prudsys RDE

The Realtime Decisioning Engine

prudsys RDE Manual

prudsys RDE

The Realtime Decisioning Engine

RDE Server Release 2.11.3

This manual relates to version 2.11.3 of prudsys RDE Server with following modules:

prudsys RDE | Recommendations

prudsys RDE | Newsletter

prudsys RDE | Pricing

prudsys RDE | Assortment Planning

prudsys RDE | Data Cleansing

prudsys RDE | Scoring

prudsys AG
The Realtime Analytics Company
Zwickauer Straße 16
09112 Chemnitz

E-Mail: info@prudsys.de

WWW: www.prudsys.de
www.prudsys.com
www.prudsys.it

Chemnitz, April 2011

Table of Contents

List of figures

List of tables

1 Introduction to prudsys RDE

Competition in retail is becoming more and fiercer. Data mining permits to shape and forecast customer behaviour, thus turning it into an important tool for withstanding the competitive pressures. However, current data mining tools mostly require manual operation by expert users. What, indeed, is needed are closed-loop solutions which cover, fully automatically, the entire cycle of customer interactions across all customer touch points, generating a consistent experience throughout the customers entire lifecycle.

Recommendation engines are a part of this system, providing personalized, highly relevant purchase recommendations to the customer in real time. The prudsys recommendation solution, integrated into the prudsys RDE (real-time decisioning engine) is an automated, self-learning recommendation engine for multi-channel applications.

prudsys RDE in action

The prudsys RDE modules primary aims at online and stationary retailers. First of all, it is designed to increase the turnover of online shops and portals (both, B2B and B2C). At the same time, it can also be used for other channels, such as content-management systems and in-store devices (terminals, checkouts, scales, PSAs/PDAs, etc.). New application areas include self-service systems and call centres (e. g. address selection, provision of optimized discussion guidelines).

The prudsys RDE learns autonomously based on constant interaction with users, making use of the interrelation between experience and research (i.e. "exploit" and "explore"). On the one hand, the prudsys RDE makes use of methods for analyzing shopping transactions and customer data to extract strong links between bought products for customers. On the other hand, the system tests new recommendations to extend its scope of service. The results are continuously evaluated and are used to further improve the recommendations. Thanks to its real time approach and its constant user interaction, changing customer behaviour is instantly reflected in the recommendations displayed by the system.

Data sources

One source of analysis are either the shop database or the log files of the Web server, but also data from other channels (Data Warehouse, ERP, CRM) can be included into the analysis. However, the most important source of analysis is the behaviour of the current shop visitor and her reaction towards the recommendations, either products or content, to deduce in real-time the most relevant offers for her current interest.

In addition, shop administrators can pre-define manual rules on product and category level which are then also used for the automatic evaluation. On top of that, it is also possible to pre-set filters for negative lists, availability, price optimization, etc. These features help shop administrators to control recommendations directly and to customize them according to their requirements.

REST interface

The functionalities of the prudsys RDE are highly automatable and can easily be integrated into existing IT structures, due to its open, non-proprietary REST (REpresentational State Transfer) interface. The communication between clients and the RDE Server is exclusively based on HTTP requests.

For the comfortable and easy configuration and evaluation of the recommender system, a graphical administration interface is delivered together with the web application.

2 Basic Installation of the RDE Server

2.1 Integration of the RDE Server into the IT infrastructure

HTTP protocol

The communication with the RDE Server is carried out exclusively on the basis of the HTTP protocol. In order to query the properties of an object or to create, change or delete an object, a simple HTTP request has to be executed. The complete administration of a recommendation engine within the RDE Server as well as the query of product recommendations can be executed via HTTP requests. Thus, every software application that handles HTTP requests can communicate with the RDE Server.

Depending on the integration format, the software must be able to send GET, POST, PUT and DELETE requests as well as display structured input data in JSON format or interpret the return data in JSON format, respectively.

Usually, the installation and implementation of the RDE Server is executed by prudsys employees. As this process is very simple, it can also be executed by the shop operator if desired.

2.2 System requirements

From a technical point of view, it is possible to install the prudsys recommendation service on the same resources as the online shop. Nevertheless, it is advisable to install and implement the prudsys RDE on an own server.

Minimum server standards

The server designated for the RDE Server should fulfil the following minimum requirements:

- Linux / Unix operation system, at least 2 GB RAM
- 40GB hard drive for approximately 60.000 products
- Sun Java 6
- jsvc binary from Apache Tomcat Project

There are two possibilities for the installation of the required server. prudsys provides a package which contains all necessary components, but it is also possible to install your own Apache Tomcat environment. We recommend using the prudsys package as all components have been tested and approved for the prudsys RDE.

2.3 prudsys RDE-Tomcat-Setup

2.3.1 Advantages

The prudsys installation package provides the following advantages compared to an independent installation of single components:

- Standardized RDE installation leads to more efficient support by the prudsys consulting team
- No dependence on the Tomcat packages provided by Linux distributors
- Easy installation of several RDE Tomcat instances on one system
- Reduction of error sources during the RDE installation by hosting partners

2.3.2 Installation

Step 1: Unpacking the directory

The directory "rde_server-Version.tgz" can be unpacked anywhere and can be executed right away. It creates a directory called "rde/" which includes all corresponding directories and files including the Tomcat.

If you want to run several RDE Tomcat installations simultaneously on different ports, unpack the package several times. New directories will be created in "rde/", for example "rde/master/" and "rde/slave/".

Note: The directory name can be changed, but the structure must not be changed.

Step 2: Configuration

There are two different configuration approaches, one for Linux and one for Windows.

Linux Installation

The configuration is carried out solely via the file "rde/etc/rde.conf". All variables which are processed by the start scripts are defined within this file.

Parameter	Description
Java specific parameters	
JAVA_HOME	Java environment variable
RDE_DAEMON	Jsvc (java service) environment variable
RDE_TOMCAT_USER	Jsvc starts tomcat as this user
RDE_TMP_DIR	Default temp directory
RDE_PID_PATH	Path for PID files
Global RDE parameters	
RDE_NAME	Used for PID files
RDE_PATH	RDE path

Parameter	Description
RDE_TOMCAT_SHUTDOWN_PORT	Shutdown port
RDE_TOMCAT_CATALINA_PORT	Catalina port
RDE_TOMCAT_AJP_PORT	AJP port
RDE_H2DB_PORT	Database port
RDE_TOMCAT_MEMORY_MAX	RDE memory
RDE_TOMCAT_MEMORY_START	RDE memory
RDE_DATA_DIR	RDE data directory
RDE_GCLOG_DIR	RDE log directory
RDE_TOMCAT_GC	Additional garbage collector options
RDE specific parameters (for more than one RDE application)	
RDE_CONFIG_COUNT	Number of installed RDE applications
RDE_<ID>_NAME	RDE specific parameters will be identified with <ID>

Table 1: RDE Tomcat configuration parameters

Windows Installation

The installation for Windows is carried out via the file “rde/apache-tomcat-6.0.26/bin/setenv.bat“ with the same basic parameters.

Step 3: Starting the RDE Server

Start and Stop of the RDE Server are carried out using the script “rde/bin/rde.sh“ (Linux) or “rde/apache-tomcat-6.0.26/bin/startup.bat“ (Windows).

2.4 Separate Installation

The separate installation of the prudsys RDE Server (deployment) is executed using a web application archive (WAR file) named "rde_server.war". The packaged application has to be copied into the deploy directory of the web application server.

2.4.1 Required installation files

- rde_server.war
- re_server_tomcatrealm.jar
- h2-1.1.116.jar

2.4.2 Deployment

Step 1:

- rde_server.war is put into the deploy folder of the web application server, for example: /TOMCAT_HOME/webapps/

Step 2:

- re_server.tomcatrealm.jar and h2-1.1.116.jar are put into the library folder of the web application server, for example: /TOMCAT_HOME/lib/

Step 3:

- The TOMCAT RAM has to be adapted, for example within the file /etc/default/TOMCAT-HOME/bin/setenv.sh
- example: CATALINA_OPTS=”-Xmx1024M”
- Note: Lib files must exist within the class path of the Tomcat

Step 4:

- Create and configure an operational path for the RDE Server within the Tomcat class path or within a separate file
- for example: -Dre_server.data.dir=/home/RDE

Step 5:

- Set the data base port for the RDE Server
- within the Tomcat class path or within a separate file within TOMCAT_HOME/bin/
- -Dre_server.h2.db.port=9092

Additionally, it is advisable to configure the garbage collector in the following way:

- within the Tomcat class path or within a separate setenv.sh file in TOMCAT_HOME/bin/
- -verbose:gc
  -XX:+CMSIncrementalMode
  -XX:+UseConcMarkSweepGC
  -XX:+DisableExplicitGC
  -Xloggc:/RDE/gclog/gc.out

Step 6:

- Start the Tomcat

2.5 Testing the installation

If the deployment was successful, the RDE Server files will be created within the folder “re_server“ within the operational path that was set during the configuration:

- Operational path/
  - re_server/
    - client/
    - config/
    - db/
    - fonts/
    - logs/
    - newsletter/
    - re_<RDE-ID>
    - report

Additionally, a new folder rde_server/ will be created within the directory /TOMCAT_HOME/webapps/.

Now the RDE client (see Chapter 3) can be used to test if a connection to the server can be created successfully (see Chapter 4.1).

Note: The standard port of the server normally is 8080. If you want to user another port, the parameter RDE_TOMCAT_CATALINA_PORT has to be changed (therefore see Table 1).

2.6 Updating the RDE Server

To update the RDE Server, the file rde_server.war has to be replaced. Therefore, the following steps are recommended.

Step 1:

Deactivate each RDE application within the client. This process is described in Chapter 5.3.

Step 2:

Stop the Apache Tomcat.

Step 3:

Back up the directory data/re_server.

Step 4:

Delete the file TOMCAT_HOME/conf/Catalina/localhost/rde_server.xml.

Step 5:

Delete the directory TOMCAT_HOME/webapps/rde_server.

Step 6:

Copy and paste the new rde_server.war in directory TOMCAT_HOME/webapps.

Step 7 (optional):

Copy and paste new libraries in directory TOMCAT_HOME/lib.

Step 8:

Start the Apache Tomcat.

Step 9:

Check the update process:

- Test the Catalina log within directory TOMCAT_HOME/logs for errors.
- Check the file data/re_server/config/revision.txt for the latest revision number.
- Additionally check the latest server log in directory data/re_server/logs/serverlog for the convert information, e.g. *"finishing Convert release 2.9.3 -> 2.10.0"*

Note: A new RDE Client will be provided for each new server revision. Only the corresponding RDE Client operates with all new functions.

3 RDE Client

3.1 Installation

RDE Client

The RDE Server features a user interface (RDE Client) which can be executed on any Windows or Unix system or via web browser. The only requirement is a Java Virtual Machine (version 6 or higher).

Linux version

The RDE Client can be started through the execution of rde_client-<version>.jar. The directory can be chosen freely by the user.

Windows version

There also exists an RDE Client version for Windows users. This version contains an install wizard which guides the user comfortable through the installing process.

Webstart version

Additionally, the Client can be started via web browser using the following URL:

*http://host:port/rde_server/client/**webstart***

There are some differences between the normal versions and the webstart. The webstart client only knows his own server. There is no possibility to add, edit or delete this server. Only users with admin rights can add new known servers within the Server Management.

Additionally, the function auto connect is disabled. The login information user and password are required for every connection.

Figure 1: webstart client

3.2 Structure

With the help of the RDE Client, all administrative tasks necessary for the creation and configuration of prudsys RDE modules can be executed comfortably without any programming knowledge.

The RDE Client has the following structure with three main components:

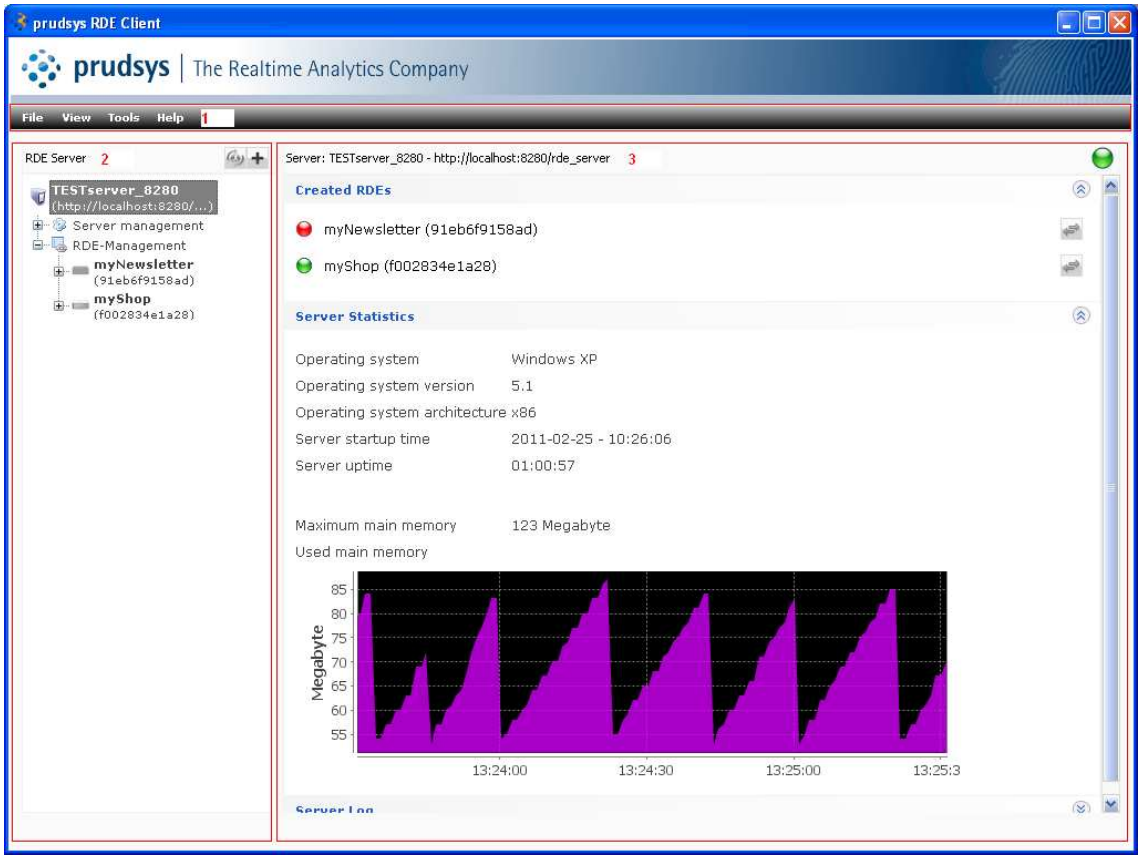


Figure 2: RDE Client – three components

- Main Menu (Figure 2 – component 1): general RDE Client settings
- Explorer (Figure 2 – component 2): navigation to different areas, such as the RDE server management (“Server management”) and the corresponding RDE applications within the RDE server (“RDE Management”)
- Content Area (Figure 2 – component 3): display of available information and dialogues corresponding to the area which has been selected in the explorer.

One client can be used for the administration of several RDE servers. All available servers will appear within the explorer, where they can be selected for further configuration. Every RDE server may contain several RDE applications which also will appear within the explorer.

4 RDE Server Management

4.1 Creating a server connection

In order to manage one or more RDE servers and its corresponding RDE applications, a connection to the RDE server has to be created. Select File → New Connection in the main menu of the client or click the plus button () in order to start the following connection dialogue:

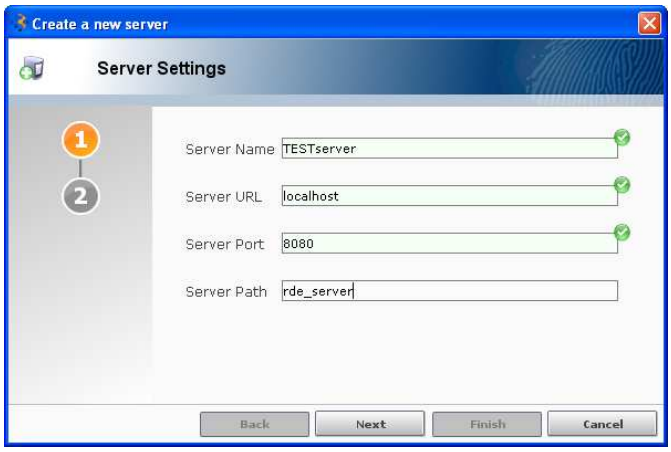


Figure 3: server connection wizard

The default access data to a recently installed server is:

- username: admin
- password: admin

This data should be changed for security reasons (therefore see Chapter 4.3.3, step 1).

