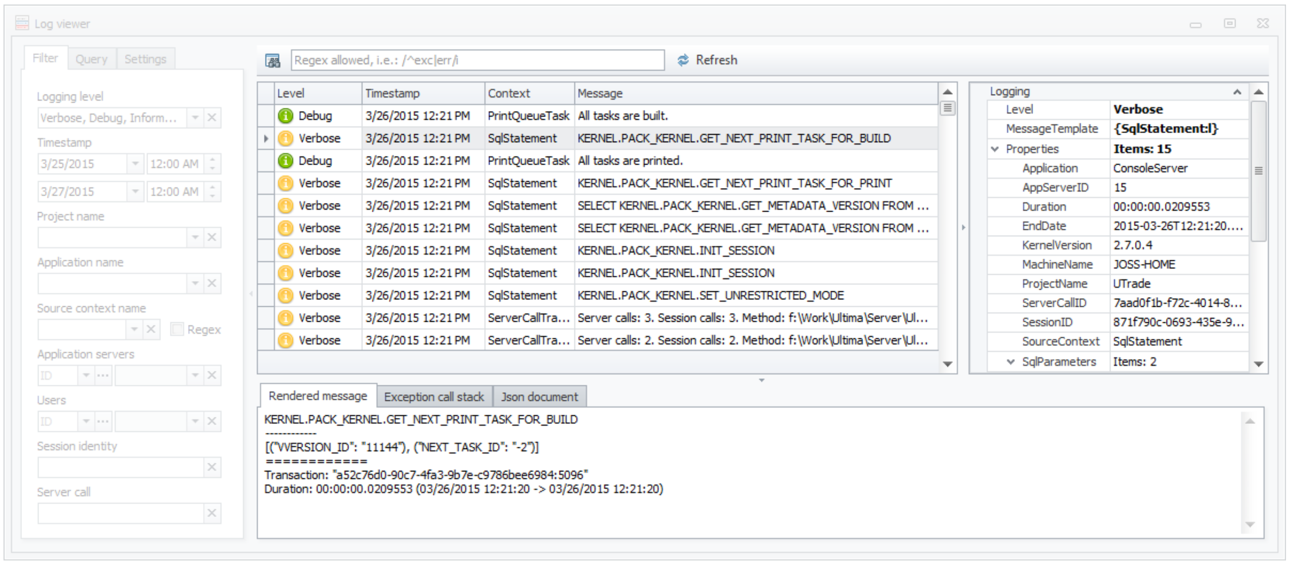
Settings for filtering are in the "Filter" tab.

During the process of logging, events are automatically supplemented with the execution context, that is: user code, IDs of transaction, current call and session. A session code helps to monitor all events made in the system by a particular user in a specified time period. Using a call ID, it is possible to find out, which SQL queries were executed during a server call, and so on.

In "Query" tab, a JSON query to MongoDB is given; the query is generated according to settings in the "Filter" tab. If needed, one can write arbitrary queries to the storage having changed the query manually.

Queries to Mongo are formulated in the form of example documents. In the example, one specifies either the values of desired fields (e. g., *Level: Debug*), or the special comparison operators (*$in*, *$gt*, *$gte*, *$lt*, *$lte*, *$regex* etc.). To limit the values of the document enclosed (e. g., *Properties.UserID*), dot-notation is used. More details on the subject provided in the MongoDB documentation.

A list of filtered events is displayed in the right part of the form. The events are arranged in the order of descending date. Displayed events can be additionally filtered, including through regular expressions (detailed description of the syntax can be found on MSDN website: eng/rus), in the search string in the upper part of the form. The filter is applied by clicking the button " Refresh":



	Level	Timestamp	Context	Message
	Debug	3/26/2015 12:21 PM	PrintQueueTask	All tasks are built.
▸	Verbose	3/26/2015 12:21 PM	SqlStatement	KERNEL.PACK_KERNEL.GET_NEXT_PRINT_TASK_FOR_BUILD
	Debug	3/26/2015 12:21 PM	PrintQueueTask	All tasks are printed.
	Verbose	3/26/2015 12:21 PM	SqlStatement	KERNEL.PACK_KERNEL.GET_NEXT_PRINT_TASK_FOR_PRINT
	Verbose	3/26/2015 12:21 PM	SqlStatement	SELECT KERNEL.PACK_KERNEL.GET_METADATA_VERSION FROM ...
	Verbose	3/26/2015 12:21 PM	SqlStatement	SELECT KERNEL.PACK_KERNEL.GET_METADATA_VERSION FROM ...
	Verbose	3/26/2015 12:21 PM	SqlStatement	KERNEL.PACK_KERNEL.INIT_SESSION
	Verbose	3/26/2015 12:21 PM	SqlStatement	KERNEL.PACK_KERNEL.INIT_SESSION
	Verbose	3/26/2015 12:21 PM	SqlStatement	KERNEL.PACK_KERNEL.SET_UNRESTRICTED_MODE
	Verbose	3/26/2015 12:21 PM	ServerCallTra...	Server calls: 3. Session calls: 3. Method: f:\Work\Ultima\Server\Ul...
	Verbose	3/26/2015 12:21 PM	ServerCallTra...	Server calls: 2. Session calls: 2. Method: f:\Work\Ultima\Server\Ul...

Logging	
Level	**Verbose**
MessageTemplate	**{SqlStatement:l}**
Properties	**Items: 15**
Application	ConsoleServer
AppServerID	15
Duration	00:00:00.0209553
EndDate	2015-03-26T12:21:20....
KernelVersion	2.7.0.4
MachineName	JOSS-HOME
ProjectName	UTrade
ServerCallID	7aad0f1b-f72c-4014-8...
SessionID	871f790c-0693-435e-9...
SourceContext	SqlStatement
SqlParameters	Items: 2

```
KERNEL.PACK_KERNEL.GET_NEXT_PRINT_TASK_FOR_BUILD
------------
[("VVERSION_ID": "11144"), ("NEXT_TASK_ID": "-2")]
============
Transaction: "a52c76d0-90c7-4fa3-9b7e-c9786bee6984:5096"
Duration: 00:00:00.0209553 (03/26/2015 12:21:20 -> 03/26/2015 12:21:20)
```

The events of each relevance level (*Logging level*) begin in the list with a corresponding icon:

- 🟡 – Verbose level;
- 🟢 – Debug level;
- 🔵 – Information level;
- 🟣 – Warning level;
- 🔴 – Error level;
- ❌ – Fatal level.

In the "Logging" section, properties of a selected event are displayed in a complicated tree structure: the exceptions are saved together with the enclosed exceptions and the Data properties set. Below is the detailed information on an event selected in the list:

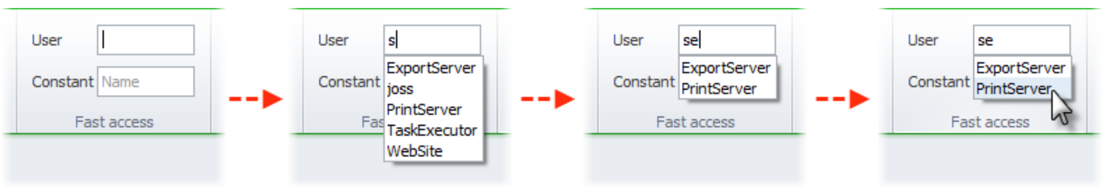
- in "Rendered message" tab, a text description of the event is shown;
- in "Exception call stack" tab, an exception call stack is shown;
- in "JSON document" tab, the event's JSON presentation is displayed.

Fast access tools

Group tools are used for fast access to the objects:

- *User* – user choice for his login (*Login*);
- *Constant* – choice of a constant according to the name (*Name*).

When adding the values in the tool field the auto substitution function is able, that offers a choice from the list of objects, the name of which includes the adding fragment:

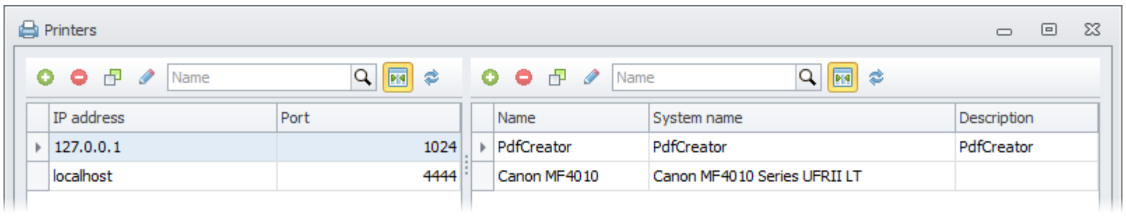


The chosen object is opened in the form of editing of the corresponding dictionary. The choice is carried out by left click by on the object in the list of auto substitution or on pressing a key Enter (in this case the added name has to coincide completely with the value of the corresponding property of the object).

Printing tools

Printers

Control of printers and print servers is carried out through the "Printers" form:



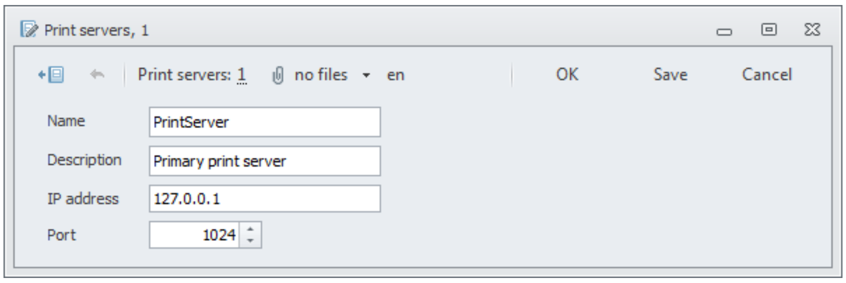
IP address	Port
127.0.0.1	1024
localhost	4444

Name	System name	Description
PdfCreator	PdfCreator	PdfCreator
Canon MF4010	Canon MF4010 Series UFRII LT	

Window of the dictionary is divided into two parts: on the left the list of print servers is displayed, on the right - printers that belong to them. The choice of print server on the left filters the list of printers on the right, leaving only those that belong to it.

Print servers can be filtered according to the *Name* (*Name*), and also according to the *Name* printers be can filtered (*Name*).

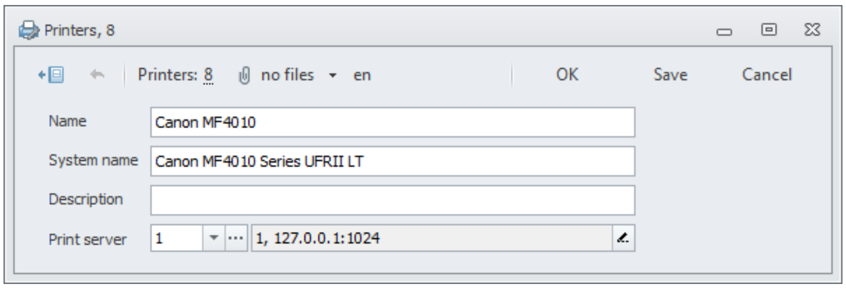
The form of editing of print server allows to set, besides the *Name* (*Name*) and *Description* (*Description*), *IP address* of the computer (*IP address*) and *Port* (*Port*), on which it functions:



Name	PrintServer
Description	Primary print server
IP address	127.0.0.1
Port	1024

Print server identifier, displayed in the toolbar edit form, is used at print server application setting (parameter *PrintServerID* in the file of configuration PrintServer.exe.config).