4.2 Connecting the client to a server

Before you can use the client to administer, configure and monitor the different RDE applications, the client has to be connected to an existing RDE Server whenever it is started.

Therefore, select an RDE Server from the list of available servers in the explorer and press the button Connect ( ). Alternatively, right-click the RDE Server in the list and select “Connect” from the context menu.

Figure 4: connecting the client to a server

As soon as the server is connected, an overview will appear within the content area. A green icon at the top of the content area signalizes that the selected server is connected ( ).

4.3 Server Settings

The server management area provides the following functionalities:

- Server Statistics: compact overview over the most important system characteristics like the operating system. The overview shows at a glance if the RDE Server is running properly or not.
- User Management: creation and configuration of user accounts and their corresponding rights.
- Roles Management: creation and configuration of user roles.
- Rights Management: displaying of all available user rights.

4.3.1 Rights Management

Within the prudsys RDE Server, all resources are controlled through a user, roles and permission management. This allows the client to set specific permissions for different business users.

The following permissions are pre-configured within the server: view, add, edit and delete and their different combinations.

The rights manager only shows the different available permissions and their description. It is not possible to create new permissions (Figure 5).

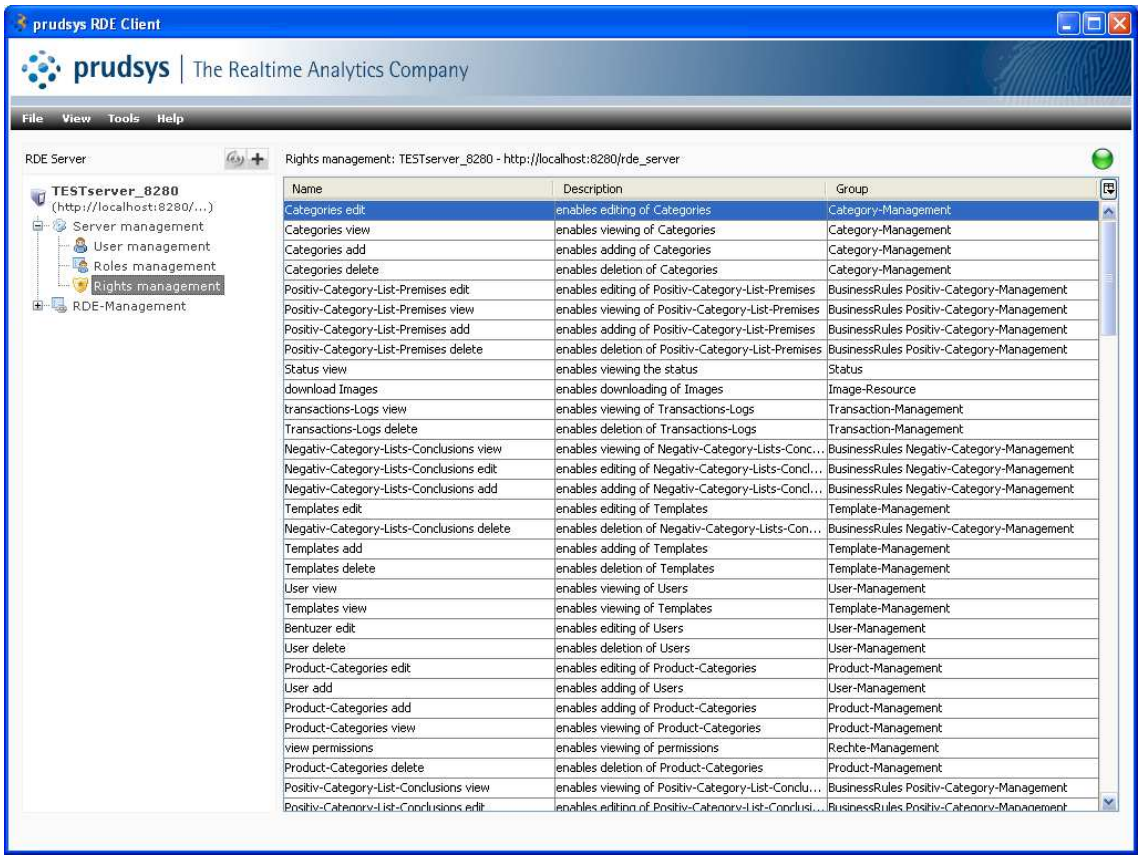


Figure 5: rights management

The RDE Server features over 30 right groups for the following main functionalities:

- RDE administration
- RDE monitoring
- Data management
- RDE handling
- Statistic management

In order to show the most needed permissions (to execute the HTTP requests within this manual) the following table is only an extract.

Note: In general, all requests including “admin” need permissions.

HTTP request	Permission	Right group	Description
Backup and restore an RDE application			
http://host:port/rde_server/admin/backup/export/<RDE-ID>/<LASTTRANSLOGS>	Create backup	Backup-Management	enables creating backups
http://host:port/rde_server/admin/backup/import/<FILENAME>	Restore backup	Backup-Management	enables restoring backups
http://host:port/rde_server/admin/backup/list	List backups	Backup-Management	enables listing backups
http://host:port/rde_server/admin/backup/download/<FILENAME>	Download backup	Backup-Management	enables downloading backups
http://host:port/rde_server/admin/backup/upload/<FILENAME>?override=true|false	Upload backup	Backup-Management	enables uploading backups
http://host:port/rde_server/admin/backup/remove/<FILENAME>	Remove backup	Backup-Management	enables removing backups
Control processes			
http://host:port/rde_server/admin/processes	Control processes	Process-Management	enables controlling of processes
http://host:port/rde_server/admin/processes/<PROCESS-ID>	Control processes	Process-Management	enables controlling of processes
http://host:port/rde_server/admin/processes/<PROCESS-ID>/pause	Control processes	Process-Management	enables controlling of processes
http://host:port/rde_server/admin/processes/<PROCESS-ID>/resume	Control processes	Process-Management	enables controlling of processes
http://host:port/rde_server/admin/processes/<PROCESS-ID>/cancel	Control processes	Process-Management	enables controlling of processes
Event log			
http://host:port/rde_server/admin/logs/eventlog/res/<RDE-ID>/list	Event log	Event Log Management	viewing and deleting of event logs
Get log level			
http://host:port/rde_server/admin/logs/loglevel	View RDE log level	Log Management	enables viewing the current log level
Interactive statistic			
http://host:port/rde_server/admin/res/statistic/interactive?locale=en|de	Interactive statistic view	Interactive Statistic	enables viewing of the interactive statistic
	Download images	Image Resource	enables downloading of images

HTTP request	Permission	Right group	Description
Interactive newsletter statistic			
http://host:port/rde_server/ admin/res/newsletter/ statistic/interactive?locale=en|de	Interactive statistic view	Interactive Statistic	enables viewing of the interactive statistic
	Download images	Image Resource	enables downloading of images
	Newsletter statistic view	Newsletter statistic	Enables viewing search configuration newsletter statistic
Newsletter			
http://host:port/rde_server/admin/res /<NL-RDE-ID>/newsletter/ imagetemplate/newsletter_image.jrx ml	Newsletter add	Newsletter-Management	Upload of the newsletter template
http://host:port/rde_server/admin/ res/<NL-RDE-ID>/newsletter/ createtextimages/run	Newsletter view	Newsletter-Management	Execute the text image creation
http://host:port/rde_server/admin/ res/<NL-RDE-ID>/newsletter/ newsletterupdate/run ?lastdays=<DAYS> &shop_reid=<SHOP-RDE-ID>	Newsletter view	Newsletter-Management	Download the required translog files
Online CSV update for items, categories, item categories and banners			
http://host:port/rde_server/admin/res /<RDE-ID>/items/ insertOnlineData? separator=%7c&updateData=true	Products edit	Product-Management	enables editing of products
http://host:port/rde_server/admin/res /<RDE-ID>/categories/ insertOnlineData? separator=%7c&updateData=true	Categories edit	Category-Management	enables editing of categories
http://host:port/rde_server/admin/res /<RDE-ID>/itemcategories/ insertOnlineData? separator=%7c&updateData=true	Product categories edit	Product-Management	enables editing of product categories
http://host:port/rde_server/admin/res /<RDE-ID>/banners/ insertOnlineData? separator=%7c&updateData=true	Banners edit	Banner-Management	enables editing of banners
Online update for one attribute of one item or banner			
http://host:port/rde_server/admin/res /<RDE-ID>/event/ updateItemAttribute ?item=<ITEM-ID>&attribute= <ATTRIBUTE>&value= <NEW-VALUE>	Get event information	Recommendation-Event-Management	enables getting event information, in this case: update an attribute for one item

HTTP request	Permission	Right group	Description
Online update for several attributes of one item or banner			
http://host:port/rde_server/admin/res/<RDE-ID>/event/updateItemAttributes?item=<ITEM-ID>&attribute=<ATTRIBUTE>&value=<NEW-VALUE>	Get event information	Recommendation-Event-Management	enables getting event information, in this case: update several attributes for one item
Online update for one attribute of several items or banners			
http://host:port/rde_server/admin/res/<RDE-ID>/event/updateItemsAttribute?items=<ITEM-ID-1>,<ITEM-ID-2>&attribute=onlineFlag&values=<VALUE-1>,<VALUE-2>	Get event information	Recommendation-Event-Management	enables getting event information, in this case: update an attribute for several items
Request log			
http://host:port/rde_server/admin/logs/requestlog/res/<RDE-ID>/list	Request log	Request Log Management	viewing and deleting of request logs
Save models			
http://host:port/rde_server/admin/res/<RDE-ID>/event/savemodels	Get event information	Recommendation-Event-Management	enables getting event information
Server status			
http://host:port/rde_server/admin/status	Status view	Status	enables viewing the status
Set control group			
http://host:port/rde_server/admin/res/<RDE-ID>/event/setcontrolgroup/sid/<SESSION-ID>/<CONTROLGROUP-ID>	Get event information	Recommendation-Event-Management	enables getting event information, in this case: set control group
Set log level			
http://host:port/rde_server/admin/logs/loglevel/<level>	Set RDE log level	Log Management	enables viewing the current log level
Translog files			
http://host:port/rde_server/admin/res/<NL-RDE-ID>/translogs	Transaction logs view	Transaction-Management	List all translog files
http://host:port/rde_server/admin/res/<NL-RDE-ID>/translogs/translog.<DATE>	Transaction logs view	Transaction-Management	Get one translog file as zip file
http://host:port/rde_server/admin/res/<NL-RDE-ID>/translogs/archive	Transaction logs view	Transaction-Management	Get all existing translog files as zip files
http://host:port/rde_server/admin/res/<NL-RDE-ID>/translogs/archive?lastdays=<DAYS>	Transaction logs view	Transaction-Management	Get the translog files of the last <days>

HTTP request	Permission	Right group	Description
View online items			
http://host:port/rde_server/admin/res/<RDE-ID>/items/online/<ITEM-ID>	Products view	Product-Management	enables viewing of products
http://host:port/rde_server/admin/res/<RDE-ID>/banners/online/<BANNER-ID>	Banners view	Banner-Management	enables viewing of banners
http://host:port/rde_server/admin/res/<RDE-ID>/categories/online/<CATEGORY-ID>	Categories view	Category-Management	enables viewing of categories

Table 2: rights management

4.3.2 Roles Management

For an easier handling, several permissions can be grouped in roles. Every user is assigned with one or several roles, not with single permissions. New roles can be created and existing roles can be edited at any time. Users can only delete the roles they have created by their own. The pre-configured roles system administrator and customer administrator can not be deleted for security reasons.

Step 1:

Create a new role with a role name and a short description

Step 2:

Set permission within the tab “Role Rights” by marking the corresponding check boxes (Figure 6).

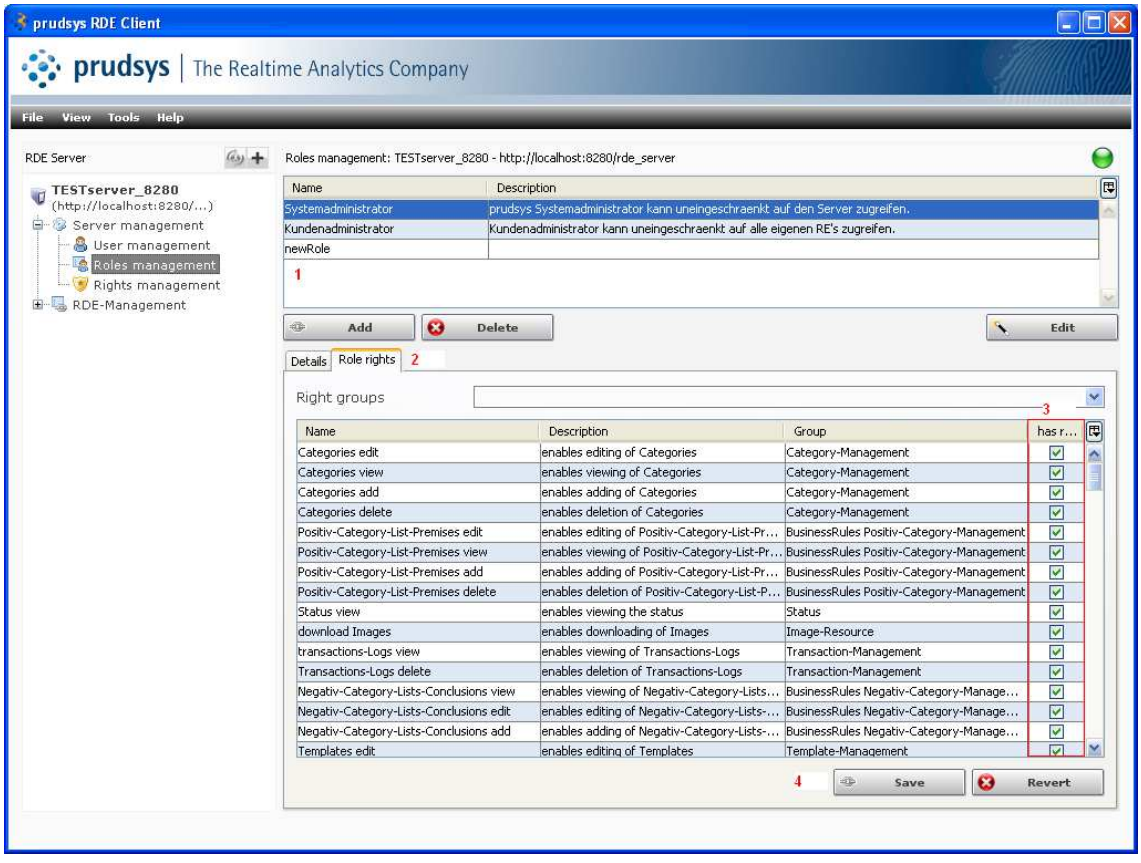


Figure 6: assigning permissions to roles

4.3.3 User Management

Within the user management area, users can be created, deleted, edited and assigned with existing roles. The pre-configured standard users “system“ and “admin“ can not be deleted for security reasons.

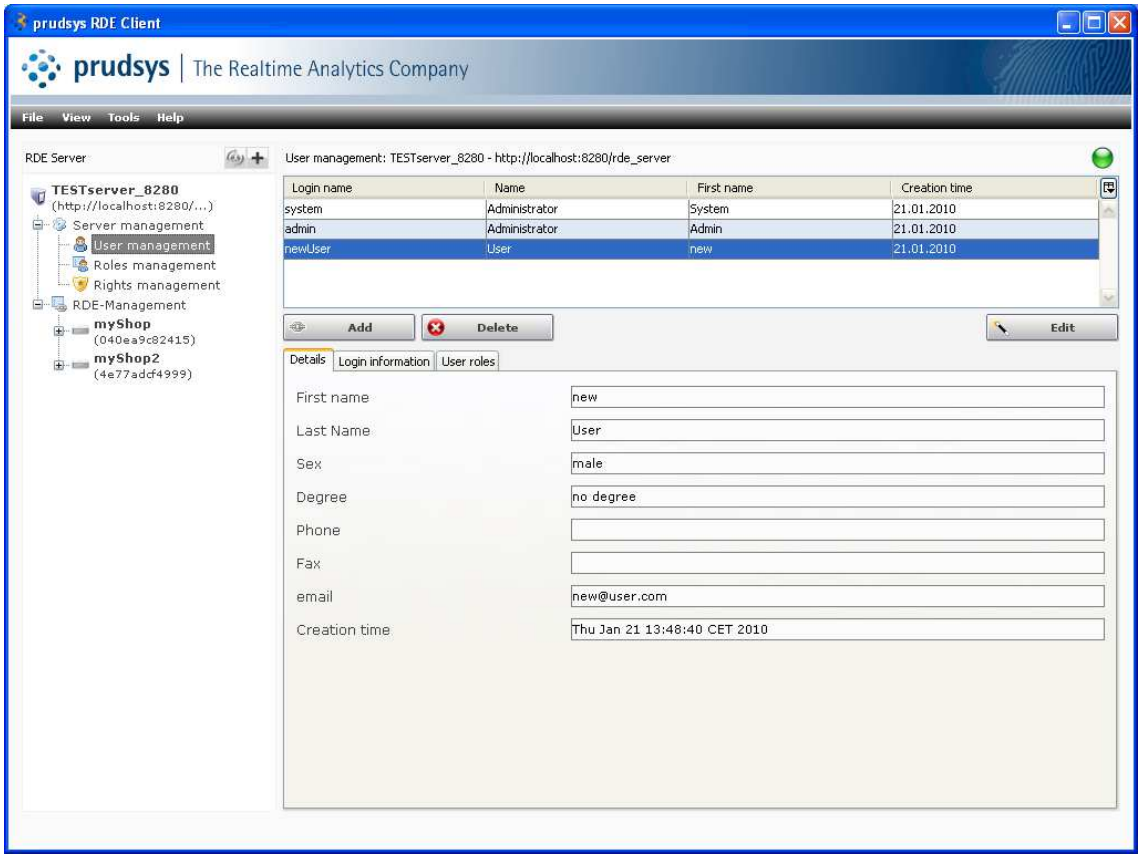


Figure 7: user management - overview

Step 1:

Create a new user by clicking the button Add and set the following attributes (attributes marked in bold are obligatory):

- **first name**, **last name**
- sex, degree
- telephone, fax, **e-mail**
- **username**, **password**

The new user will appear in the list (Figure 7) beyond the existing users “system” and “admin”.

Step 2:

Assign one or more roles within the tab “User Roles” by marking the corresponding check boxes (Figure 8). Changes within the settings have to be saved using the button Save. Otherwise they will get lost.

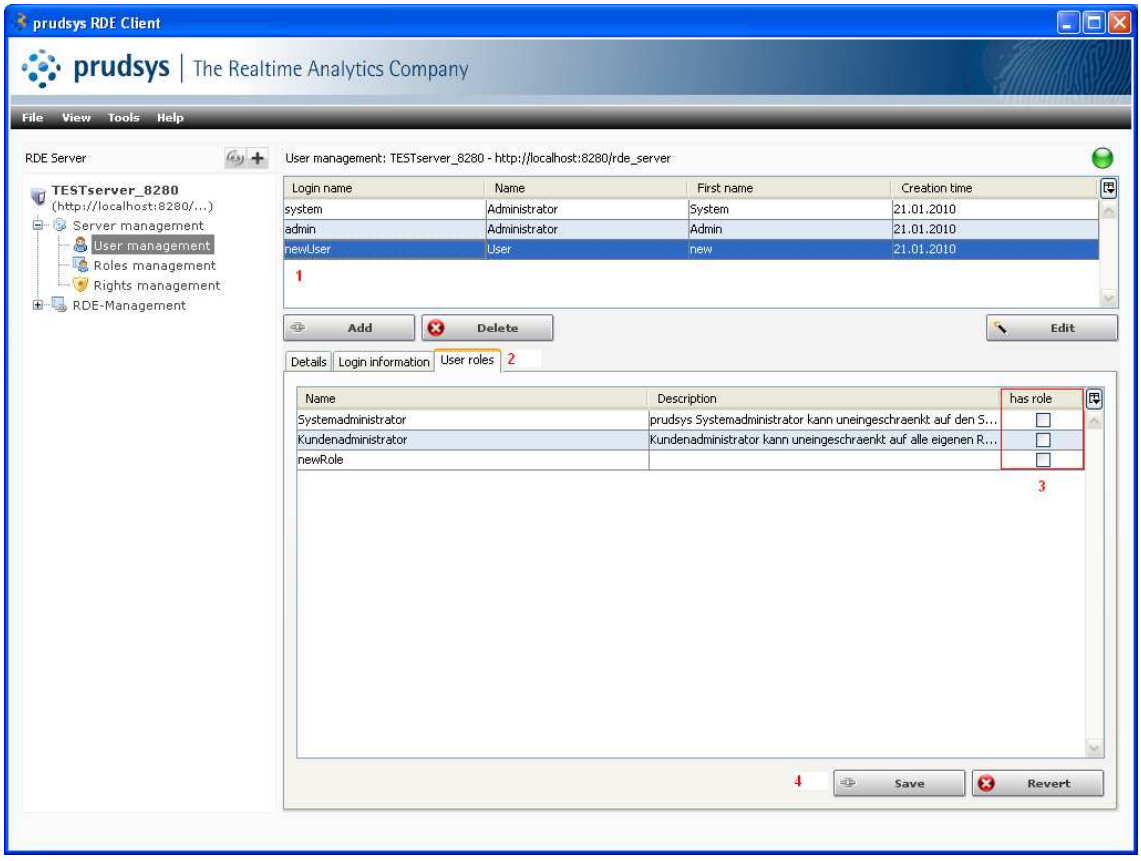


Figure 8: assigning roles to users

5 General RDE Application Management

5.1 Overview

The RDE Management provides all necessary functionalities for creation and configuration of RDE applications.

Several RDE applications can be created within one RDE server. Click on an RDE server → RDE Management within the explorer to see all corresponding RDE applications and their status overview.

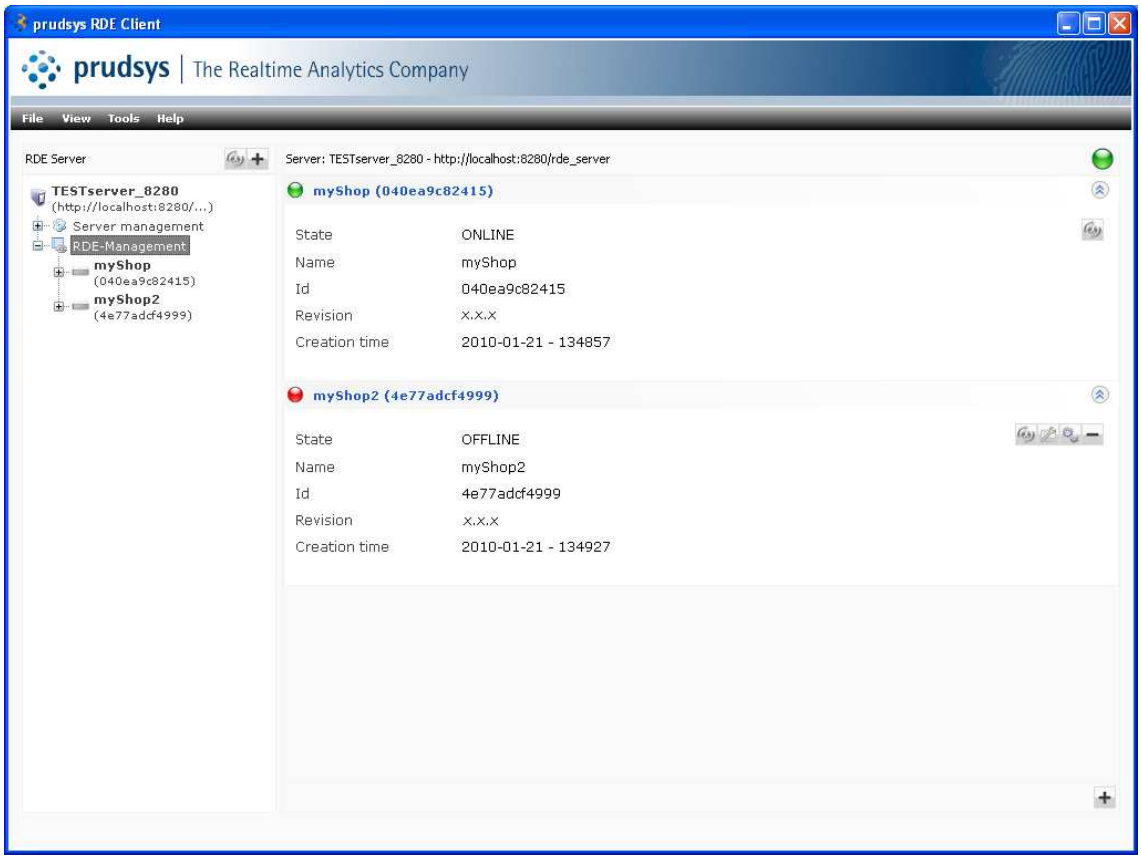


Figure 9: RDE management - overview

5.2 Creating a new RDE application

To create a new RDE application within your RDE Server, select RDE Server → RDE Management and click the button “New RDE” (+) at the lower right corner of the content area. A wizard will guide you through the creation process.

It is possible to create the RDE application with a specific RDE-ID. Valid characters are ‘a-z’, ‘A-Z’, ‘0-9’ and ‘_’. Space characters are not valid.

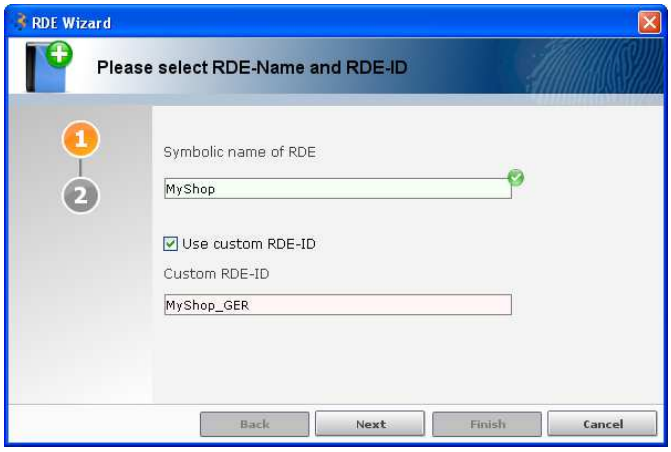


Figure 10: creating a new RDE application

The latest created RDE application will appear in the list of all available applications within the selected RDE Server.

Directory structure

Additionally, within the operational path of the RDE Server a new directory will be created:

- re_server/
  - re_<new_RDE-ID>/
    - config/
    - data/
    - datamigration/
    - download/
    - logs/
    - newsletter/
    - report/
    - tempdata/
    - templates/
    - translog/
    - upload/

5.3 Activating and deactivating an RDE application

To activate or deactivate an RDE application, select the RDE Server → RDE Management → RDE Application (e.g. “myShop”) in the explorer. An overview of the selected application appears as well as all configuration possibilities within the content area.

In order to activate the RDE application, select “Activate” within the RDE application menu (Figure 11). The icon in the headline indicates if the RDE application is active (green icon) or not (red icon).

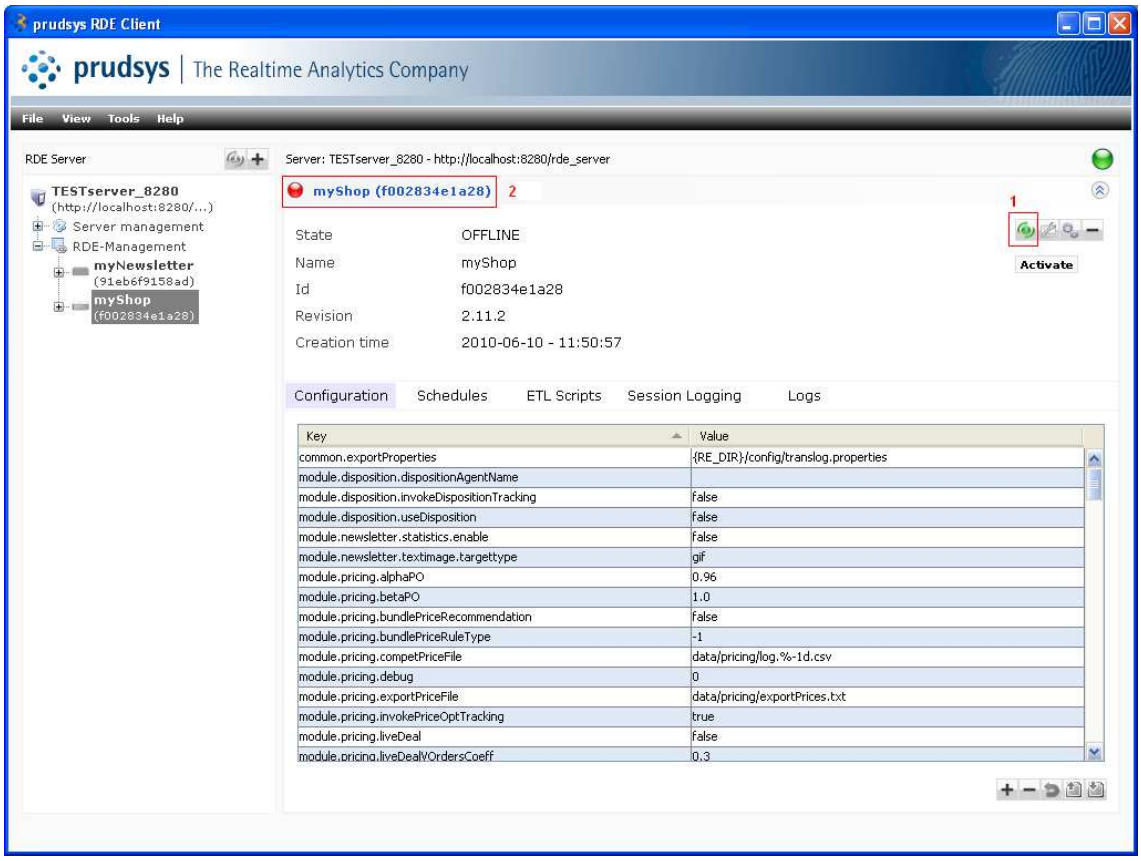


Figure 11: activating and deactivating an RDE application

If you want to configure an RDE application, it has to be deactivated first and then re-activated when the configuration process is finished.

5.4 Backup and restore an RDE application

The RDE Server provides a method to backup and restore RDE application data. The following steps describe the backup and restore process in detail.

Please note: Users need permissions to backup and restore an RDE application (therefore see Table 2).

Backup an RDE

Step 1:

Deactivate the RDE application.

Step 2:

Start the backup process by sending the PUT request e.g. with curl:

curl –X PUT --digest –u username:password
http://host:port/rde_server/admin/backup/export/<RDE-ID>/<LASTTRANSLOGS>

The parameter “last translogs” defines the last days of translog files which will be included into the backup.

The backup will be stored within the directory re_server/backup/ as a ZIP file.

Additionally, there is the possibility to pause, resume or cancel the backup process. Therefore, send the following requests via browser.

To monitor the backup process:

http://host:port/rde_server/admin/processes/<PROCESS-ID>

To stop the backup process:

*http://host:port/rde_server/admin/processes/<PROCESS-ID>/**pause***

To resume the backup process:

*http://host:port/rde_server/admin/processes/<PROCESS-ID>/**resume***

To cancel the backup process:

*http://host:port/rde_server/admin/processes/<PROCESS-ID>/**cancel***

Step 3:

To download a created backup, send the following GET request via browser:

*http://host:port/rde_server/admin/backup/**download**/<FILENAME>*

All existing backup files can be listed via browser by requesting the URL:

*http://host:port/rde_server/admin/backup/**list***

Within this list, the following information is available:

filename | *reid* | *size* | *timestamp* | *version* | *lastNTranslogs* | *principal*

Restore an RDE

Step 1:

Check the current server and RDE application version. An RDE application can only be restored if server version and RDE version are completely equal. The server version can be checked via browser:

*http://host:port/rde_server/serverinfo/**version***

The RDE application version can be checked via browser by requesting the URL

*http://host:port/rde_server/admin/backup/**list***

Step 2:

If the RDE application already exists, deactivate the RDE first.

Step 3:

Start the restore process by sending the POST request e.g. with curl:

curl –X POST --digest –u username:password
*http://host:port/rde_server/admin/backup/**import**/<FILENAME>*

All existing backup files can be listed via browser by requesting the URL:

*http://host:port/rde_server/admin/backup/**list***

Additionally, there is the possibility to upload an existing backup by sending the POST request e.g. with curl:

curl –X POST –T "file path" --digest –u username:password
*http://host:port/rde_server/admin/backup/**upload**/<FILENAME>?override=true|false*

The restore process can also be paused and resumed like the backup process, but can not be cancelled. Therefore, use the requests described above (backup an RDE, Step 2).