Form of printer editing allows to set:

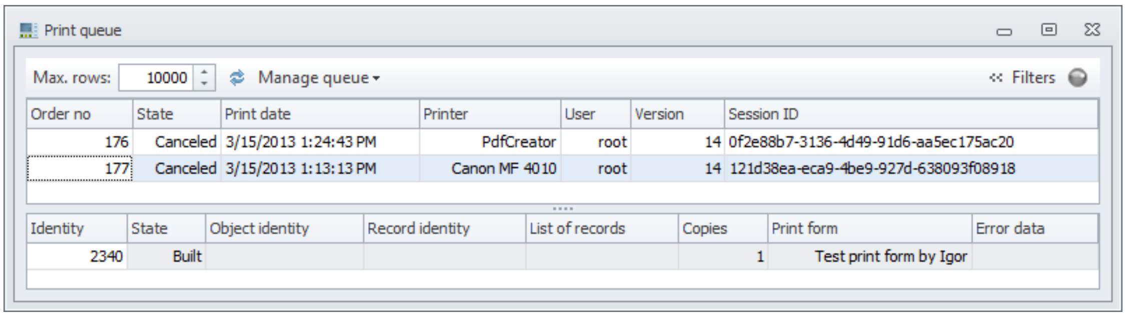


Name	Canon MF4010
System name	Canon MF4010 Series UFRII LT
Description	
Print server	1 · 1, 127.0.0.1:1024

- *Name* – name of the printer which will be displayed in the interface of the system Ultimate Solid;
- *System name* – system name of the printer, has to coincide with the name of the corresponding printer in an operating system of the computer on which PrintServer functions;
- *Description* – office field, serving for detailed, if it is necessary, description of the printer (this information is not displayed anywhere any more);
- *Print server* – print server on which the printer functions.

Print queue

Print queue management is done via the form "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	

Queue window is divided into two parts: Job packages waiting to be printed are displayed on top, from below – there are jobs of the package selected on top.

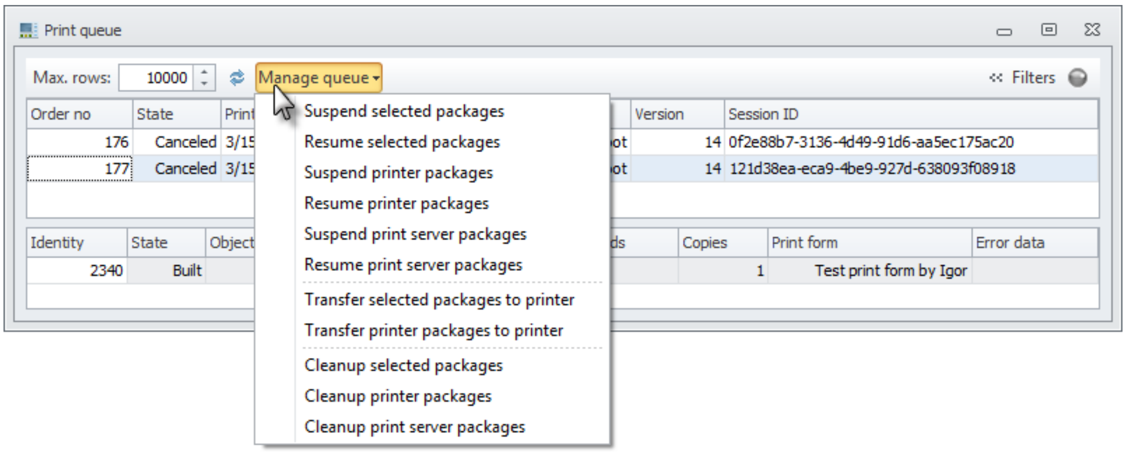
The following information is displayed in the job package list:

- *Order no* – a package number in print queue. Sequence of deck printing corresponds to the order of their numbers in the queue;
- *State* – package state:
  - *Active* – a package in print;
  - *Canceled* – a package is canceled;
- *Print date* – package print date;
- *Printer* – a printer wherein the package was sent;
- *User* – a user login who sent the package to print;
- *Version* – a version of metadata by means of which the client application Ultimate Solid of the user who sent the package works;
- *Session ID* – session in which the package was sent.

The following information is displayed for the jobs of the selected package:

- *Identity* – job ID;
- *State* – job state:
  - *NonActive* – the job is not active;
  - *Building* – printed form of the job is built;
  - *Built* – printed form of the job was built;
  - *Printing* – the job is printed;
  - *Printed* – the job was printed;
  - *Invalid* – a job misprint;
- *Object identity* – an object ID, record or records of which were sent to print;
- *Record identity* – record identity that was sent to print;
- *List of records* – list of the records that were sent to print;
- *Copies* – number of job copies that are sent to print;
- *Print form* – a print form used when printing job;
- *Error data* – information on errors.

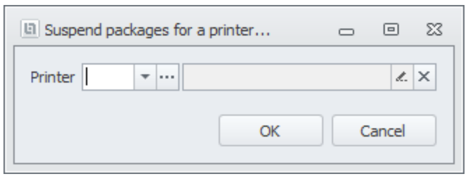
The form functional allows the following operations for job packages:


Print queue
Max. rows: 10000
Manage queue
Filters
Suspend selected packages
Resume selected packages
Suspend printer packages
Resume printer packages
Suspend print server packages
Resume print server packages
Transfer selected packages to printer
Transfer printer packages to printer
Cleanup selected packages
Cleanup printer packages
Cleanup print server packages

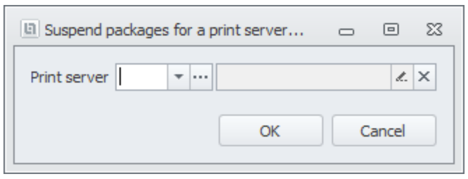
Order no	State	Print	Version	Session ID
176	Canceled	3/15	14	0f2e88b7-3136-4d49-91d6-aa5ec175ac20
177	Canceled	3/15	14	121d38ea-eca9-4be9-927d-638093f08918

Identity	State	Object	Copies	Print form	Error data
2340	Built		1	Test print form by Igor	

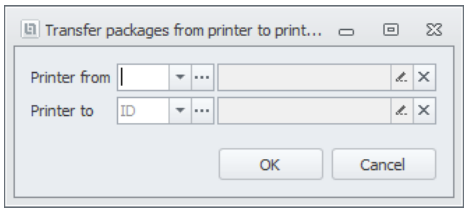
- *Suspend selected packages* – to suspend printing of the selected packages;
- *Resume selected packages* – to resume printing of the selected packages. When suspended print is resumed the packages will be moved to the end of the queue and their numbers *Order no* will also be changed;
- *Suspend printer packages* – to suspend printing of all packages that were sent to the selected printer. Operation is preceded by opening of a selection form of the printer;



- *Resume printer packages* – to resume printing of all packages that were sent to the selected printer. Operation is preceded by opening of a selection form of the printer. When suspended print is resumed the packages will be transferred to the end of the queue;
- *Suspend print server packages* – to suspend printing of all packages that were sent to the selected print server. Operation is preceded by opening of a selection form of the print server;



- *Resume print server packages* – to resume printing of all packages that were sent to the selected print server. Operation is preceded by opening of a selection form of the print server. When suspended print is resumed the packages will be transferred to the end of the queue;
- *Transfer selected packages to printer* – to transfer printing of the selected packages to the specific printer. Operation is preceded by opening of a selection form of the printer. When a printer is changed the packages will be transferred to the end of the queue;
- *Transfer printer packages to printer* – to transfer to the selected printer printing of all packages that were sent to the specific printer on the selected printer. Operation is preceded by opening of a selection form of printers. When a printer is changed the packages will be transferred to the end of the queue;

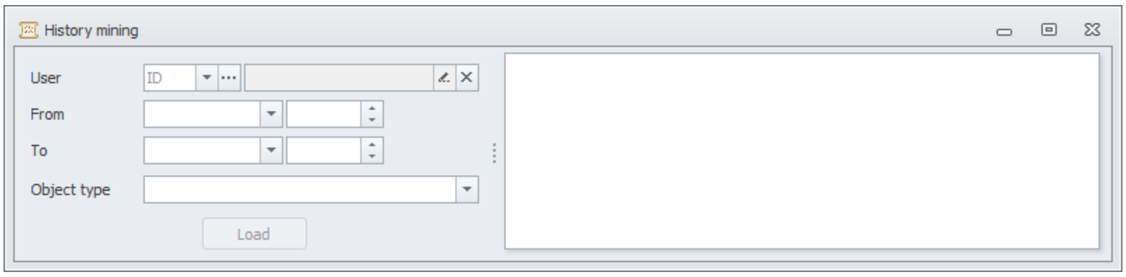


- *Cleanup selected packages* – to cleanup printing of the selected packages;
- *Cleanup printer packages* – to cleanup printing of all packages that were sent to the selected printer. Operation is preceded by opening of a selection form of the printer;
- *Cleanup print server packages* – to cleanup printing of all packages that were sent to the selected print server. Operation is preceded by opening of a selection form of the print server.

History tools

History search

The search in the history of object changes can be carried out using the History mining form:

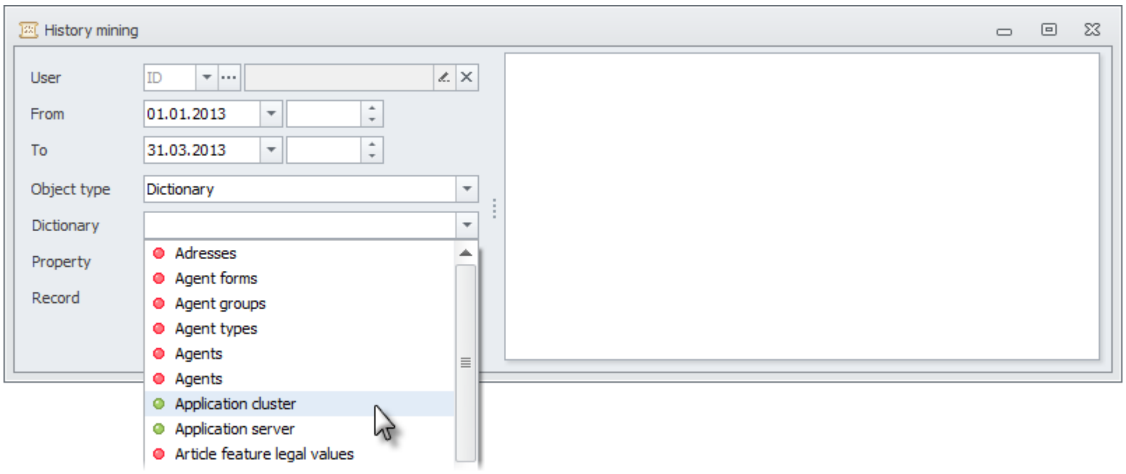


The search tool allows to localize the search area with maximum detail with the help of the following properties:

- *User* – the user who made the changes. This is the only option which is not required to be selected for search;
- *From* – initial date (and optionally time) of the period to be searched;
- *To* – end date (and optionally time) of the period to be searched:
- *Object type* – type of the object which change history is searched:
  - *Dictionary* – dictionary;
  - *Document* – document;
  - *Link table* – link table;
  - *Table part of document* – table part of the document.