Step 4:

To remove no longer required backups, send the following DELETE request e.g. with curl:

curl –X DELETE --digest –u username:password
*http://host:port/rde_server/admin/backup/**remove**/<FILENAME>*

5.5 Data Upload

All data required for the operation of the RDE server can be uploaded with the following approaches:

- CSV file upload

Structure

The following basic structure is required for each data file:

- A headline depending on the data file (separately explained in the next sections)
- Columns have to be separated by the sign pipe ("|")
- Data records have to be separated by line breaks, e.g. \n
- CSV files should only be saved in UTF-8 (without BOM) to make sure that special characters will be displayed properly

Required data

The data files for calculation of recommendations can be classified into three parts: basic, ideal and additional files. These are described in the next sections.

The **basic** data file is implicitly for installation of all RDE Modules:

- Product master data (items.csv)

Furthermore, we recommend these data files for **ideal** recommendation calculation:

- Categories (categories.csv)
- Category hierarchy (categoryhierarchy.csv)
- Product to catalogue relations (itemcategories.csv)
- Historic transaction data (transactions.csv)

Additional data for individual recommendations:

- Business rules
- Banner master data

The limit values for master data are the following:

Master data type	Limit value
Products and banners	1.000.000.000
Categories	1.000.000.000
Product to catalogue relations	1.000.000.000
Transaction data	1.000.000.000
Business rules	1.000.000.000
Online rules	1.000.000.000

Table 3: limit values for master data

5.5.1 Product Master Data

Product master data files (items.csv) contain all available information attached to the products. The column names for the standard attributes are defined in the following way:

Name	Description
pid	unique ID of the product
netUnitPrice	sales price
sku	additional product ID
name	product name
masterUID	master article ID (empty if there are no variants)
brand	brand
manufacturer	manufacturer
quantityUnit	quantity unit
quantity	quantity
netPurchasePrice	purchase price
strikeOutPrice	strike out price
reward	defines the value of the product for your company and sales strategy, necessary for the self learning and optimization mechanism of the engine
description	description of the product
URL	URL of the product
imageURL	URL of the picture which belongs to the product
onlineFlag	Binary flag which signalizes the availability of the product (value “1” or “true” for available, else for not available)
param1 to param20	parameters which can be set individually, for example for colour, gender (men’s, women’s, kids, …), etc.
stock	Products in stock; will be calculated in real time: (quantity – order), there will be no recommendations for stock = 0 (will be ignored if the column is empty)
type	type
rank	Attribute for top seller calculation (preCalcTopsellerType = 2)
initstock	Initial stock
reducedPrice	(Pricing) Reduced price, used for control group
minUnitPrice	(Pricing) The minimum discount of a product as percentage
maxUnitPrice	(Pricing) The maximum discount of a product as percentage
defaultVariationUID	Default variation UID which will be displayed as recommendation instead of the master ID

residualPrice	(Assortment Planning) residual price; It is essential: residual price < purchase price < sales price
oosCost	(Assortment Planning) out-of-stock cost rate; percentage rate of purchase price (value between [0..1])

Table 4: product master data

For the calculation of recommendations, the following data is required (minimum requirement):

- pid
- netUnitPrice
- reward

Hint: If there is no single reward data, the column netUnitPrice should be duplicated.

Other attributes depend on the requirements of the online shop and the chosen output format (in the case of html, for example, all the information that has to be displayed must be uploaded).

The file items.csv should be similar to the following example:

pid	master UID	name	description	URL	netUnitPrice	**reward**	onlineFlag	defaultVariationUID
3		radio B16			140.45	140.45	0	3
4		TV set Trinitron			897.00	897.00	0	4
5		alarm clock HRadio			54.99	54.99	1	5
8		cable AP			4.75	4.75	1	8A
8A	8	cable AP 177			4.75	4.75	1	8A
8B	8	cable AP 178			4.95	4.95	0	8A
19		coin cell K7			0.55	0.55	1	19

Table 5: example of product master data

Product master and variants

Provided that within the internal data model product masters and variants are used, it is important to indicate the master-variant relations. Usually online shops operate on the basis of product variants, i.e. the size variant of a product which is added to a shopping cart is directly added to the order log and thus straight to the RDE. The RDE will transform the variant IDs into master IDs on the basis of the master-to-variant-mapping provided by the shop operator since recommendation rules are usually calculated on the

basis of master IDs. In order to display recommendations within the shop, the RDE will be requested for masters IDs. In order to implement this mechanism it is important to know the master-variant relationships.

The format of master variant mapping is important. Table 5 shows an example for the correct representation of the required information: For every variant (labelled with an alphabetic character) the corresponding master is indicated within a separate row. The availability of an article (onlineFlag) should be related to its master. Whenever a variant of a product is available in the row of the corresponding master ID, the column onlineFlag should contain “1“ or “true” (not case sensitive). Therefore, it is not necessary to indicate the availability of variants (see Table 5, the corresponding values are marked in grey).

Usually the RDE will transform variant IDs into master IDs. In contrast, some shops need variation IDs as output. In this case, the column defaultVariationUID has to be filled with the required variation ID. This variation will be used as output instead of the master ID.

Please note: There can only be one defaultVariationUID per master (e.g. variation 8A within Table 5). The column defaultVariationUID also has to be filled for products without variations. In this case, the column will be filled with the pid. Products without defaultVariationUID won't be recommended.

Obligatory columns

The columns marked in bold are indispensable for the calculation of recommendation rules. Depending on the shop requirements and the return format of the recommendations, it is possible to pass further attributes into the RDE. If the return format should contain the product name it has to be passed to the RDE within the items.csv file.

Format of netUnitPrice

Concerning the column netUnitPrice, please take into account that the given format is mandatory. The decimal places have to be separated from the pre-decimal places by a decimal point (English format). The price type, for example retail price or purchase price, is not relevant for the prudsys RDE, but it should be set consistent for all products. Also any virtual price which acts as an indicator of the value of a product can be processed by the RDE.

If the return format of the recommendations must contain the price of the product, the prudsys RDE uses the information provided within the master data file. In this case, the price type should be equivalent to the price which is displayed within the web shop.

With regards to the column reward, please note that the content of this column corresponds to the column netUnitPrice. It should be delivered as a copy of the price column by the shop operator. The reward column can be added to the file ex post by prudsys, though generating an additional data pre-processing step for the prudsys RDE.

OnlineFlag

In order to avoid the display of temporarily not available products, it is advisable to administer the availability of the products within the master data file using the column onlineFlag. Nevertheless, the availability of the products should be filtered again within the shop frontend.

Note: Temporarily not available products should be left within the product master data, because otherwise all related product rules will be deleted.

5.5.2 Catalogue Hierarchy

Catalogue hierarchies (categoryhierarchy.csv): hierarchies within the given catalogue and its corresponding categories (categories.csv). For every catalogue, a hierarchy tree with a complete list of all categories has to be created. The following information is required:

- cid: categoryID,
- cidParent: parent category ID

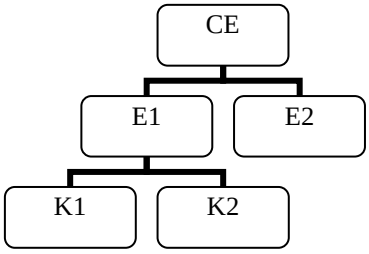
The category ID (cid) must correspond to the category ID of the product assignment in Chapter 5.5.3 and the parent category ID must be the ID of the next higher category.

Table 6 is an example of a three-stage category hierarchy. Consumer Electronics (CE) already is a root category and has no parent ID. To allocate a virtual root category additionally an arbitrary string, e.g. "-", has to be assigned. This could be helpful to unite all main shop categories and make use in recommendation calculation.

cid	cidParent
CE	-
El	CE
E2	CE
K1	El
K2	El

Table 6: example of catalogue hierarchy data

The described hierarchy data could be illustrated in the following way (not using the virtual root category "-"):



Additionally, all category IDs which appear within the hierarchy tree have to be listed within the separate file categories.csv (therefore see 5.5.4).

The category hierarchy is e.g. required for level based recommendations and the definition of negative lists ("black lists"). For the simple calculation of recommendation rules, it is sufficient to know only the parent category of a product. For the sake of completeness, the category hierarchy should be provided nonetheless.

5.5.3 Product to Catalogue Relations

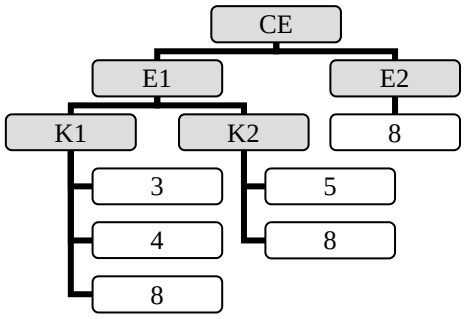
If the products are grouped in different catalogues, this information should also be made available to the engine. The file itemcategories.csv should contain the following information:

- pid: articleID,
- bid: bannerID
- cid: categoryID

pid or bid	cid
3	K1
4	K1
5	K2
8	K1
8	K2
8	E2

Table 7: example of product / banner to catalogue data

The described catalogue data can be illustrated in the following way:



Note: A product or banner can be grouped in more than one category and level, e.g. product 8.

Additionally, all category IDs which appear within the product to catalogue relation have to be listed within the separate file categories.csv (therefore see 5.5.4).

The category assignment should be carried out on the basis of the master IDs and at the lowest category level. If there is a master variant relationship like in the example in Table 5, always the master ID is used (although the column is named “pid”).

5.5.4 Categories

The file categories.csv has to contain the complete list of all category IDs within the hierarchy (therefore see 5.5.2, including virtual categories like “-”) and the product to catalogue relations (therefore see 5.5.3) using the identifier “cid“. The column “category name” can be set optionally.

The following category data is based on the previous examples.

cid	cname
K1	name of K1
K2	name of K2
El	name of El
E2	name of E2
CE	name of CE
-	root

Table 8: example of category data

5.5.5 Historic Transaction Data

Transaction data files (transactions.csv) normally contain information about the content of orders or shopping carts. The following columns are required:

- tid: transactionID (shopping cart ID, order ID),
- pid: itemID (productID)

sorted by transactionID.

Example: Three orders are given. The first one contains the products 5 and 8A, the second only product 4, the third contains products 3, 8B and 19. The attributes “tid” and “pid” may also contain strings like “A7” or “IDSESSION88”. If the data model does not contain the mapping of product masters and variants, please indicate an article ID according to your internal guidelines instead of the variant ID (which is used to specify colours, sizes, etc.).

tid	pid
4	5
4	8A
9	4
11	3
11	8B
11	19

Table 9: example of transaction data

5.5.6 Business Rules and taxonomies

Option 1: CSV files

The RDE Server allows setting Business Rules to model certain sales strategies and business scenarios which override the automatically generated recommendation rules of the RDE Server. The user can choose between two options. He/she can opt for the configuration of globally applicable rules which are stored in special CSV files. This option will be described in the following.

Option 2: templates

The second option is the definition of business rules or filter on the level of certain recommendation requests. The configuration of this type of rules has to be undertaken within the different recommendation templates (therefore see Chapter 5.10.3).

Constraints can easily be set via lists and rules. As those are completely cached, the shop performance will not be influenced. The RDE Server will only require slightly higher storage capacities. However, those filter rules are global, which means they will be applied within the complete online shop. If filters and rules are defined within the recommendation templates, they can be configured separately for every recommendation request.

Note: If both global and request specific filter rules are set, the global rules always override the request specific rules.

There are three differentiation criteria for business rules which are described in the following. All three criteria can be combined.

Lists / Rules Lists consist of only one column A and contain a certain number of products or categories. It is necessary to specify if a list refers to the premise (input) or the conclusion (output). Rules always consist of two columns A and B and represent the conclusion rule A → B.

Product / Category Level Constraints can be defined on the level of products or categories. The product level consists of the product master and its variations (size, colour, etc.). Categories can be taken from any level of the category hierarchy. The prudsys RDE Server supports multiple taxonomies (e.g. shop categories and brand categories). Every category has to be related to a taxonomy name to make sure that it is assigned properly.

Positive / Negative Positive means that the list / rule will **always** apply. Negative means that the list / rule may **never** apply. For example, the negative rule A → B means that in the case of product A, product B must not be recommended. The positive rule A → B means that in the case of product A, product B always has to be recommended. If there are more positive rules for A, their corresponding products can also be recommended (but no other products).

On the basis of those three criteria, the following 12 different combinations are possible.

List / Rule	List								Rule			
Product / Category	Product				Category				Product		Category	
Positive / Negative	Positive		Negative		Positive		Negative		Positive	Negative	Positive	Negative
Premise / Conclusion	P	C	P	C	P	C	P	C				
Type	1	2	3	4	5	6	7	8	9	10	11	12

Table 10: combinations of business rules

Example 1: Type 4 Negative list on product level as conclusion (type 4):

pid
345678
667767
M5536

The products 345678, 667767 and M5536 must never be recommended. If this list had been specified as premise, those products would never receive a recommendation.

Example 2: Type 11 Positive rules on category level (Type 11):

cidPremise|cidConclusion
c1|c2
c1|c3
c3|c1

Products from the category c1 may only receive product recommendations from the categories c2 and c3. Products from the category c3 may only receive product recommendations from the category c1.

Note: As can be seen in the example above, it is not possible to define multilateral relations. The relations c1-c3 and c3-c1 have to be defined separately.

Every constraint type (1 to 12) has to be defined within a separate file. Lists only consist of one column A, rules always consist of two columns A and B.

The following business rules types are available.

Type	CSV filename	Columns	Description
Product level filtering			
9	br_pos_item_rules	pidPremise| pidConclusion	Product positive rules: only the conclusion product will be recommended in the case of the premise product
10	br_neg_item_rules	pidPremise| pidConclusion	Product negative rules: the conclusion product will never be recommended in the case of the premise product
1	br_pos_item_list_premise	pid	Product positive list premise: only the products in the list will receive recommendations
2	br_pos_item_list_conclusion	pid	Product positive list conclusion: only the products in the list may be recommended
3	br_neg_item_list_premise	pid	Product negative list premise: the products in the list will never receive a recommendation
4	br_neg_item_list_conclusion	pid	Product negative list conclusion: the products in the list must never be displayed as a recommendation
Category Level Filtering			
11	br_pos_category_rules	cidPremise| cidConclusion	Category positive rules: only products from the conclusion category will be recommended for products in the premise category
12	br_neg_category_rules	cidPremise| cidConclusion	Category negative rules: products from the conclusion category may never be recommended for products in the premise category
5	br_pos_category_list_premise	cid	Category positive list premise: only categories and their products in this list will receive recommendations
6	br_pos_category_list_conclusion	cid	Category positive list conclusion: Only products from the categories in this list may be displayed as recommendations

Type	CSV filename	Columns	Description
7	br_neg_category_list_premise	cid	Category negative list premise: the categories and their products in this list may never receive recommendations of any type
8	br_neg_category_list_conclusion	cid	Category negative list conclusion: The categories and their products in this list may never be displayed as a recommendation

Table 11: list of all business rules files

Taxonomies

It is also possible to define different hierarchical classifications (taxonomies) and set rules and filters for each of them. Therefore, it is possible to define different business rules which have the same structure as the globally applicable rules.

For example, there could be two different hierarchies within one online shop. For each hierarchy, individual business rules can be defined.

Taxonomy 1			Taxonomy 2			
Taxonomy 1 – Business Rules			Taxonomy 2 – Business Rules			
Cat_Men	Cat_Women	Cat_Children	Cat_Pants	Cat_Pullover	Cat_Shoes	Cat_Shirts
Product 1					Product 1	

Table 12: example shops with taxonomies

One product could be listed, for example, within the category “men” and at the same time within the category “shoes”.

If you want to work with several hierarchies, this has to be configured within the menu “upload options“ (field Taxonomy, see Figure 14). Individual business rules can be defined for every hierarchy. Therefore, the business rules have to be assigned to a hierarchy name in the following way:

<taxonomy-name>.br_pos_item_rules.csv

If there are several rules for one product because it belongs to several taxonomies, the rules are linked with the logical conjunction AND.

5.5.7 Banner Master Data

Content recomms

Content such as teasers, pop ups and advertising and campaign banners highlight special products and services and bring them to the attention of online shop visitors. In most cases all visitors are presented with the same content no matter what their individual interests happen to be.

prudsys RDE will not only automate the real-time testing and display of content variants on the basis of overall visitor behaviour but it will also offer personalised output with content for the individual visitor.

For this process prudsys RDE is supplied with ready-made variants of display content such as teasers indicating various areas of your online shop. prudsys RDE will

automatically select and display the best advertising material from this library of ready-made material. Displays can be optimised on the basis of customer reactions to each piece of advertising. For example, if a particular banner attracts more clicks, shopping baskets and sales, it will be given a higher rating and will be displayed more often. Changes in customer behaviour due to external factors such as seasonal changes, changes in the weather and parallel marketing actions will be recorded and taken into account in displays. prudsys RDE ensures that only the most effective advertising is used without the need for any time-consuming manual testing.

If the system recognises a particular individual visitor and there is profile and transaction information (e.g. clicks, purchases, search queries) available for this visitor, the system will display advertising material which meets the interest profile of the customer. This high-relevance, personalised approach can be linked to the home or the landing page so that the visitor can be guided directly to the relevant content and offers.

Any type of content

prudsys RDE can control any type of content, document, link or FAQ. It will maximise conversion rates, sales and the browsing time of a visitor to the online shop (e.g. through content portals and service areas). Banner master data files (banners.csv) contain all available information attached to the banners. The column names for the standard attributes are defined in the following way:

Name	Description
bid	ID of the banner
netUnitPrice	sales price
sku	additional product ID
name	product name
masterUID	master article ID
brand	brand
manufacturer	manufacturer
quantityUnit	quantity unit
quantity	quantity
netPurchasePrice	purchase price
strikeOutPrice	strike out price
reward	defines the value of the product for your company and sales strategy, necessary for the self learning and optimization mechanism of the engine
description	description of the product
URL	URL of the product
imageURL	URL of the picture which belongs to the product
onlineFlag	Flag which signalizes the availability of the product
param1 to param20	parameters which can be set individually, for example for colour, gender (men’s, women’s, kids, …), etc.
stock	Products in stock; will be calculated in real time: (quantity – order), there will be no recommendations for stock = 0
type	Required to insert a banner during runtime
minDelivery	Minimal delivery of a banner
maxDelivery	Maximal delivery of a banner

rank	Attribute for top seller calculation (preCalcTopsellerType = 2)
initstock	Initial stock
reducedPrice	Reduced price
minUnitPrice	Sales discount as percentage, for example 50%: minimum price is up to 50% lower than the normal price
maxUnitPrice	Sales discount as percentage, for example 50%: maximum price is up to 50% higher than the normal price
defaultVariationUID	Default variation UID which will be displayed as recommendation instead of the master ID

Table 13: banner master data

For the calculation of banner recommendations, the following data is required (minimum requirement):

- bid

Additionally, the property *re.recommendations.banner = true* is required. For more information see Table 32.

Note: If banners are included into categories like products, the files categories.csv and itemcategories.csv have to be updated as well.

New file reitems

After uploading banners, the data file reitems.csv within directory /data will be filled automatically with both information from items.csv and banners.csv. This file should be checked whether all columns and rows contains the right information, especially column “type”=b for banners.

Attention: reitems

Attention: Due to the merging process, the column types (numeric or categorical) have to be identically within the items.csv and banners.csv. To configure different types, see Table 14.

5.6 Sending Data to the RDE Server

Two options to transfer data

There are two different ways to send the required data to the RDE Server. Option 1 (Chapter 5.6.1) is the manual upload of CSV files which can be executed comfortably via the client. The data is transferred directly from any directory to the RDE server.

Option 2 (5.6.2) is the automated data migration via the ETL library Kettle. Therefore, the data has to be made available within a special directory of the RDE Server. Together with integrated scheduler, any data migration process can be automated (see also Chapter 5.7).

Pros and cons

Option 1 is the best way for first shop RDE initialization. Users can check their data files for the first time, so problems with data formats can be exposed immediately. By contrast option 2 is the best way for periodical data update because there is no manual intervention needed. Besides there are more checking processes available with Kettle scripts.

Note: Copying the data manually into a server directory has no effect. Although the format of the file and the existence of all obligatory columns will be checked, there is no check of the content of the files.

5.6.1 Option 1: Upload of CSV files

Select the section Master Data to get an overview of the currently available data. If there is no data yet, an upload wizard will guide you through the upload process, which is started using the button Upload (Figure 12).

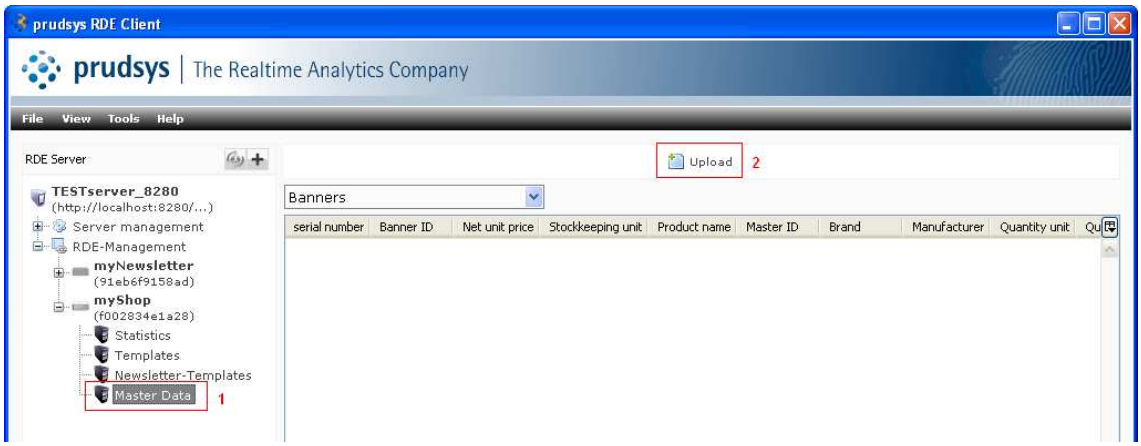


Figure 12: master data - overview

Offline upload

Step 1: Select upload file and type

Use the button Choose File (Figure 13) to select and upload a file from any directory. To define the data type which will be uploaded, select one of the existing file types from the drop down menu (Figure 13). The file type names indicate the file type:

- Product data

Banners
Products
Product categories
Product positive rules
etc.

- Category data

Categories
Category hierarchy
Category positive rules
etc.

- Special data

Transactions
Online recommendation rules
Transprobs

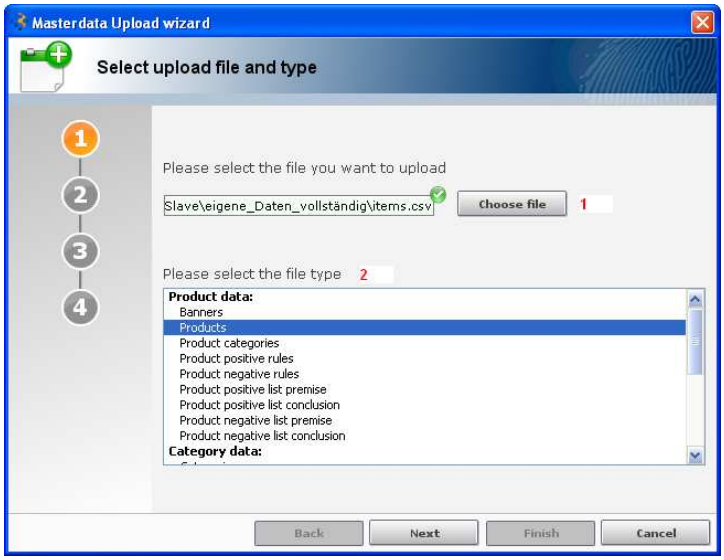


Figure 13: upload wizard

Step 2: Select the desired upload options

These options can be chosen at will, although the standard default settings are already pre-set.

It is necessary to fill the taxonomy name if there are different hierarchies and business rules within the shop (for more information see Chapter 5.5.6).

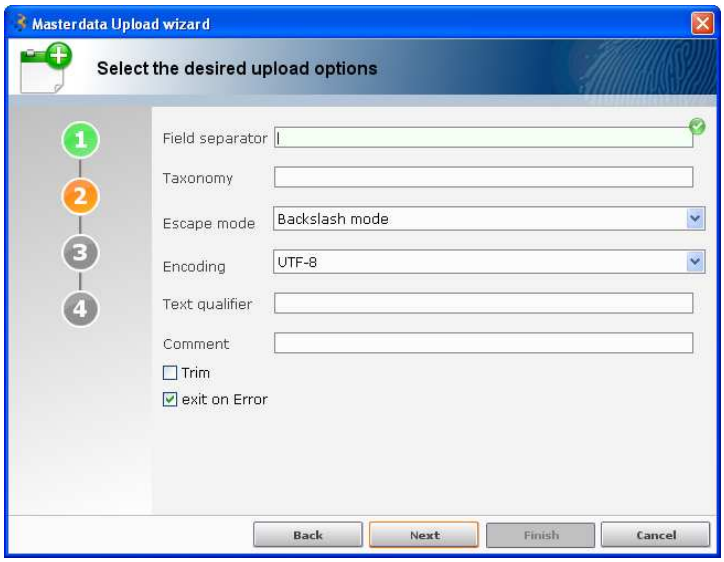


Figure 14: upload options

After uploading the files, the RDE has to be deactivated and re-activated, otherwise the changes will not become effective (see also Chapter 5.3).

CSV files should only be saved in UTF-8 (without BOM) to make sure that special characters will be displayed properly.

After the successful upload, the following window appears (Figure 15). The most common mistake which leads to an interrupt of the upload process is a wrong format of the master data files. The files must contain the exact number of columns with correct indication, otherwise the RDE Server is not able to assign the data properly. The order of columns is not relevant.

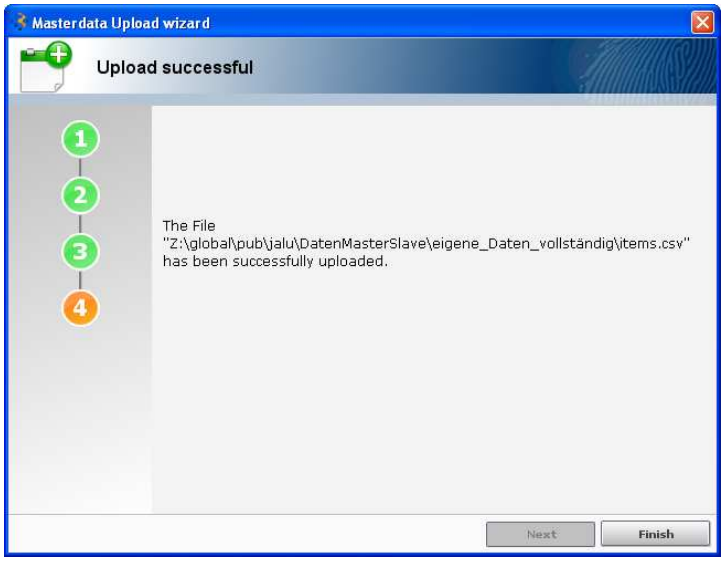


Figure 15: upload dialog

Afterwards, the uploaded data is displayed within the Master Data menu.

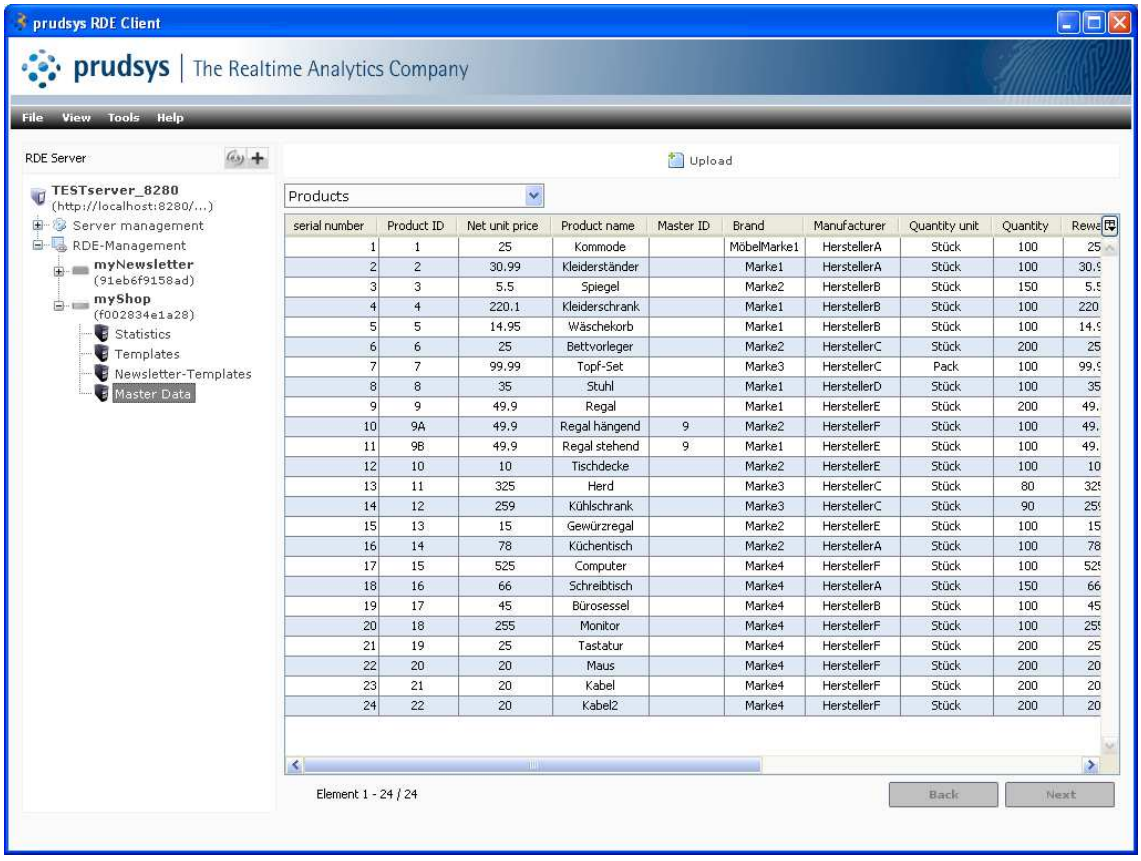


Figure 16: uploaded products

CSV upload during runtime

Additionally, there is the possibility to send new products or content to the RDE Server at runtime. The RDE Server differentiates between the upload of only one product (or content) and more than one product.

Please note that the data updated at runtime only persist as long as the prudsys RDE is active. Therefore, this functionality should only be used for unscheduled updates at short notice. Normal, scheduled updates of the product (or content) master data should be carried out via daily updates of the files items.csv or banners.csv.

Note: the described upload functionality can only be used when the prudsys RDE is online. The updated data will not be displayed in the client, because it is not stored within the master data. The following request can be used to show the new products (or content items):

http://host:port/rde_server/admin/res/<RDE-ID>/items/online/<ITEM-ID>

http://host:port/rde_server/admin/res/<RDE-ID>/banners/online/<BANNER-ID>

Note: Users need permissions to show new products (therefore see Table 2).

An alternative way to upload CSV files is described within Chapter 5.12.2.

Update attributes only

To update only single attributes of already existing products, the RDE Server provides an additional request (therefore see Chapter 5.12.2).

The following steps describe how to upload the CSV files during runtime.