The search can be performed only for non-system objects. That is, for example, the search in the history of changing the kernel (system) dictionaries containing metadata is impossible.

Depending on the selected type of object, the relevant additional information will be available:



From 01.01.2013

To 31.03.2013

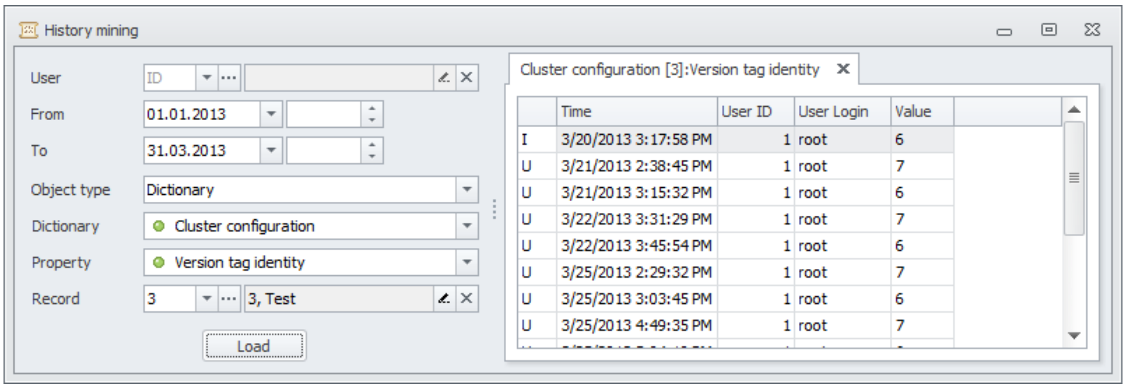
Object type Dictionary

When localizing the object, for example, selecting a certain dictionary and its properties which changes need to be searched, one can see additional icons before the name of the object. The green icon ● means that at the moment logging is on for such object. The red icon ● means that at the moment logging for the object is off (logging is described in detail in the relevant section).

Icons ● and ● show the current status of object logging at the time of the search.

It may well occur that logging for the object marked by the red icon ●, was toggled off only the day before, and search of older changes for it will still yield results.

The change history mining results are displayed in the right part of the form in the tab which title reflects the search criteria. The new search result will be displayed in a new tab:


History mining

User	ID	
From	01.01.2013	
To	31.03.2013	
Object type	Dictionary	
Dictionary	Cluster configuration	
Property	Version tag identity	
Record	3	3, Test

Load

Cluster configuration [3]:Version tag identity

	Time	User ID	User Login	Value
I	3/20/2013 3:17:58 PM	1	root	6
U	3/21/2013 2:38:45 PM	1	root	7
U	3/21/2013 3:15:32 PM	1	root	6
U	3/22/2013 3:31:29 PM	1	root	7
U	3/22/2013 3:45:54 PM	1	root	6
U	3/25/2013 2:29:32 PM	1	root	7
U	3/25/2013 3:03:45 PM	1	root	6
U	3/25/2013 4:49:35 PM	1	root	7

In the title of the tab reflecting search criteria, the information is displayed in the following format: *ObjectName [ObjectID]: ObjectPropertyName*, where:

- *ObjectName* – object of metadata to be searched;
- *ObjectID* – object record ID;
- *ObjectPropertyName* – subsidiary property of the object which changes are searched.

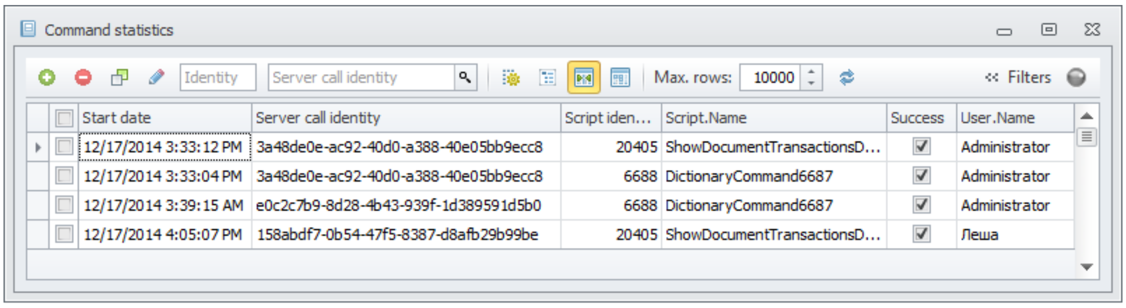
The following information is available for the found changes:

- *first untitled column* – type of operation which resulted in changing the values of the object:
  - *I* – insert, the value was inserted (for the first time);
  - *U* – update, the value was changed;
  - *D* – delete, the value was deleted;
- *Time* – time the change was made;
- *User ID* – ID of the user making the changes;
- *User* – login of the user making the changes;
- *Value* – value of the object after change.

History search is also integrated directly into the edit form of Dictionaries and Documents. If you select any field of dictionary or document, pressing hotkey Alt + H shows changes history of this field. Similarly, this shortcut works in table parts.

Command Calls

Search in the call history of interactive commands can be carried out in the dictionary "Command statistics":


Command statistics

Identity | Server call identity | Max. rows: 10000 | Filters

	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...	✓	Administrator
☐	12/17/2014 3:33:04 PM	3a48de0e-ac92-40d0-a388-40e05bb9ecc8	6688	DictionaryCommand6687	✓	Administrator
☐	12/17/2014 3:39:15 AM	e0c2c7b9-8d28-4b43-939f-1d389591d5b0	6688	DictionaryCommand6687	✓	Administrator
☐	12/17/2014 4:05:07 PM	158abdf7-0b54-47f5-8387-d8afb29b99be	20405	ShowDocumentTransactionsD...	✓	Леша

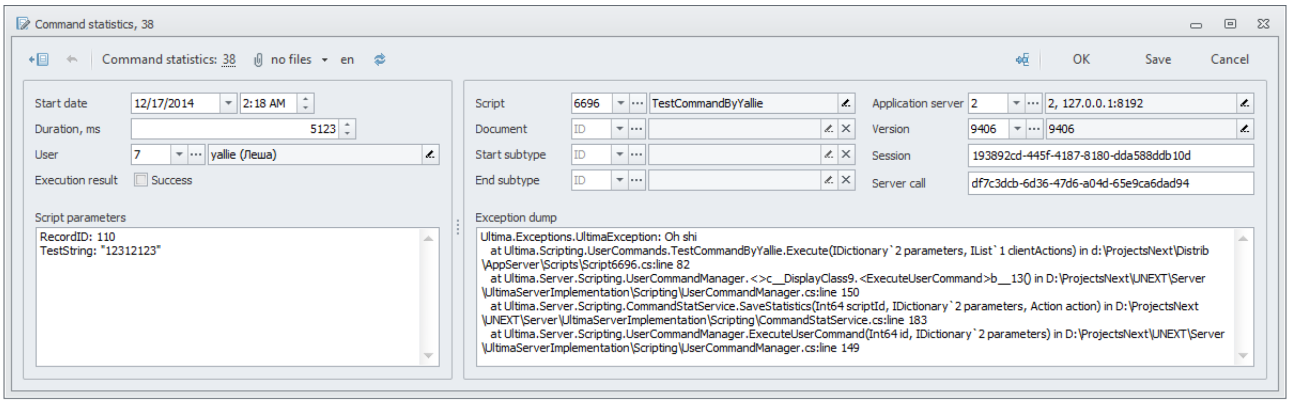
Calls are fixed by the system:

- user commands;
- commands on the document;
- commands on the list of documents;

- commands on the dictionary record;
- commands on the dictionary records.

Dictionary records can be filtered on *Server Call Identifier* (*Server call identity*).

For each called command the following information is saved in system:


Command statistics, 38

Command statistics: 38 | no files | en

Start date	12/17/2014 2:18 AM	Script	6696 TestCommandByYallie
Duration, ms	5123	Document	ID
User	7 yallie (Леша)	Start subtype	ID
Execution result	☐ Success	End subtype	ID
		Application server	2 — 2, 127.0.0.1:8192
		Version	9406 — 9406
		Session	193892cd-445f-4187-8180-dda588ddb10d
		Server call	df7c3dcb-6d36-47d6-a04d-65e9ca6dad94

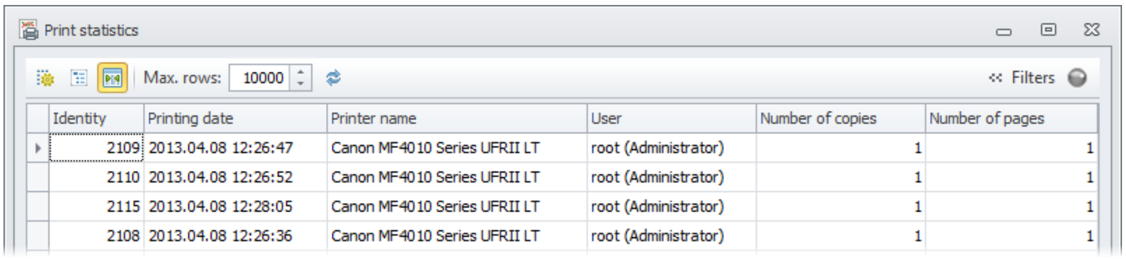
Script parameters
```
RecordID: 110
TestString: "12312123"
```

Exception dump
```
Ultima.Exceptions.UltimaException: Oh shi
   at Ultima.Scripting.UserCommands.TestCommandByYallie.Execute(IDictionary`2 parameters, IList`1 clientActions) in d:\ProjectsNext\Distrib
\AppServer\Scripts\Script6696.cs:line 82
   at Ultima.Server.Scripting.UserCommandManager.<>c__DisplayClass9.<ExecuteUserCommand>b__13() in D:\ProjectsNext\UNEXT\Server
\UltimaServerImplementation\Scripting\UserCommandManager.cs:line 150
   at Ultima.Server.Scripting.CommandStatService.SaveStatistics(Int64 scriptId, IDictionary`2 parameters, Action action) in D:\ProjectsNext
\UNEXT\Server\UltimaServerImplementation\Scripting\CommandStatService.cs:line 183
   at Ultima.Server.Scripting.UserCommandManager.ExecuteUserCommand(Int64 id, IDictionary`2 parameters) in D:\ProjectsNext\UNEXT\Server
\UltimaServerImplementation\Scripting\UserCommandManager.cs:line 149
```

- *Start date* – date and time of start of the command;
- *Duration, ms* – duration of command execution in milliseconds;
- *User* – user who started the command;
- *Execution result* – execution result of the command. Execution is successful if the flag *Success* is set;
- *Script parameters* – parameters and their values transferred at command execution to script *Script*, in the format [*Parameter*]: [*Value*];
- *Script* – script of the called command;
- *Document* – document on which the command was executed;
- *Start subtype* – start subtype of the document before command execution;
- *End subtype* – end subtype of the document on completion of the command;
- *Application server* – application server on which the command was executed;
- *Version* – version of metadata on which the command was executed;
- *Session* – session identifier in which the command was executed;
- *Server call* – server call in which the command was executed;
- *Exception dump* – exception, if it was the result of the command.