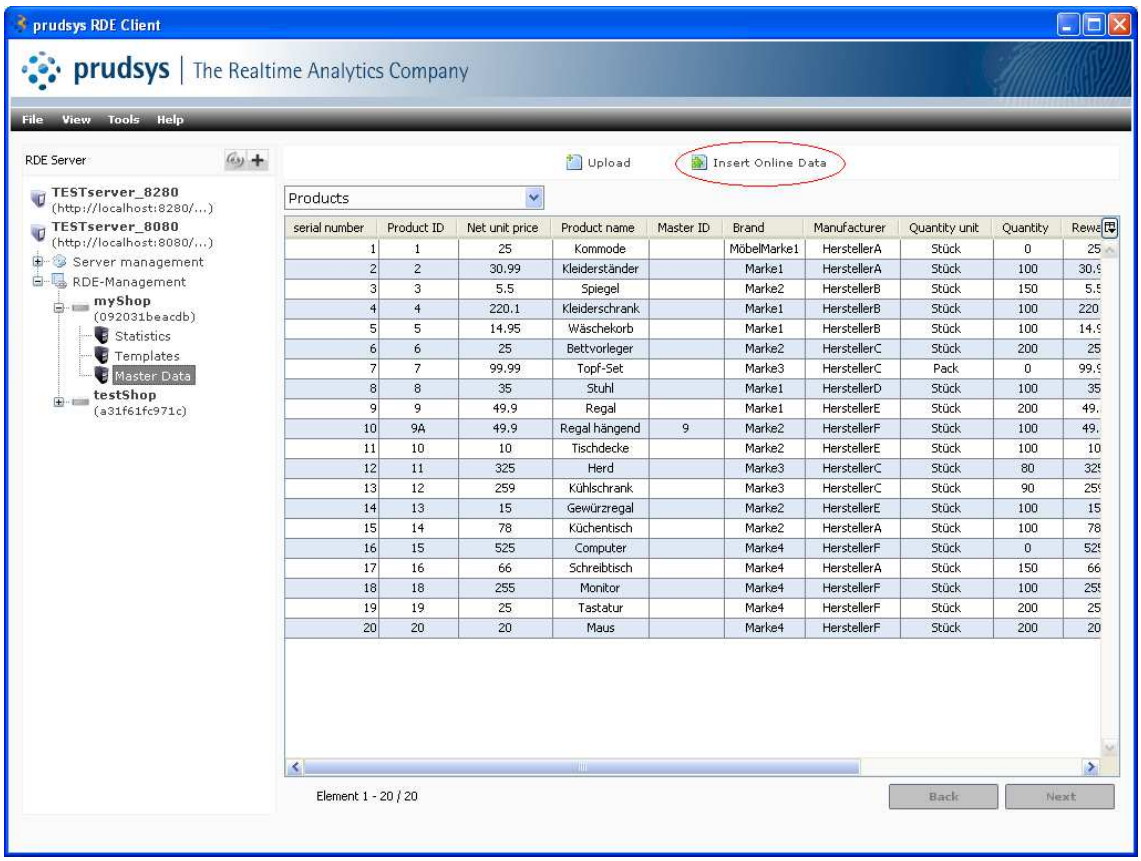


Figure 17: insert online data

Step 1: Choosing the insert mode

- Insert data from csv file
- Insert single object

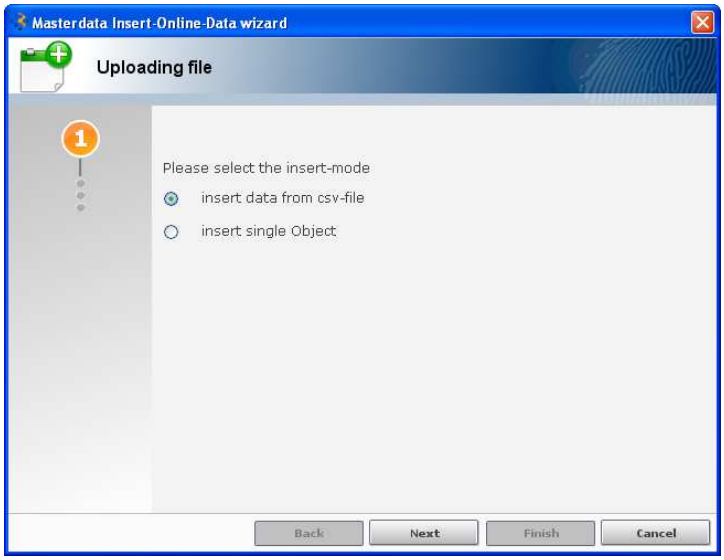


Figure 18: upload wizard

Option A: Insert CSV file

Step A2: Select upload file and type

- Banners, products, product categories or categories
- Mark button “override existing data” to update previous data (otherwise, only new data will be inserted)
- Mark button “merge with existing data” to update data column-wise (for more information see page 103)

Step A3: Select the desired upload options

- Same options like offline upload

Option B: Insert single object

Step B2: Select file type

- Banners or products
- Mark button “override existing data” to delete all previous data

Step B3: Filling the required information

- Required fields for products: pid, reward
- Required fields for banners: bid

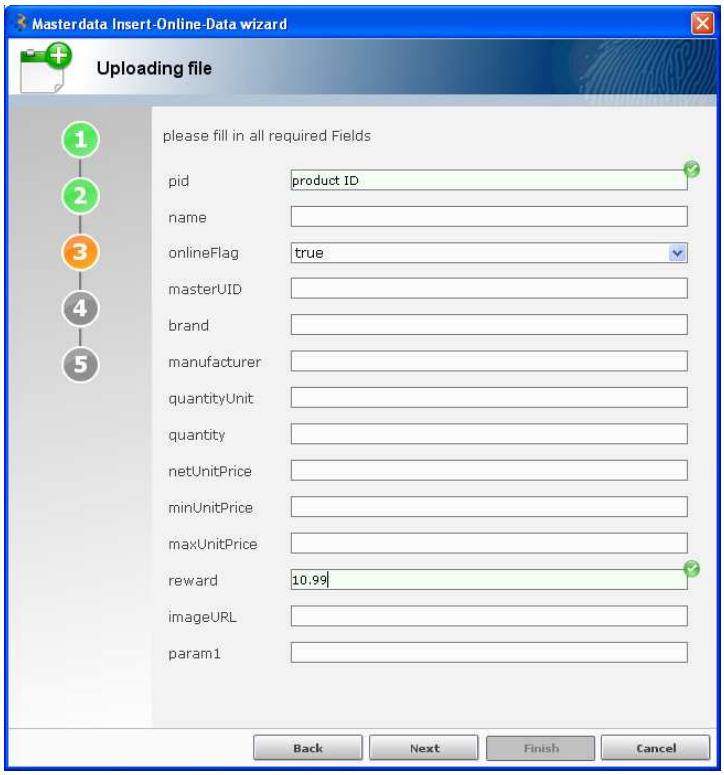


Figure 19: insert single object

5.6.2 Option 2: Data transfer with Kettle scripts

ETL script

The required data files (items.csv, banners.csv, categories.csv, categoryhierarchy.csv, itemcategories.csv, transactions.csv and business rules) have to be copied into the directory /data/re_server/re_<RDE-ID>/upload/. The corresponding ETL script is uploaded in the menu ETL Scripts (Figure 20) by pressing the button „plus“.

Copy-from-Upload

The prudsys standard script “copy-from-upload” can be executed by pressing the button Execute. All data files which are within the upload directory will be automatically transferred into the standard data directory /data.

However, Kettle offers a huge variety of functionalities, such as the filtering, sorting and pre-formatting of CSV files. These actions can be combined in Kettle scripts.

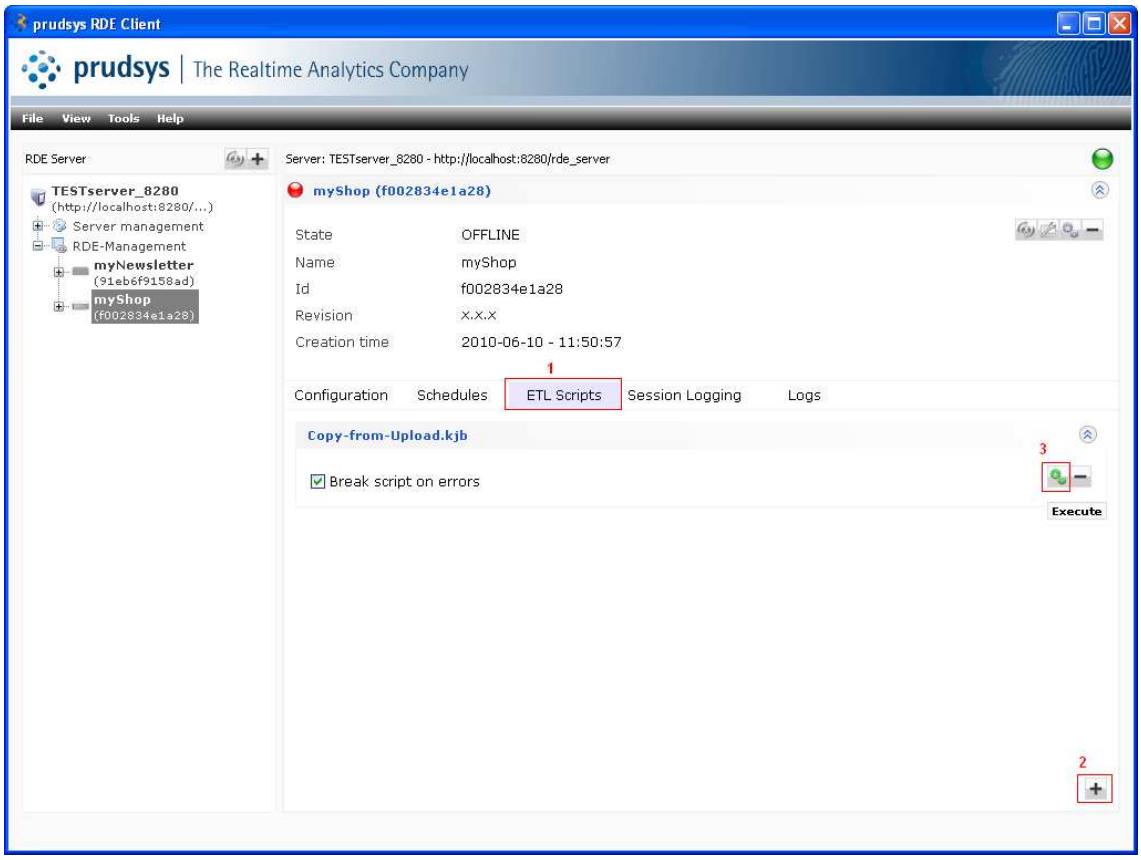


Figure 20: upload and execution of an ETL script

5.7 Server jobs (scheduled jobs)

The data upload processes are often daily and very time-consuming. Therefore, the RDE Servers features functionalities for the automated upload of product and catalogue data to the server.

Data migration processes

Therefore, functionalities from the ETL (Extract, Transform, Load) library Kettle (Data Integration Service) from the Business Intelligence Suite Pentaho have been integrated. Complex data migration processes can be configured with the corresponding client application („Spoon"), transferred to the RDE server and then executed.

In combination with an integrated scheduler, any data migration process, so called server jobs, can be started. The processes can be scheduled and will be executed automatically.

The following server jobs are available:

Data migration (1-3):

- automated execution of a Kettle script which has to be uploaded in the menu ETL Scripts (see also Chapter 5.6.2)
- parameter: name of the Kettle script
- not active per default (has to be activated)

NlStatSync:

- automated storage of newsletter statistic files
- not active per default
- described in detail within Chapter 7.10.1

Restart:

- deactivation and re-activation of the RDE application which will lead to the automated storage of the translog files
- not active per default

Stat:

- automated storage of the current RDE statistic files
- active per default: once per hour

UpdateTempSupport:

- algorithm for the decrease of the weight of older recommendation result rules to strengthen and fasten the impact of the current, most up to date result rules
- active per default: once per week, Monday at 0.15 am

There are only 3 steps to configure your own server job.

Step 1: Configure server job (1)

- job name: any name can be chosen
- parameters: if necessary (e.g. selection of a Kettle script)
- activate schedule: yes/no

Step 2: Configure server job (2)

- interval settings: every … second / minute / hour / day / week / month or once

Step 3: Configure server job (3)

- specific time settings, depending on the settings of step 2
- Time / day / month

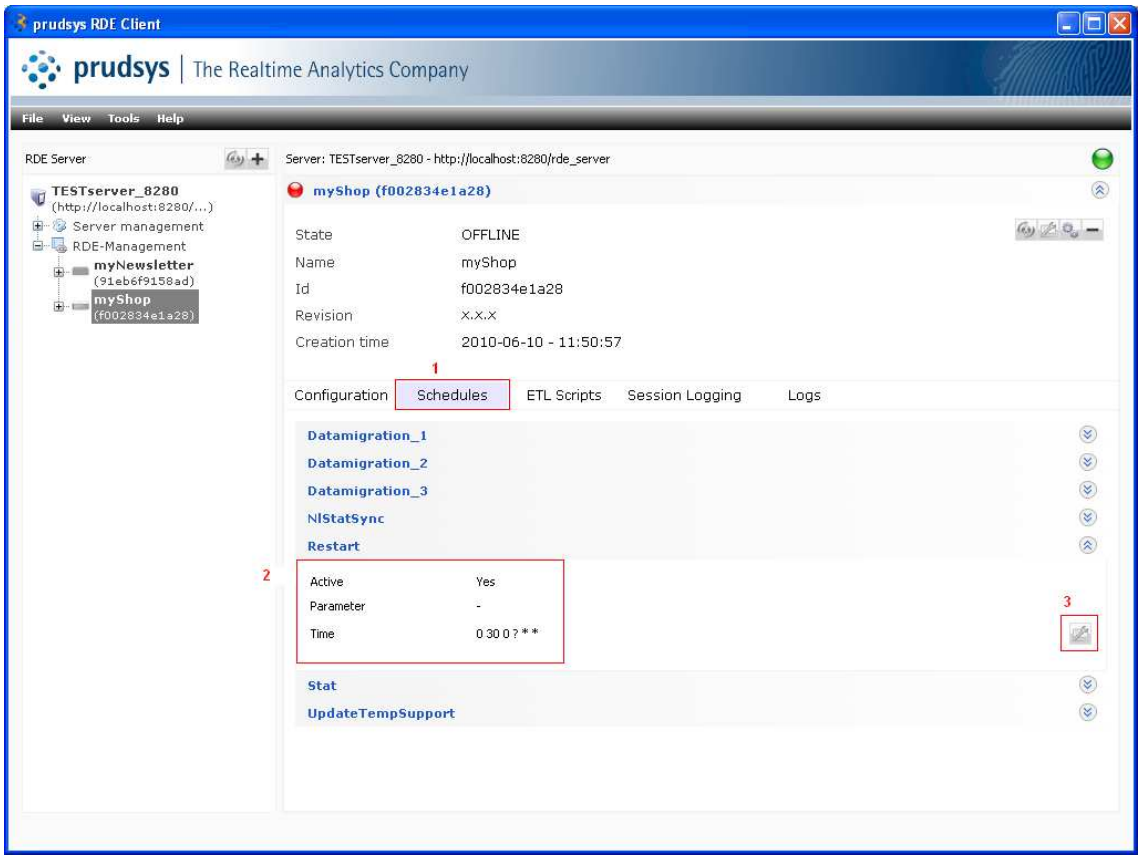


Figure 21: configuration of schedules

5.8 Setting of configuration parameters

All necessary configuration parameters for the operation of an online shop RDE as well as a newsletter RDE can be set comfortably using the menu “configuration” within the client. Customized properties can be added if required via the “plus” button (Figure 22-1).

Additionally, there is a possibility to export and import the configuration setting (Figure 22-2).

Activate settings

After changing existing or creating new settings, the configuration is saved automatically. To activate the settings, the RDE application has to be deactivated and activated again (therefore see Chapter 5.3).

To restore default parameters, click on the “arrow” button next to the “minus” button.

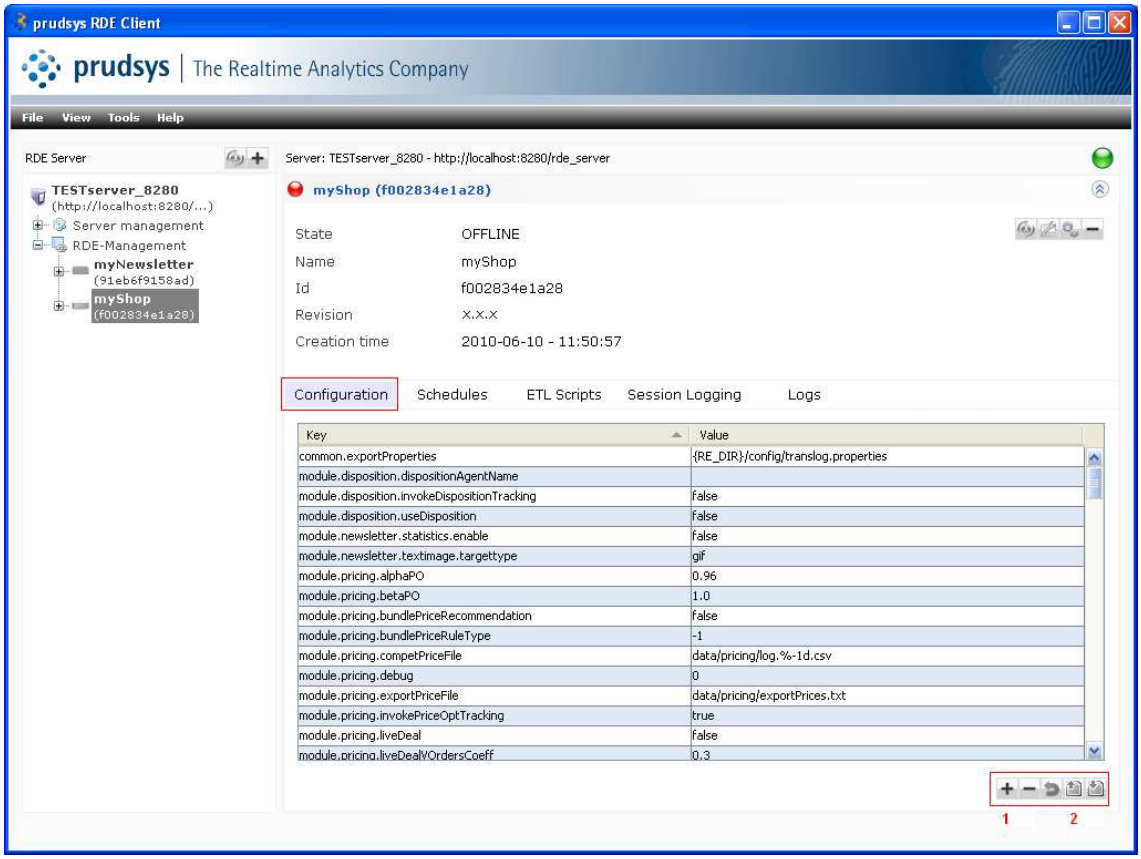


Figure 22: configuration - overview

In the following, all general parameters for RDE configuration will be presented and explained. Depending on the RDE modules there are additional parameters which will be explained in the corresponding chapters.