Print statistic

Print statistics can be viewed by using the form "Print statistics":


Print statistics

Max. rows: 10000

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

on each statistical record the following information for printing are available:

- *Printing date* – date and time of completion of the print and removal of a task from the print queue;
- *Printing user* – user that initiated the printing;
- *Printer* – printer on which the printing was carried out;
- *Printer Name* – system printer name on which the printing was carried out;
- *Print from* – print form which was used for printing;

- *Version* – version of metadata;
- *Number of pages* – number of pages of the printed task (one copy);
- *Number of copies* – number of copies of the printed task. The total number of the printed pages of the task will be equal to production of number of pages of the task (*Number of pages*) to number of its copies (*Number of copies*).

Total limits resets

To reset the limits of totals is possible by means of the command “Total limit resets”.

Open by ID tools

Group tools are used to perform the following operations on objects:

- *Document subtype history* – viewing of change of subtype history of the document chosen by the *identifier* (*Document ID*) of the document ;
- *Restore document* – restore (attempt) of the deleted document by its *identifier* (*Document ID*).

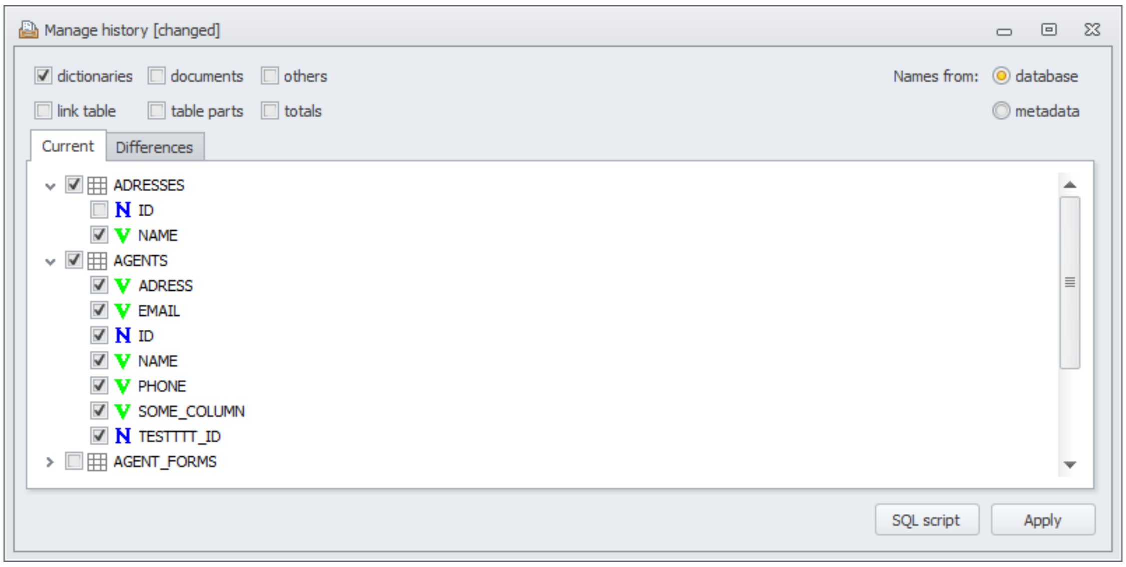
To perform the operation on the object it is necessary to add its identifier into the respective field and to press the key Enter.

Manage tools

Metadata objects logging setup

Ultimate Solid system has a logging function that provides a possibility to store history of changes made in non-system tables of the database. The change history search tool can be found in the History tools.

The logging settings are made using the Manage history form:



Manage history [changed]

[✓] dictionaries	[ ] documents	[ ] others	Names from: [✓ database]
[ ] link table	[ ] table parts	[ ] totals	metadata

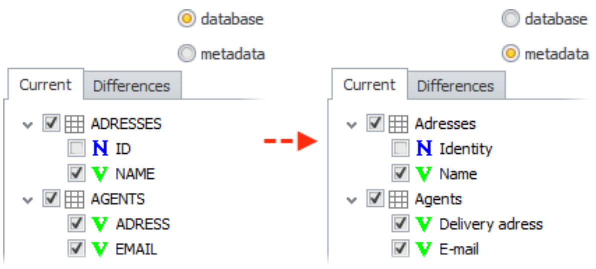
Current | Differences

- [✓] ADRESSES
  - [ ] N ID
  - [✓] V NAME
- [✓] AGENTS
  - [✓] V ADRESS
  - [✓] V EMAIL
  - [✓] N ID
  - [✓] V NAME
  - [✓] V PHONE
  - [✓] V SOME_COLUMN
  - [✓] N TESTTTT_ID
- [ ] AGENT_FORMS

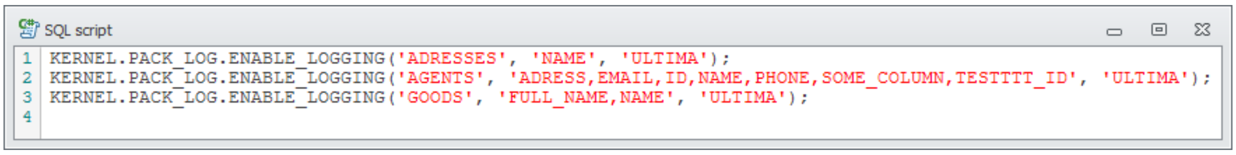
The essence of the settings comes down to selecting flags of the object properties that need to be logged:

- in the top left corner of the form, one can check the types of objects, and display them in the "Current" tab:
  - *dictionaries* – dictionaries;

- *link table* – link tables;
- *documents* – document types;
- *table parts* – table parts;
- *others* – other objects;
- *totals* – totals;



- in the top right corner of the form, one can select the format of displaying the names of the objects (*Names from*) in "Current" and "Differences" tabs:
  - *database* – names of the objects (tables and their columns) in the database;
  - *metadata* – names of metadata classes corresponding with the objects;
- *SQL script* – pressing the button opens a form containing an SQL script to apply the logging settings made. The script may be necessary in case the settings were not made in the time most optimal for their application and need to be applied later:


SQL script

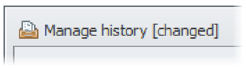
```
KERNEL.PACK_LOG.ENABLE_LOGGING('ADRESSES', 'NAME', 'ULTIMA');
KERNEL.PACK_LOG.ENABLE_LOGGING('AGENTS', 'ADRESS,EMAIL,ID,NAME,PHONE,SOME_COLUMN,TESTTTT_ID', 'ULTIMA');
KERNEL.PACK_LOG.ENABLE_LOGGING('GOODS', 'FULL_NAME,NAME', 'ULTIMA');
```

  The script is to be executed in the *Kernel* scheme in any application that works with the relational databases and supports SQL (PL SQL Developer, TOAD, etc.) ;
- *Apply* – clicking the button will applies the logging settings immediately.

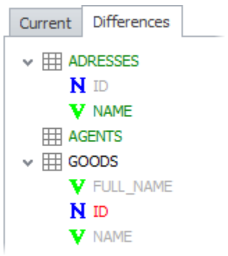
The "Current" tab in the tree structure (parent object and its properties - children) displays the current logging settings for the object properties, types of which are checked with flags in the top left corner of the form. All properties checked with flags will be logged. Here also one can toggle on logging for new properties by selecting them in flags, or deselect flags for the properties which logging is not required anymore. The names of the properties are preceded by icons specifying their type.:

- *N* – numerical properties (number) – types *long*, *decimal* or *bool*;
- *D* – dates (date) – types *date* or *DateTime*;
- *V* – text (varchar), not exceeding 4,000 characters – types *string*, *text* and *LargeText*;
- *C* – text (clob), not exceeding 4,000 characters – type *LargeText*;
- *B* – arrays/binary files (blob) – type*byte[]*.

If the logging settings contain changes which have not been yet applied, the heading, a relevant warning will be displayed in the title of the form: *changed*.



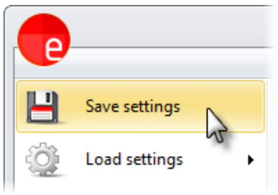
In the "Differences" tab, the tree structure displays the changes made in the logging settings and not yet saved:

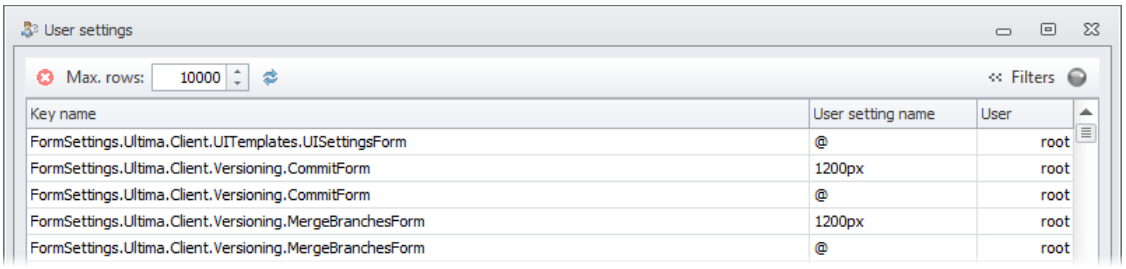


- the objects and their properties which logging was on, are marked in green;
- in such case, if the green marks the object which has no nested properties, that means that the logging was on for all properties of that object;
- if the logging was on not for all properties of the object, the non-logged properties will be marked by grey, and the object itself will be marked by green;
- the objects and their properties which logging was off, are marked red;
- in such case, if the red marks the object which has no nested properties, that means that the logging was off for all properties of that object;
- if the logging was off not for all properties of the object, the still logged properties will be marked by grey (and the object itself will be marked by black).

User settings

Users can save the settings of the active window/tab (window size, size and location of columns, their sorting, etc.). Settings can be saved in template by default or set any name to the template. The list of all user settings can be found in the form "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

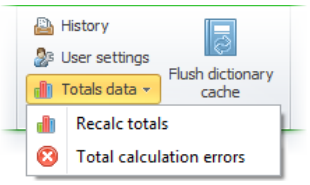
The following information is provided about each setting:

- *Key name* – the object which settings were saved. The name of object represents a record of the form *FormSettings.XXX*, where XXX – is a name of the form;
- *User setting name* – name of the template, in which settings were saved. The name of "@" corresponds to the template by default;
- *User* – user login, who saved the settings.

The chosen templates with settings can be removed with the button in a toolbar. Removal may be necessary if the template changes hinder the form opening, conflicting, for example, with changes made in the form by application developers.

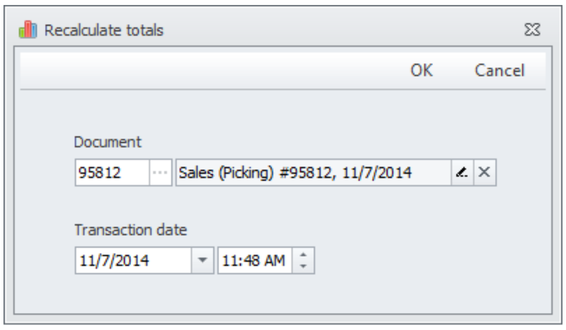
Recalculate totals

Recalculate Totals can be started forcibly with the command *Recalc totals*.



In the opened form "Recalculate totals" it is necessary to specify the date on which the results should be recalculated. It is possible to do it in two ways:

- to specify the date and time directly in the field *Transaction date*;
- to type the code of the document which change has caused the necessity to recalculate the totals into the field *Document*. The date of its holding will automatically be substituted into the field *Transaction date*.

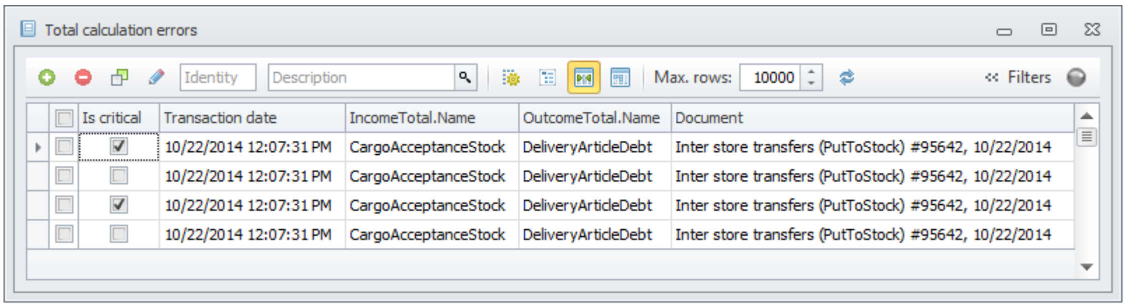


Document	95812 — Sales (Picking) #95812, 11/7/2014
Transaction date	11/7/2014 11:48 AM

The process of totals recalculation from the specified date will be started by clicking the "OK" key button.

Errors, arisen during totals recalculation, can be seen using the command *Total calculation errors*.

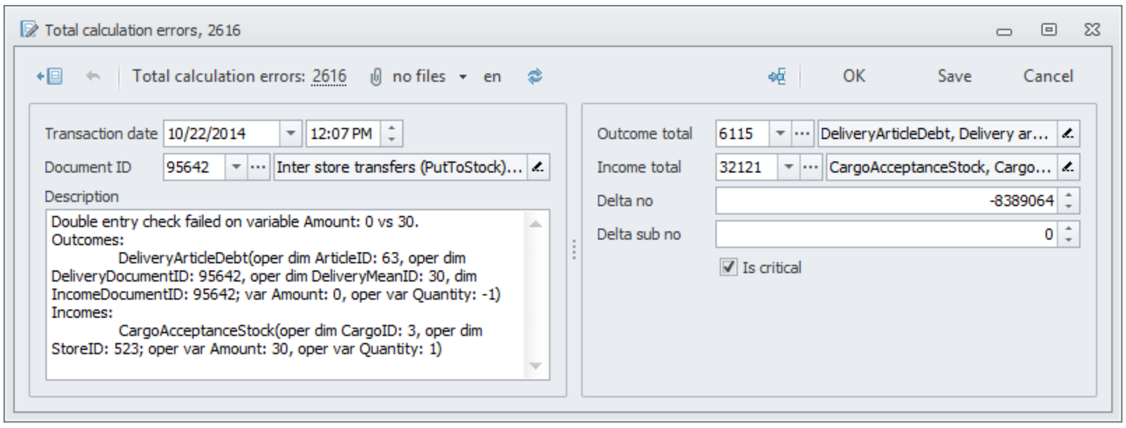
The list of all errors is provided in the opened the list form "Total calculation errors ":



	Is critical	Transaction date	IncomeTotal.Name	OutcomeTotal.Name	Document
	✓	10/22/2014 12:07:31 PM	CargoAcceptanceStock	DeliveryArticleDebt	Inter store transfers (PutToStock) #95642, 10/22/2014
		10/22/2014 12:07:31 PM	CargoAcceptanceStock	DeliveryArticleDebt	Inter store transfers (PutToStock) #95642, 10/22/2014
	✓	10/22/2014 12:07:31 PM	CargoAcceptanceStock	DeliveryArticleDebt	Inter store transfers (PutToStock) #95642, 10/22/2014
		10/22/2014 12:07:31 PM	CargoAcceptanceStock	DeliveryArticleDebt	Inter store transfers (PutToStock) #95642, 10/22/2014

Dictionary records can be filtered according to the *Description* of the error (*Description*).

The following is stored in the system for each error:


Total calculation errors, 2616
Total calculation errors: 2616
no files
en
OK
Save
Cancel

Transaction date	10/22/2014 12:07 PM
Document ID	95642 Inter store transfers (PutToStock)...
Outcome total	6115 DeliveryArticleDebt, Delivery ar...
Income total	32121 CargoAcceptanceStock, Cargo...
Delta no	-8389064
Delta sub no	0
Is critical	✓

Description

```
Double entry check failed on variable Amount: 0 vs 30.
Outcomes:
        DeliveryArticleDebt(oper dim ArticleID: 63, oper dim
DeliveryDocumentID: 95642, oper dim DeliveryMeanID: 30, dim
IncomeDocumentID: 95642; var Amount: 0, oper var Quantity: -1)
Incomes:
        CargoAcceptanceStock(oper dim CargoID: 3, oper dim
StoreID: 523; oper var Amount: 30, oper var Quantity: 1)
```

- *Transaction date* – transaction date;
- *Document ID* – a document, according to which the transaction was carried out;
- *Description* – error description;
- *Outcome total* – total, from which the delta is carried out;
- *Income total* – total, to which the delta is carried out;
- *Delta no* – delta number;
- *Delta sub no* – number of the calculated delta after subdivision;
- *Is critical* – the set flag indicates that the error is critical, for example, rule violation of double record.

Flush dictionary cache

Function of Flush Dictionary Cache *Flush dictionary cache* sends a command to the application server, the need to dump the cache of all the dictionaries on users' local computers. The server, in turn, sends the command to all currently connected clients. The clients, who have received the command, dump the data cache of the dictionaries.

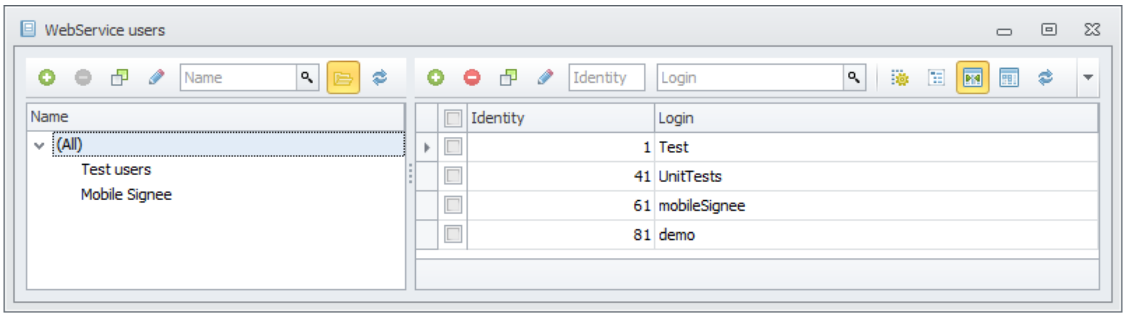
WebService tools

Access to web to services is limited and is provided only to the authorized users of web of services.

The issuance of the rights to perform web services is implemented by roles similar to the ordinary system users. Multiple roles of web services may be assigned to the user of web service. The role, in turn, may provide access to several web services. As a result, the user gets the access to all the web services of all chosen roles.

Web service users

The list of all web service users can be found in the dictionary "WebService users":



WebService users

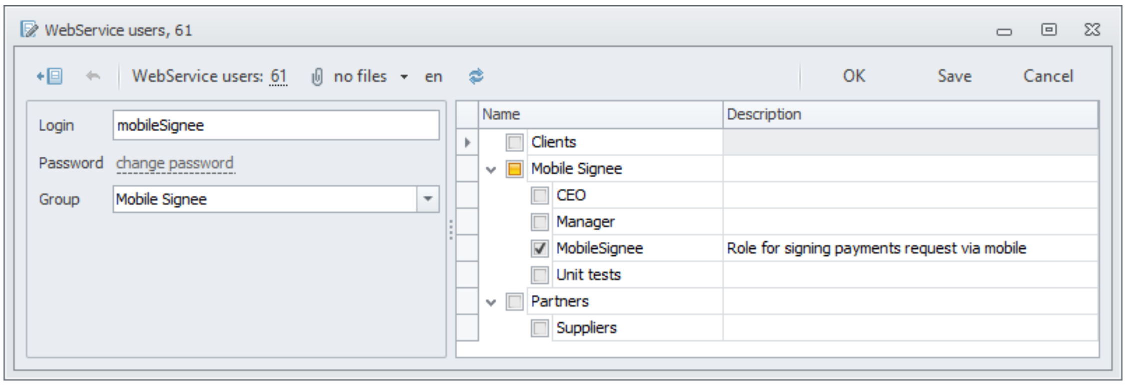
Name
(All)
Test users
Mobile Signee

Identity	Login
1	Test
41	UnitTests
61	mobileSignee
81	demo

Dictionary window is divided into two parts: on the left, there is a tree of user groups, on the right – the list of users of the group selected on the left.

User groups can be filtered by *Name* of groups (*Name*), and users by *Login* (*Login*).

The web-service user editing form is divided into two parts:



WebService users, 61

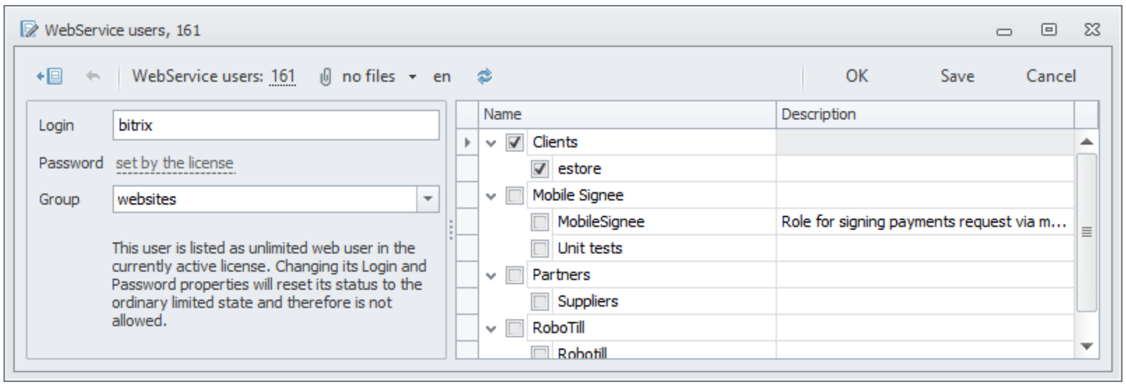
Login	mobileSignee
Password	change password
Group	Mobile Signee

Name	Description
☐ Clients	
▨ Mobile Signee	
☐ CEO	
☐ Manager	
✓ MobileSignee	Role for signing payments request via mobile
☐ Unit tests	
☐ Partners	
☐ Suppliers	

On the left, user properties are listed:

- *Login* – login;
- *Password* – reference *change password*, by clicking which the password change form opens;
- *Group* – the group, to which the role belongs to.