5.8.1 General RDE properties

RDE administration		
Property	Value	Description
re.alg.debuglevel	0 – disabled (default), 1 – default, 2 – extensive	Debug level
re.analysis.settings.usepreprocessing	true | false default: false	True: enable pre-processing (for more information see Chapter 5.11.1)
re.config.automatically	True | false default: true	False: file config_online.xml could be change manually True: file config_online.xml will be generated automatically and can not be changed manually
re.itemdata.columns.numeric	[param1..param20] default: <empty>	Comma separated list of columns from items.csv which should be read in as numeric values

Table 14: configuration parameters – RDE administration

Logging modes

The RDE logging is used to save information and error messages instead of displaying them within the standard display. The advantage lies in the possibility to set different logging modes which can be used to filter important messages.

- OFF: logging is deactivated
- FATAL: sever errors that cause premature termination
- ERROR: other runtime errors or unexpected conditions
- WARN: other runtime situations that are undesirable or unexpected, but not necessarily "wrong"
- INFO: interesting runtime events (startup/shutdown)
- DEBUG: detailed information (methods and parameters)
- TRACE: more detailed information
- ALL: all information

Hint: We recommend the log level INFO for live mode, because too detailed information could result in performance problems.

Changes can be made via curl or the RDE Client.

Changing the log level via curl

To change the log level, send the following URL e.g. with curl.

curl –X POST --digest –u username:password
http://host:port/rde_server/admin/logs/loglevel/<LEVEL>

Note: Users need permissions to set a new log level (therefore see Table 2).

Changing the log level via client

The display of the server logs can be activated within the Server Overview Menu. Therefore click the arrow button shown in Figure 23. Server log and log level turn up in the lower area. The current log level will be displayed. To change the level, select the desired one within the drop down menu.

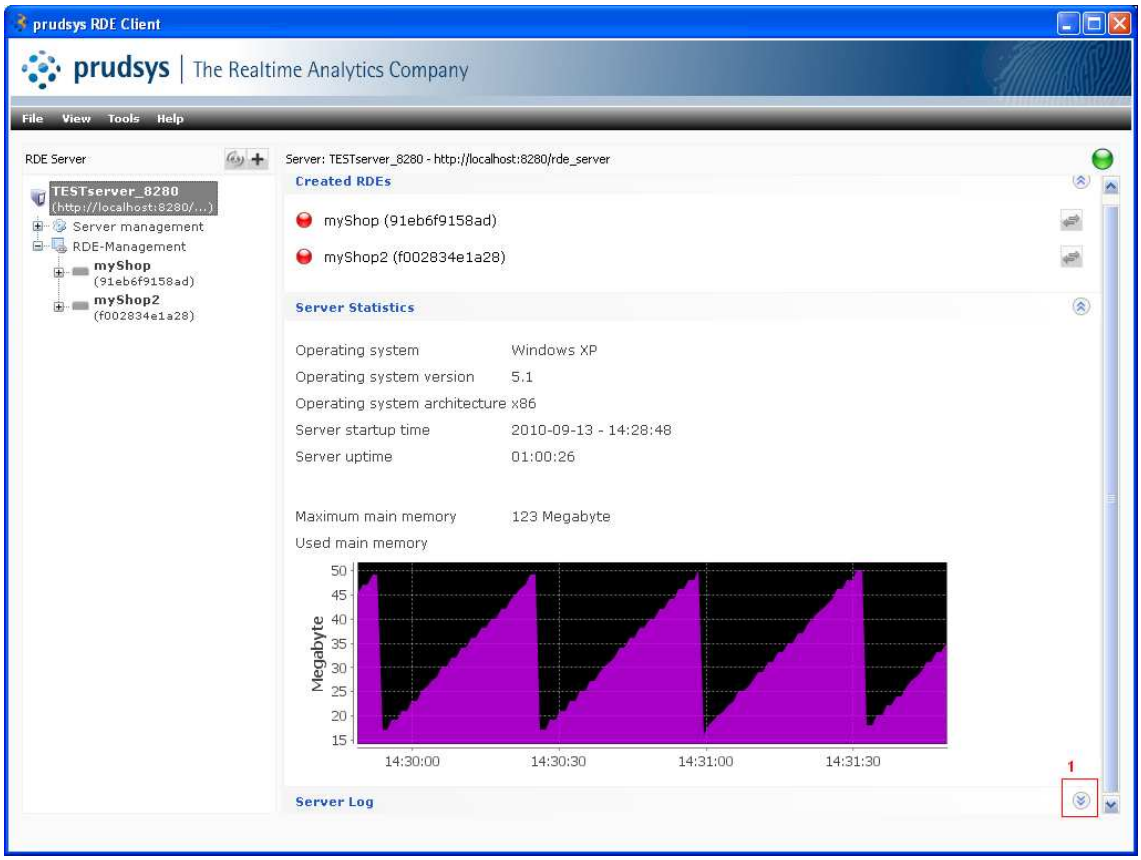


Figure 23: display of server log

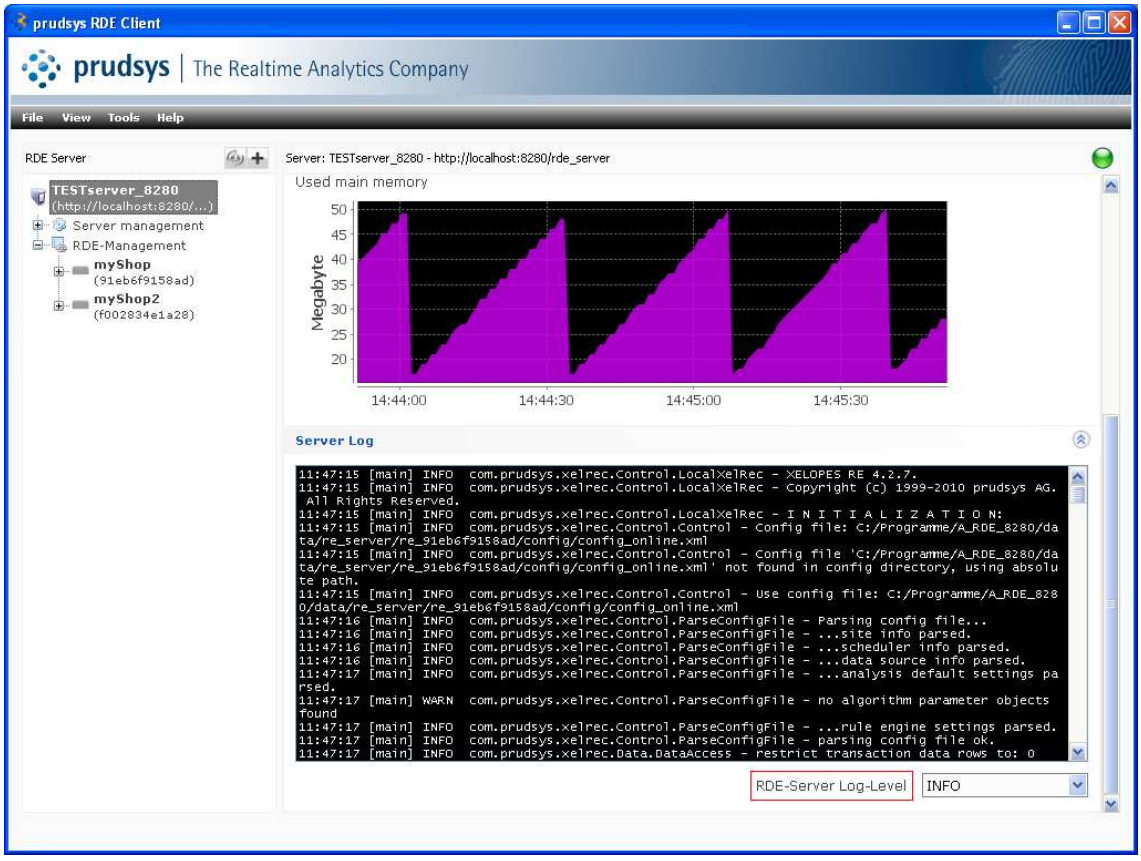


Figure 24: change server log level

Request current log level

To check the current log level, the following request can be sent via browser. Note: Users need permissions to check the log level (therefore see Table 2).

http://host:port/rde_server/admin/logs/loglevel

5.8.2 Fallback settings

Top seller / top categories		
Property	Value	Description
re.algorithm.topseller.items	true | false default: false	Automated calculation of top sellers / top categories for recommendation and newsletter templates
re.alg.precalcTopSellerNumber	[0..integer_max] default: 200	Number of top seller items which are available at runtime
re.alg.preCalcTopsellerType	0 – from rules (default), 1 – from transaction data, 2 – from item rank attribute	Method for top seller / top category calculation
re.alg.topSellerSubType	0 – clicks, 1 – baskets, 2 – orders (default), 3 – revenue, 4 – sales number	Top sellers / top categories are based on column from transaction data, Note: required properties: re.alg.preCalcTopsellerType = 1 and re.algorithm.topseller.items = true
re.translog.readTranslogs	true | false default: false	True: take historic transaction log files into account
re.translog.readNLastDays	<amount> default: 100	Amount of historic transaction log files based on time stamp, if property readTranslogs = true
re.alg.readTranslogForTopsellersLastNDays	[-1..integer_max] default: -1	Number of days used for top seller calculation based on a limited amount of property readNLastDays (typical for late-breaking top sellers). Note: column timeID within the translog file is the decisive factor which could differ from the translog name. -1 – read all translogs

Table 15: configuration parameters - top seller / top category

Note: Top sellers are used as fall back if there are not enough recommendation rules. These fall backs only will be calculated with activating the parameter “re.algorithm.topseller.items“. In this case, the standard calculation rule “0” is used as per default. As those top sellers are calculated in real time, they are always up to date and therefore the most successful ones in most cases.

5.8.3 A/B Testing

A/B testing

The prudsys RDE features A/B testing functionalities, i.e. there is the possibility to test the effect to users (and the business success) towards recommendations compared to no recommendations or other recommendation types.

The shop visitors will be separated into two groups. The size of the groups can be defined individually, e.g. every 10th, but also every 2nd user can be assigned to the control group. Additionally, it is possible to assign a user manually to the control group.

If control groups are used, the statistic will be calculated separately for control group and optimised price group. So the differences can be shown directly.

A/B testing		
Property	Value	Description
re.statistics.controlgroup	true | false default: false	activate A/B tests
re.statistics.controlgroup.frequency	<frequency> default: 10	frequency: e.g. 10: assigns every 10th session to the control group Note: property re.statistics.controlgroup = true is required.
re.preprocessing.extractGroups	true | false default: false	True: Allocation of user-group will be extracted. Every user keeps his/her group for future sessions.
re.preprocessing.extractGroups.days	[0..integer_max] default: -1	Number of last days which should be read in to extract user-groups -1: all days
re.alg.useSameGroupForUser	true | false default: false	True: Allocate the same group to a user, if it exists after pre processing.

Table 16: configuration parameters - A/B testing

To check the frequency setting, simple recommendation requests can be sent. If the output format JSON is defined, the following example for control groups will be returned:

```
[
  {
    "global": {
      "controlgroup": "true"
    }
  }
]
```

5.8.4 Transaction logging

Property	Value	Description
re.alg.useSessionIdAsUserId	true | false default: false	true: session ID is used as user ID
re.translog	true | false default: false	transaction log files will be created
common.exportProperties	{RDE_DIR}/config/ translog.properties	pre-defined path for the storage of the transaction log files
translogFilePath	<path>	path for translog files, default empty (defined in common.exportProperties)
re.translog.readTranslogs	true | false default: false	for personalized recommendations (e.g. myShop or newsletter recommendations): take historic transaction log files into account
re.translog.readNLastDays	<amount> default: 100	amount of historic transaction log files based on time stamp, if property readTranslogs = true

Table 17: configuration parameters - transaction files

The storage of the so called translog files is necessary for:

- personalized recommendations
- top seller / top category calculation
- control of requests
- personalized recommendations for newsletters

Personalized recommendations are calculated for every single user based on his/her historic purchase and behavioural patterns. The transaction log files have to be activated because they record the purchases and behaviour of the shop visitors.

In order to take historic transaction log files into account (to get more result rules based on a broader data basis), this functionality has to be set explicitly and the number of historic transaction log files to be considered has to be defined.

As soon as the RDE is integrated into the online shop, it constantly learns new rules in real time on the basis of a constant interaction with the shop visitors. In order to learn new rules, the following data is tracked, analyzed and stored within the directory /data/re_server/re_<RDE-ID>/translog. It contains the following information:

Column	Value	Description
1	Date and time	Format: yyyy-mm-dd hh:mm:ss
2	User-ID	ID of the current visitor
3	Transact-ID	ID of the current transaction
4	Group	0: recommendation group 1: control group

Column	Value	Description
5	itemID	Product ID of the current product
6	transType	Transaction type: -2: search -1: click 0: basket 1: order
7	basket	Number of product <itemID> within the basket
8	order	Number of product <itemID> ordered
9	itemsAction	Products which already have been combined
10	Category path	Category path
11	Price	Item price
12	Channel-ID	Channel

Table 18: columns of translog file

Note: Translog files will only be exported after a restart of the RDE application.

5.8.5 Event logging

Besides the logging of transaction data, the RDE Server provides a method to log all recommendation and event requests with necessary information. The log file is stored within the directory re_server/re_<RDE-ID>/logs/eventlog and contains the following information:

Column	Parameter	Description
1	date	Timestamp and time zone
2	transactionID	Empty if no session-ID was set
3	userID	Empty if no user-ID was set
4	channel	Empty if no channel was set
5	group	0: recommendation group 1: control group
6	module	RE: Recommendations NL: Newsletter SC: Scoring PR: Pricing AP: Assortment Planning DC: Data Cleansing
7	requestType	Recomm, basket, order, newsletter (image URL, link URL, text image URL)
8	templateName	Template name or empty if no template was used
9	inputValues	Comma separated if there is more than one input, e.g. item-ID and amount

Column	Parameter	Description
10	inputKeys	Parameter names of all input values
11	outputValues	Comma separated if there is more than one output
12	outputKeys	Parameter names of all output values
13	Requested URL	URL

Table 19: columns of event log file

The event logging can be enabled within the logs area of the RDE Client.