For unlimited users of web services the password is specified in the license file, therefore change of the password for them is forbidden:



WebService users, 161

Login	bitrix
Password	set by the license
Group	websites

This user is listed as unlimited web user in the currently active license. Changing its Login and Password properties will reset its status to the ordinary limited state and therefore is not allowed.

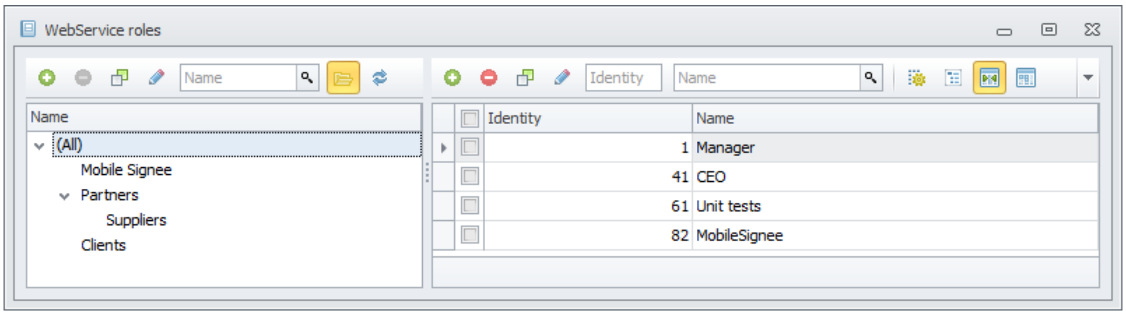
Name	Description
✓ Clients	
✓ estore	
☐ Mobile Signee	
☐ MobileSignee	Role for signing payments request via m...
☐ Unit tests	
☐ Partners	
☐ Suppliers	
☐ RoboTill	
☐ Robotill	

On the right, there is a tree of all roles of web services where roles assigned to such user are marked by flags. After assignment to the user, the role gives him access to all its web services. Several roles may be assigned to one user.

The list of the assigned roles can be edited by selecting and deselecting flags. Selecting the group in the tree results in selecting all roles included into it.

Web Service roles

The list of all roles of web services can be also found in the dictionary "WebService roles":


WebService roles

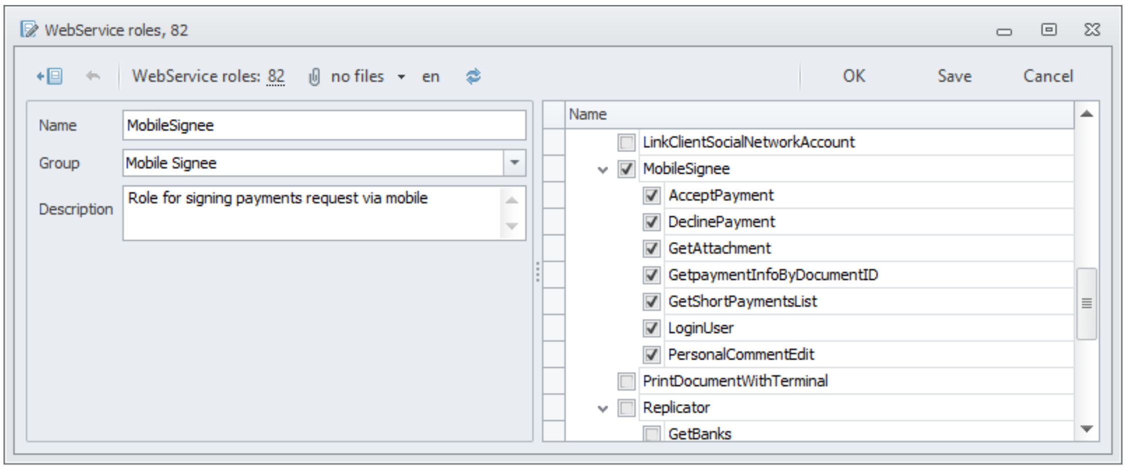
Name
(All)
Mobile Signee
Partners
Suppliers
Clients

Identity	Name
1	Manager
41	CEO
61	Unit tests
82	MobileSignee

Dictionary window is divided into two parts: on the left the tree of roles groups is shown, on the right – the roles list of the group chosen on the left.

Groups of roles and roles themselves can be filtered according to the *Name* (*Name*).

Form of roles editing of web services is divided into two parts:


WebService roles, 82

WebService roles: 82 | no files | en | OK | Save | Cancel

Name	MobileSignee
Group	Mobile Signee
Description	Role for signing payments request via mobile

Name
☐ LinkClientSocialNetworkAccount
☑ MobileSignee
☑ AcceptPayment
☑ DeclinePayment
☑ GetAttachment
☑ GetpaymentInfoByDocumentID
☑ GetShortPaymentsList
☑ LoginUser
☑ PersonalCommentEdit
☐ PrintDocumentWithTerminal
☐ Replicator
☐ GetBanks

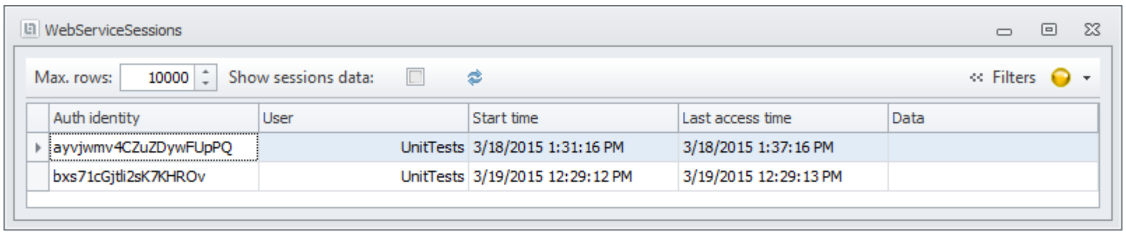
On the left the properties of the user are listed:

- *Name* – role name;
- *Group* – group that includes the role;
- *Description* – role description.

On the right the tree of all web services is displayed, where the flags marked that ones that are allowed access to the owner of the role. The list of available web services can be edited by installation or removal of the flags. Choice of group in a tree leads to the choice of all web services included in it.

Web service sessions

The list of all sessions of web services can be found in the form "WebServiceSessions":



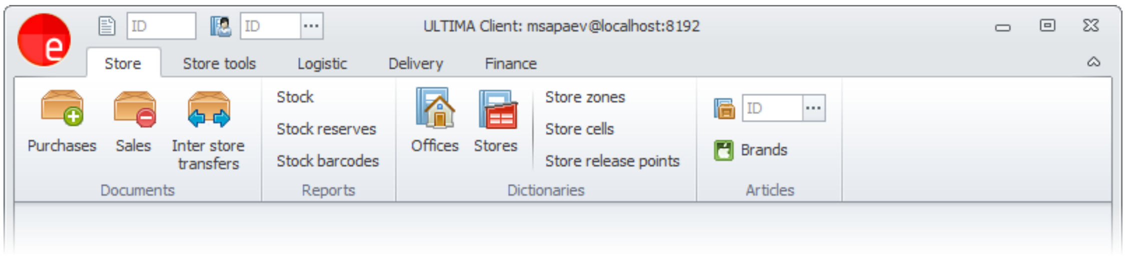
Auth identity	User	Start time	Last access time	Data
ayvjwmv4CZuZDywFUpPQ	UnitTests	3/18/2015 1:31:16 PM	3/18/2015 1:37:16 PM	
bxs71cGjtli2sK7KHROv	UnitTests	3/19/2015 12:29:12 PM	3/19/2015 12:29:13 PM	

- *Auth identity* – session identifier;
- *User* – login of service web user;
- *Start time* – start time of session;
- *Last access time* – last access time;
- *Data* – session data are displayed in the form with the set flag in the toolbar *Show sessions data*.

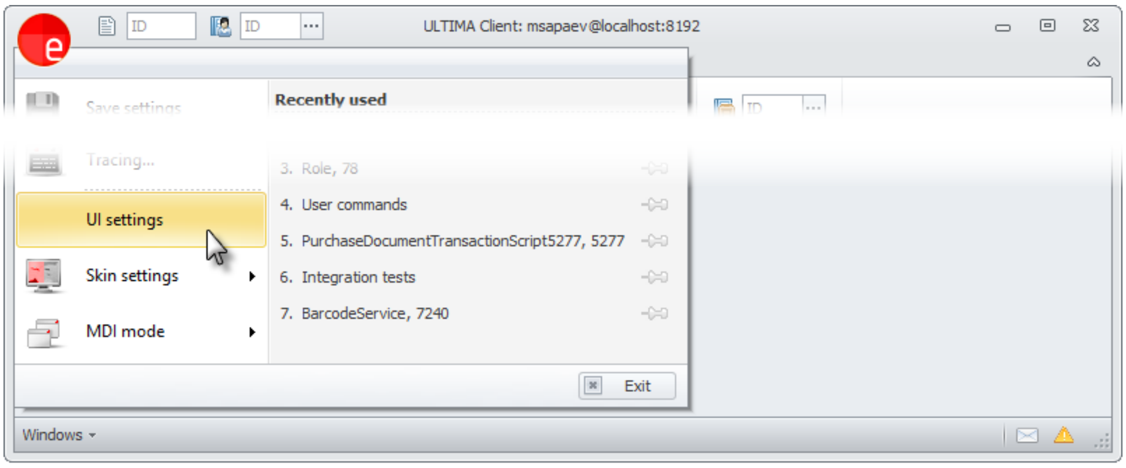
Lifetime of sessions is set in cluster settings.

User interface settings

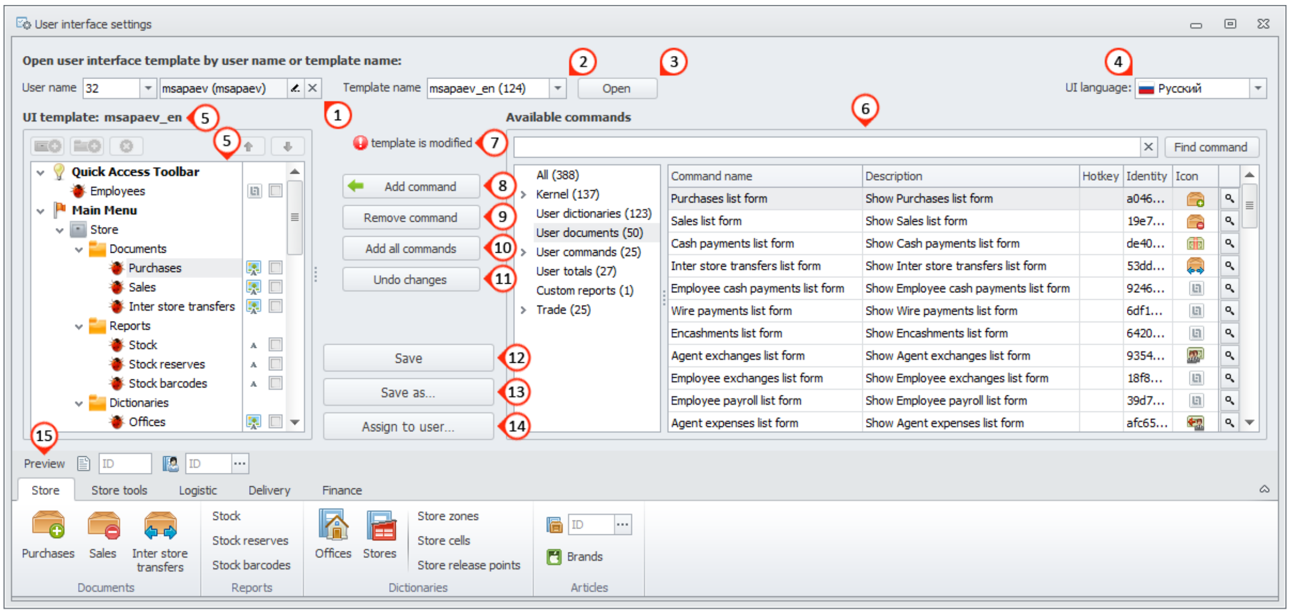
The appearance of the main menu of the main form of the client application can be customized:



The customization tool is called through the menu item “Customize user interface” Ctrl + Alt + C ::



Then the user interface customization form is opened, the template of the interface of the current user is loaded:



User name: 32 — msapaev (msapaev)
Template name: msapaev_en (124)
UI language: Русский

Command name	Description	Hotkey	Identity	Icon
Purchases list form	Show Purchases list form		a046...	
Sales list form	Show Sales list form		19e7...	
Cash payments list form	Show Cash payments list form		de40...	
Inter store transfers list form	Show Inter store transfers list form		53dd...	
Employee cash payments list form	Show Employee cash payments list form		9246...	
Wire payments list form	Show Wire payments list form		6df1...	
Encashments list form	Show Encashments list form		6420...	
Agent exchanges list form	Show Agent exchanges list form		9354...	
Employee exchanges list form	Show Employee exchanges list form		18f8...	
Employee payroll list form	Show Employee payroll list form		39d7...	
Agent expenses list form	Show Agent expenses list form		afc65...	

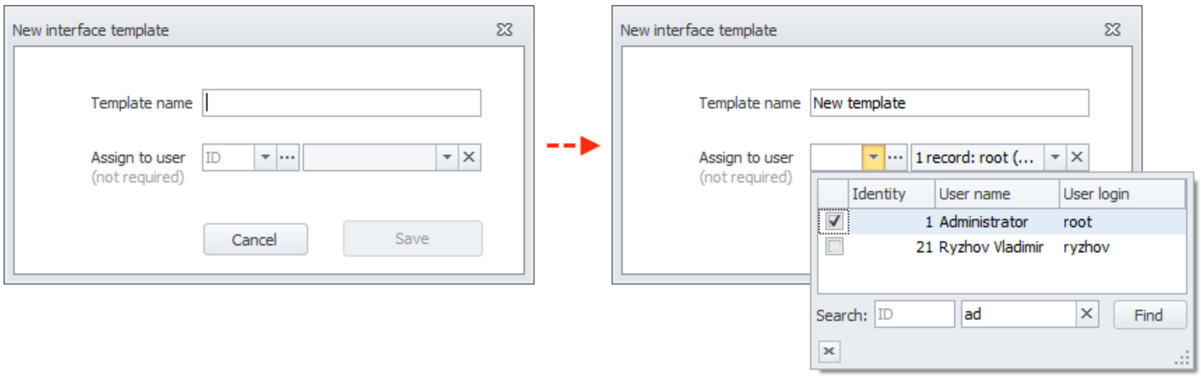
The following options are available for customization of the user interface:

- (1) a drop-down list used for selecting the user whose interface needs to be customized. By default the user under which the system was logged in is selected. In the list (2) the interface template tied to the selected user is selected automatically;
- (2) a drop-down list used for selecting the existing template which needs to be edited. The name of the interface template is unique and can be repeated.

In such case, they can be differed by identification number. (3) "Open" button loads the selected interface template;

- (4) a drop-down list used to select the language which is used in the interface template.
- Language selection in no way affects the interface itself;
- (5) the structure of the quick access panel and main menu of the loaded interface template in the form of tree of tabs, groups and commands. The name of the loaded template is duplicated in the end of the heading;
- (6) the set of commands available for adding to the interface template;
- (7) when any changes are made into the loaded template, the indicator “Template changed” appears;
- (8) the "Add command” button adds the command selected in the list to the loaded interface template;
- (9) the “Delete command” button deletes the selected command, group or tabs with all contents;
- (10) the "Add all commands button” button adds all commands selected in the list of available (according to the filter) to the loaded template of the interface;
- (11) the "Discard changes" button returns the loaded template to the original state at the time of last saving or opening if there was not saving;
- (12) the "Save" button saves all changes made in the template;

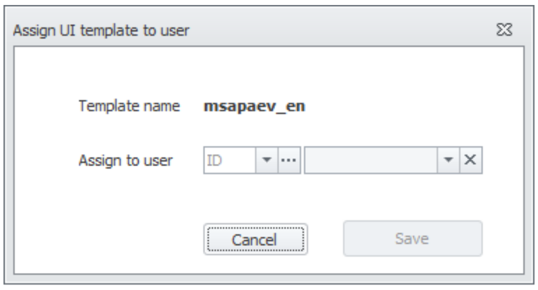
- (13) the "Save as..." saves the interface with all changes made to the new template. When clicked, the window of a new interface window opens. The name of the new template is entered into the field “Template name”. In the drop-down list “Assign to user” one can optionally select one or more users to which this new template will be assigned after saving the interface:



	Identity	User name	User login
✓	1	Administrator	root
	21	Ryzhov Vladimir	ryzhov

When saving, the new template will be opened in section (5). No changes will be made to the interface edited before (if it was not saved).

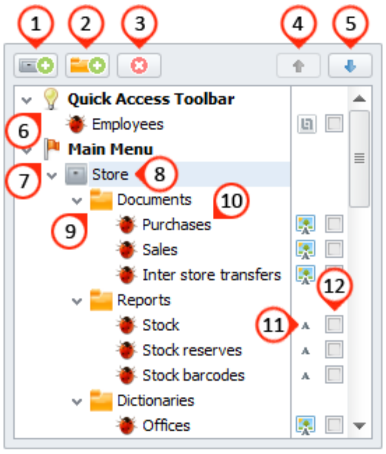
- (14) the button "Assign to user..." allows to assign the loaded interface template to one or more users. When clicked, the window of assigning a template to the user opens. In the drop-down list “Assign to user” one can one or more users to which this new template will be assigned:



Template name msapaev_en
Assign to user

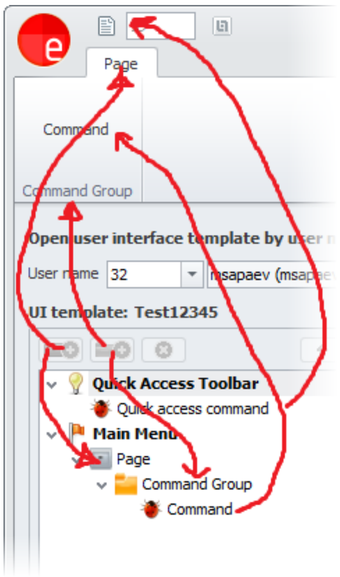
- (15) the preview of the created main menu where all changes are displayed in real time. One can toggle in it between tabs, minimize and expand the menu, but executing commands is unavailable. When selecting a tab or the relevant group or command in the "Interface template" (5), it will be opened in preview area (15).

The “Interface template...” window contains the set of commands of the “Quick access panel”(6) and the “Main menu”6) and the "Main menu" structure (7) in tree view:

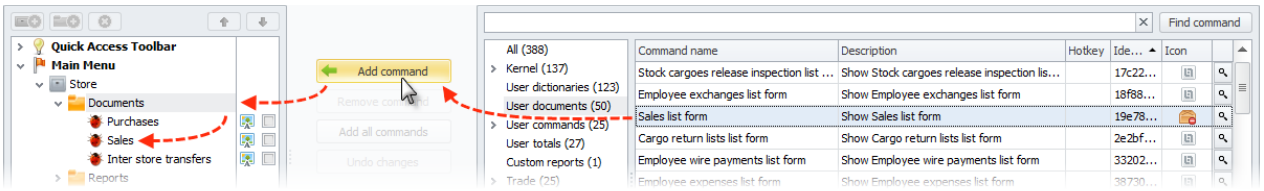


The set of commands is located in the upper part of the section:

- button (1) adds new tab to (8) main menu (7),
  button (1) adds new tab (8) to the main menu (7),



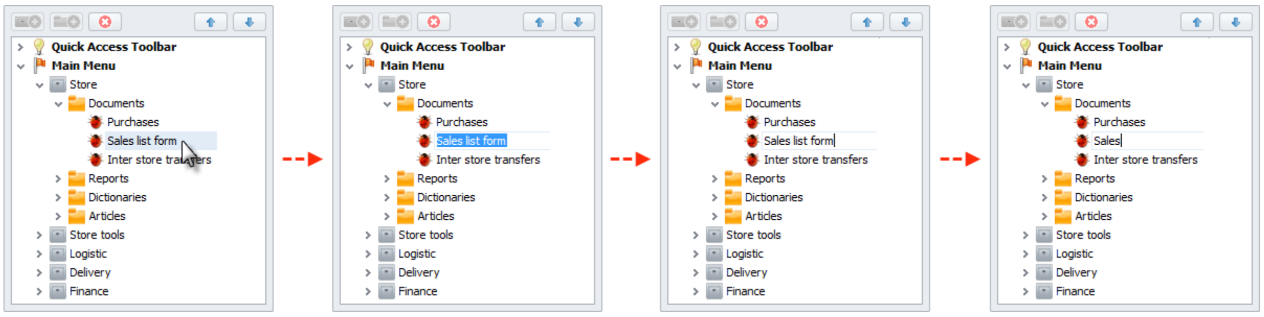
- ιν χασε α χομμανδ (**10**), group (**9**) or "Quick assess panel" (**6**) **is selected in the "Interface commands set", the button "Add command" to the right of the of the commands set becomes active.** When clicked, the command selected in the window of available for adding to the interface gets to the bottom of the list of commands of the group (in case the a group or a command included into it were selected in the "Interface command set") or to the bottom of the list of commands of the "Quick access panel" (if it or the command included into it are selected in the "Interface commands set"):



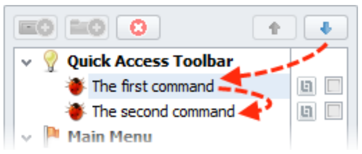
Command name	Description	Hotkey	Ide...	Icon
Stock cargoes release inspection list ...	Show Stock cargoes release inspection lis...		17c22...	
Employee exchanges list form	Show Employee exchanges list form		18f88...	
Sales list form	Show Sales list form		19e78...	
Cargo return lists list form	Show Cargo return lists list form		2e2bf...	
Employee wire payments list form	Show Employee wire payments list form		33202...	
Employee expenses list form	Show Employee expenses list form		38730...	

Also, the command can be added to the interface template by double click on the left mouse button.

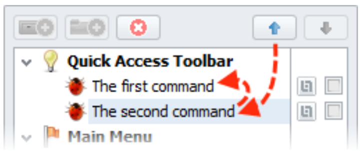
- τηε χομμανδ (**10**), group (**9**) or tab (**8**) selected in the "Interface commands set" can be renamed. For that, left click on it once and enter a new name. If the necessary row is already selected, you can start entering the new name right there and it will replace the old one. To save the name, press Enter or select another menu item. To discard the rename (before saving changes) it is sufficient to press Esc on the keyboard:



- button (**3**) deletes the command (**10**), the group (**9**) or the tab (**8**) **selected in the "Interface commands set" with all their contents;**
- βυττον (**4**) moves the command (**10**), the group (**9**) **or the tab (8) selected in the "Interface commands set" with all their contents one level down;**

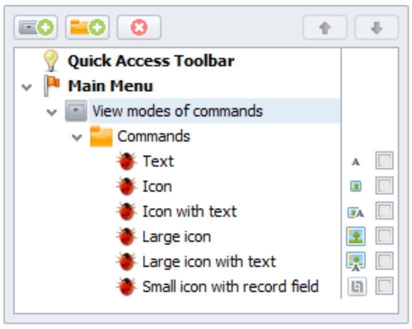


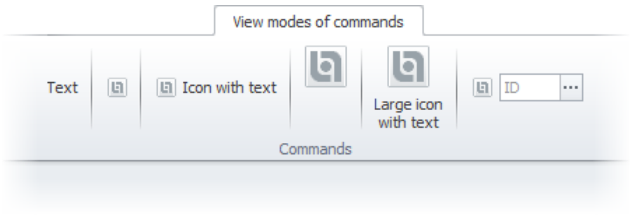
- βυττον (5) moves the command (10), the group (9) **or the tab (8) selected in the "Interface commands set" with all their contents one level up;**



- icon (11) located opposite to each command (10) **shows its view mode in the main menu.** You can change the view mode by double clicking the left mouse button on icon (11). There are following view modes:
  – text;
  – icon;
  – icon with text (on the right);
  – large icon;
  – large icon with text (below);

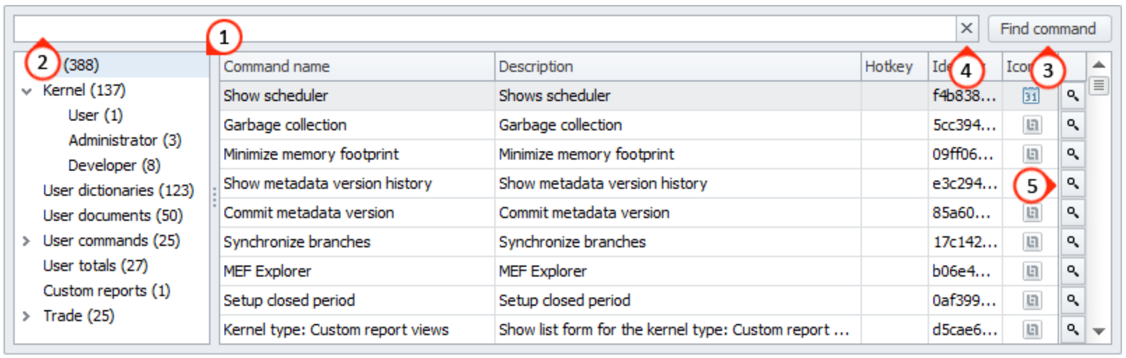
  icon - a small icon with record field allowing to open the list form of the objects and directly the object (dictionary record or the document by its ID):





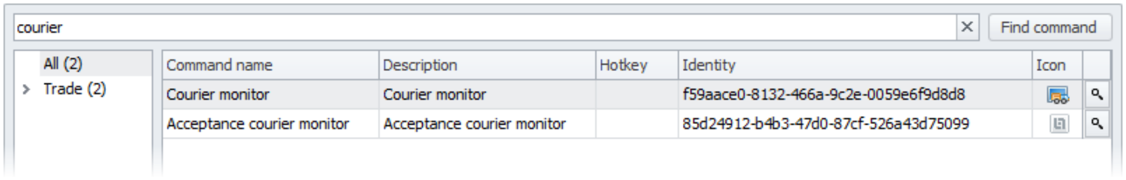
- flag (12) opposite to each command(10) allows to separate it by a vertical bar from the previous. Selecting the flag for the first command in the group (9) will yield no result.

The window "Commands for adding to the interface template" contains the list of commands which can be added to the user interface:



Command name	Description	Hotkey	Identity	Icon
Show scheduler	Shows scheduler		f4b838...	
Garbage collection	Garbage collection		5cc394...	
Minimize memory footprint	Minimize memory footprint		09ff06...	
Show metadata version history	Show metadata version history		e3c294...	
Commit metadata version	Commit metadata version		85a60...	
Synchronize branches	Synchronize branches		17c142...	
MEF Explorer	MEF Explorer		b06e4...	
Setup closed period	Setup closed period		0af399...	
Kernel type: Custom report views	Show list form for the kernel type: Custom report ...		d5cae6...	

- the commands are organized by groups (the number commands in the groups is shown in brackets) in tree mode (1). The first paragraph "All" shows all available commands, the selection of other groups limits the displayed list;
- in line (2), commands can be filtered. The search will be carried out only within the contents of the group selected in structure (1) (when selecting the item "All found" - within all commands). The filtering result will be a list of command satisfying the search condition. To carry out the search, it is necessary to enter text in the filtering line and click Enter or "Find command" button (3). For example, by entering "dictionary" in the search line, we limit the list of commands to eight:



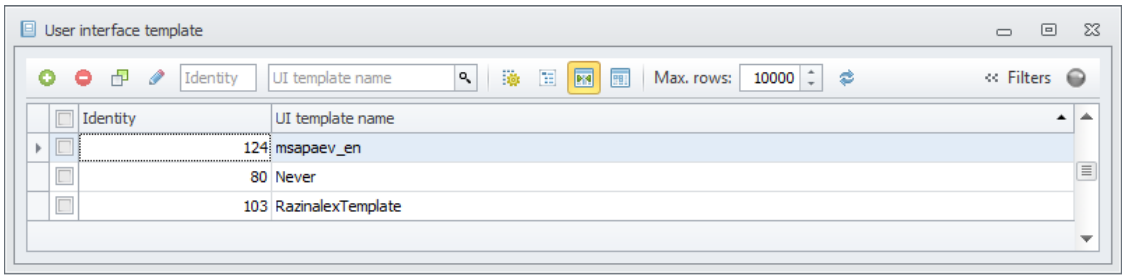
Command name	Description	Hotkey	Identity	Icon
Courier monitor	Courier monitor		f59aace0-8132-466a-9c2e-0059e6f9d8d8	
Acceptance courier monitor	Acceptance courier monitor		85d24912-b4b3-47d0-87cf-526a43d75099	

- the filter can be reset by button (4);

- using the search button (5), you can check if the command has already been added to the interface template. Each occurrence will be successively highlighted at the command corresponding to it in section "Interface template" (the next occurrence with every next click on button) (5) if it has been already added to the interface before;
- when selecting the command in the 'Interface template" (on the left), it and its group will be highlighted in the list "Commands to be added to the interface template".

UI templates

All user interface templates are stored in the dictionary User interface template:

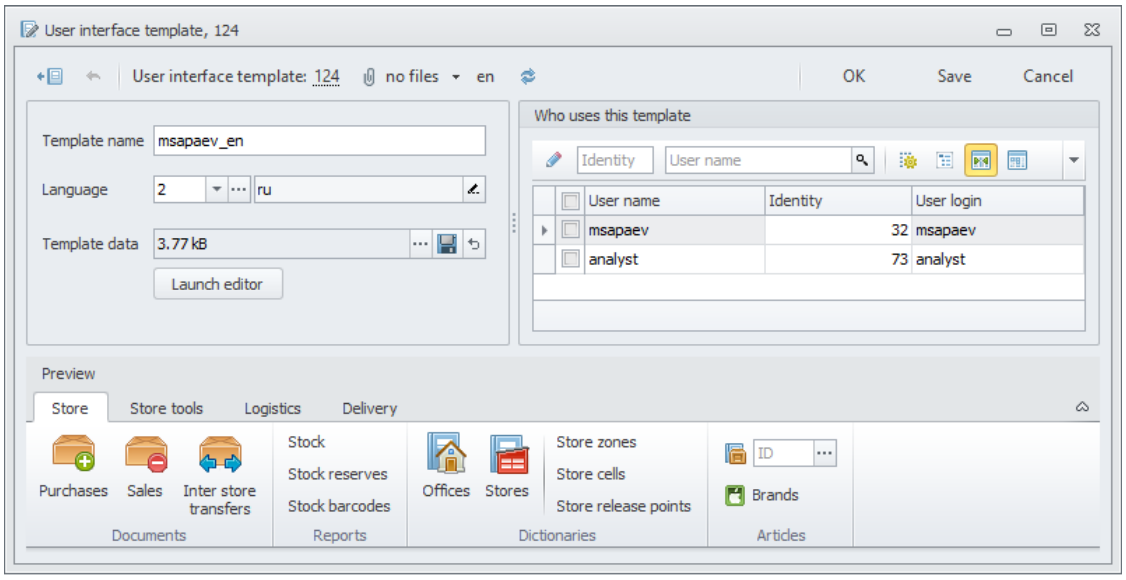

User interface template

Identity | UI template name | Max. rows: 10000 | Filters

Identity	UI template name
124	msapaev_en
80	Never
103	RazinalexTemplate

Templates can be filtered according to the *Name* (*UI template name*).

Template edit form allows to set the following properties:


User interface template, 124

User interface template: 124 | no files | en | OK | Save | Cancel

Template name	msapaev_en
Language	2 ru
Template data	3.77 kB

Launch editor

Who uses this template

User name	Identity	User login
msapaev	32	msapaev
analyst	73	analyst

- *Template name* – template name;
- *Language* – language which is used in the template. The choice of language has information character and does not affect the interface itself;
- *Template data* – directly an interface template.. To open it in visual editor is possible by clicking the button "Launch editor";
- *Who uses this template* – a list of users who are assigned this interface. The list can be filtered by *User name* (*User name*);
- *Preview* – area of a preview of the template.