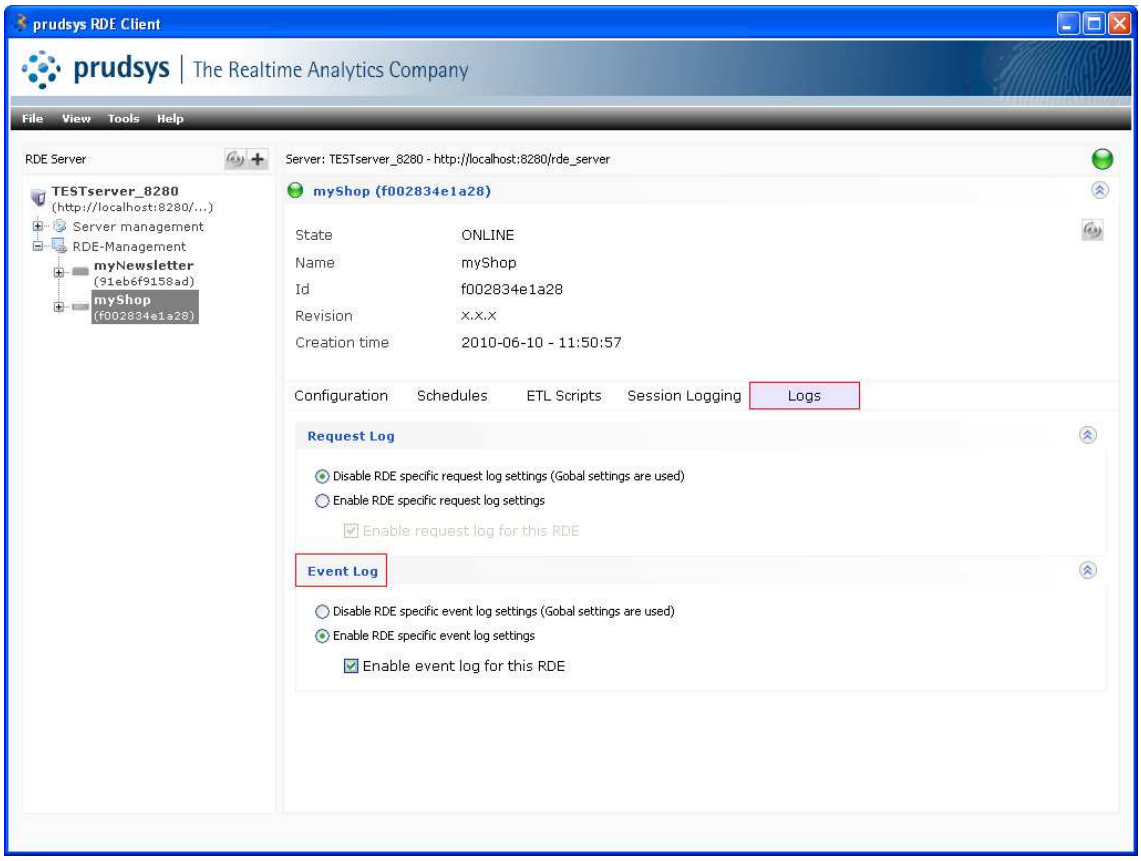


Figure 25: enable event logging

Enable event logs

Alternatively, the following POST request has to be sent e.g. with curl:

curl –X POST --digest –u username:password http://host:port/rde_server/admin/logs/loglevel/INFO/logger/com.prudsys.eventLogger

Disable event logs

To disable the event logger, the following POST request has to be sent e.g. with curl:

curl –X POST --digest –u username:password http://host:port/rde_server/admin/logs/loglevel/WARN/logger/com.prudsys.eventLogger

Get event logs

A complete list of all available event logs of a specific RDE can be requested via browser:

http://host:port/rde_server/admin/logs/eventlog/res/<RDE-ID>/list

To get only a single day, this request has to be sent via browser:

http://host:port/rde_server/admin/logs/eventlog/res/<RDE-ID>/day/<DAY>

Please note: The required format of <DAY> is YYYY-MM-DD.

Additionally, users can specify the number of latest entries by declaring the parameter "last":

http://host:port/rde_server/admin/logs/eventlog/res/<RDE-ID>/day/<DAY>?last=<LAST_ENTRIES>

Delete event logs

To delete all event logs, the following DELETE request has to be sent e.g. with curl:

curl –X DELETE --digest –u username:password
http://host:port/rde_server/admin/logs/eventlog/res/<RDE-ID>/list

To delete an event log for only one specific day, the following DELETE request has to be sent e.g. with curl:

curl –X DELETE --digest –u username:password
http://host:port/rde_server/admin/logs/eventlog/res/<RDE-ID>/day/<DAY>

Please note: Users need permissions to get or delete event logs (therefore see Table 2).

5.8.6 Offline learning properties

From the first start of the prudsys RDE, recommendations can be calculated and displayed if there are historical transaction data files. Through a pre-processing algorithm (the so called offline learning), the RDE creates a valid result rules file, which is the basis for recommendations. This means, no learning phase is required and the engine generates results from the first day on.

The offline learning process is affected by the following properties:

Offline Learning		
Property	Value	Description
re.termination.criterion.numberofreco	<amount>, default: 5	Maximum number of rules for each product (over all calculation rules)
re.alg.flatrulesdecomposition.maxitemsize	<amount>, default: 5	Products-to-product rules: maximum number of recommendations per product, weight 1.0
re.alg.flatrulesdecomposition. minimumsupportcount	<amount>, default: 2	Expert setting, threshold for learning rules (offline type 1)
re.alg.collaborativefilteringdecomposition. maxitemsize	<amount>, default: 3	Category-to-product rules: maximum number of recommendations per product, weight 3.0
re.alg.collaborativefilteringdecomposition. minimumsupportcount	<amount>, default: 3	Expert setting, default: 3, threshold for learning rules (offline type 3)
re.alg.useCategoriesToFillRules	true | false default: false	False: offline rules of type 4 will be ignored (category top seller)

Table 20: configuration parameters - offline learning

Offline learning types:

- 1 – product to product learning rule (flatrules)
- 2 – actual not in use
- 3 – category-based product to product learning rule
- 4 – category top seller will be filled up if there are not enough results

These settings define the amount of rules which will be calculated through the offline learning process.

Note: The “maxitemsize” – amounts should be the same for all algorithms so that always the same <amount> of recommendations per product will be created.

5.9 Offline Learning

First initial learning

If there are historic transaction data files (transactions.csv), these files can be used to calculate recommendation rules, for example for an initial rules feed at the beginning of a project (when there are still no rules from the real time learning). To calculate offline rules, the RDE application has to be deactivated first. Then, the offline learning is started through right-click on the RDE application and selection of the corresponding option from the context menu. After the re-activation of the RDE application, the file resultrules.csv within the directory re_server/re_<RDE-ID>/data/ will contain recommendation rules. Those rules can be used to display recommendations.

Note: The process offline learning overwrites the complete file resultrules.csv. All previous results will be lost.

5.10 Recommendation templates

In order to obtain results from the RDE application, so called recommendation templates have to be created. Additionally, these templates can be called up using an HTTP GET request to the REST interface of the RDE Server.

A recommendation template contains all necessary information needed for the generation of recommendations. The RDE Server is able to parse this information and send corresponding queries to the recommendation engine, as well as to process the recommendation returns and deliver it to the sender.

For every recommendation engine, an unlimited number of templates can be created.

Template wizard

As the recommendation templates are rather complex and their manual creation requires specific knowledge, a recommendation template wizard has been developed to support the creation of any number of complex and valid recommendation templates (see Figure 26).

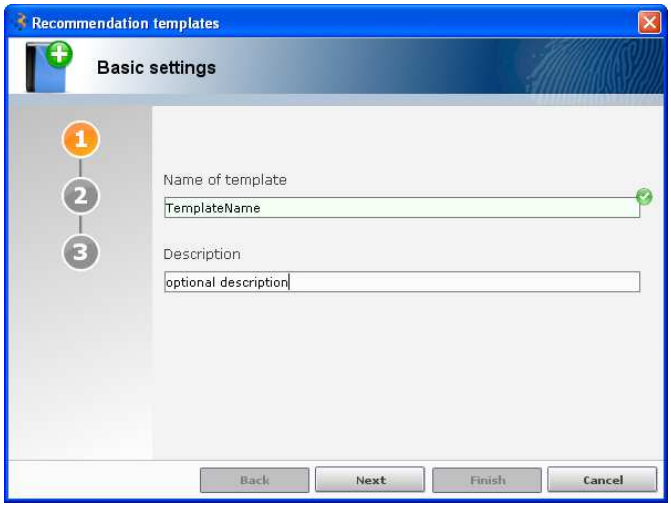


Figure 26: template wizard

With the recommendation template wizard, any type of recommendation template with a great number of recommendation functionalities can be created in only 3 steps.

5.10.1 Basic settings

- Template name and description:

Any name can be chosen. Based on the template name, the recommendation engine is queried via HTTP request.

5.10.2 Module specific settings

Depending on the RDE modules, the second step differs in the setting options and will be explained within the corresponding chapters:

- Part RDE | Recommendations: output settings for recommendation templates (Chapter 6.3.2)
- Part RDE | Newsletter: general settings for newsletter recommendations (Chapter 7.7.2)

5.10.3 Add queries – overview

In the first two steps, the basic settings have been set. The next step is the most important one because now the recommendation requests will be created (Figure 27).

Combined requests

The existing basic recommendation types can be combined within one template, for example top sellers and personalized recommendations. Duplicates that might occur due to the combination of several requests will be filtered in advance by the system to avoid the display of two identical products. Combined requests are integrated into only one coherent HTML request which is integrated into the web site.

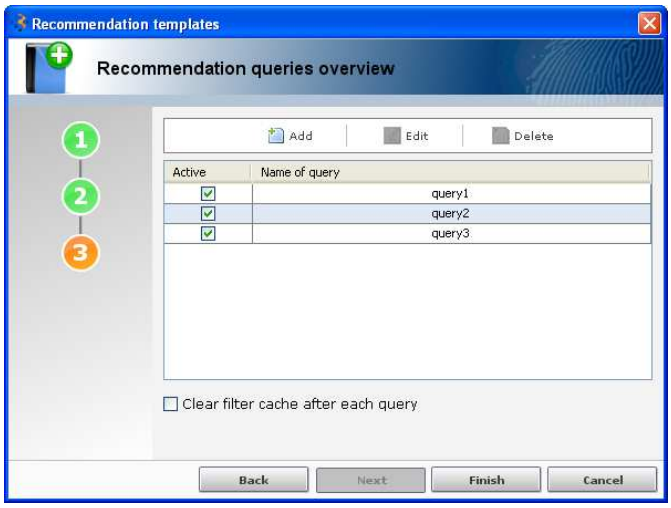


Figure 27: list of recommendation queries within one template

The following basic recommendation types are available:

RDE Module	Premise (Input)	Conclusion (Output)	Remarks	Required configuration parameter
RE, NL	product	product	typical cross and up selling, e.g. on product detail pages	default configuration, apart from top seller fallback is desired: see Table 15
RE	search term	product	within search result or landing pages, is referenced by a search engine	default configuration
RE	category	product	product recommendations based on category relations (“Customers browsing within trousers ware especially interested in the following products…”)	default configuration
RE, NL	basket (product list)	product	product recommendations based on several products	default configuration, apart from top seller fallback is desired: see Table 15
RE, NL	customer (UserID)	product	product recommendations based on the historic and current behavioural patterns of a specific user	obligatory: re.translog = true re.translog.readTranslogs = true optional: see Table 17 top seller fallback is desired: see Table 15
RE, NL	category	top seller	dynamically calculated top sellers from a specific category	obligatory: re.algorithm.topseller.items = true optional: see Table 15
RE	category	sorted product list	product list based on: - personal recommendations - category top seller - random products from category	case: - personal recommendations using translog information: see Table 17 - category top seller: see Table 15
RE, NL	no input	global top seller	dynamically calculated top sellers from the complete catalogue	obligatory: re.algorithm.topseller.items = true optional: see Table 15
RE	category	banner	banner recommendations based on categories	obligatory: re.recommendations.banner = true optional: see Table 32
RE	customer	banner	personal banner recommendations based on last visited category	obligatory: re.recommendations.banner = true optional: see Table 32
RE	no input	global top category	dynamically calculated top categories from the complete catalogue	obligatory: re.algorithm.topseller.items = true optional: see Table 15

RDE Module	Premise (Input)	Conclusion (Output)	Remarks	Required configuration parameter
RE	category	top category	dynamically calculated top categories from a specific level of the catalogue	obligatory: re.algorithm.topseller.items = true optional: see Table 15
RE, NL	customer	products from transaction history	The products are based on selectable history types: - clicked product, - product added to basket, - ordered product, - product added to basket but not ordered, - clicked but not ordered product	obligatory: re.translog = true re.translog.readTranslogs = true optional: see Table 15

Table 21: basic recommendation templates

The following short cuts are in use:

RDE | Recommendations	RE
RDE | Newsletter	NL
RDE | Pricing	Pricing
RDE | Data Cleansing	DC
RDE | Scoring	SC
RDE | Assortment Planning	AP

Table 22: RDE modules short cuts

The recommendation origin defines the type of input data for the calculation of recommendations. The following types exist:

- product
- category
- search term

Depending on the chosen type and the template definitions, the RDE calculates recommendations on the basis of product IDs, category IDs or even search terms.

Additionally, the following types exist

- customer (session)

In this case, personalized recommendations based on historic and the current user's session (stored in the translog file) are generated. To personalize recommendations, a session has to be assigned to the user via request.

- product list

If this configuration is chosen, a recommendation will be calculated on the basis of several products.

Most important types

The most important recommendation types are described in the following chapters:

- Product to product recommendation (5.10.4)
- Basket to product recommendation (5.10.5)
- Search term to product recommendation (5.10.6)
- Customer recommendation (5.10.7)
- Global top seller and category top seller (5.10.8)
- Global top category (5.10.9)
- Category to top category (0)
- Customer transactions (5.10.11)
- Category to sorted product list (5.10.12)

5.10.4 Add queries – product to product recommendation

Step 1: Basic settings

- Name of the query: is used to differentiate the queries if there are several queries within one template
- Maximal number of recommendations: set the number of recommendations which will be displayed within the templates if they exist.
- Recommendation type: define which kind of recommendation will be created, i.e. select “Product to Product” in that case.

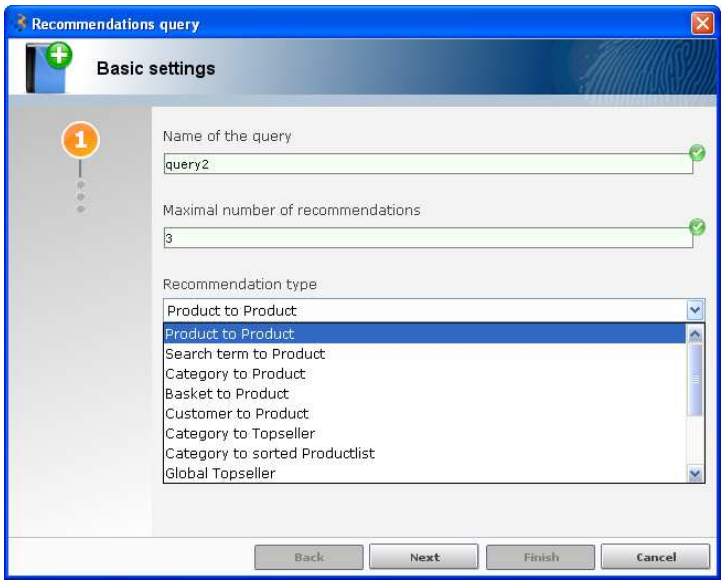


Figure 28: recommendation query - basic settings

Step 2: Recommendation return value

In case of recommendation templates, there is a possibility to configure the return of product attributes for each sub query using the output setting JSON (Chapter 6.3.2).

To choose the desired return values, mark the option and select the desired parameter (hold “Strg” for more than one).

Figure 29 shows an example with the return of four product attributes. Additionally, the fields “product number” (product ID) and “reason” are always part of the JSON answer. The field reason provides information about why a product is recommended and contains the related product or e.g. “topseller”.

Please note: This option is not available for newsletter templates.

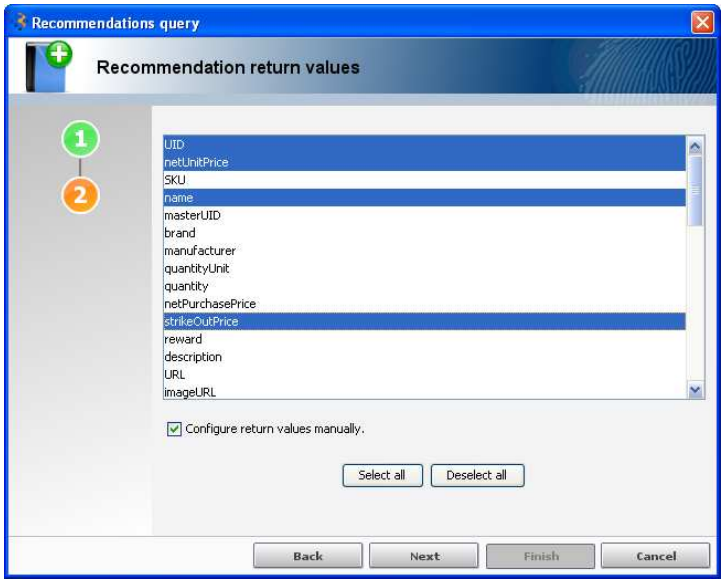


Figure 29: manual recommendation return value

Step 3: Expert settings

- Name of query parameter: input parameters which have to be transferred during the request for a recommendation: product (item), category (cat) or search term (search). The names of the queries can be set at will.
- Name of query filter parameter: this parameter is used as a dynamic attribute filter. Dynamic attribute filters can simply be attached with “&” to the template URL only if necessary. Arithmetic operators can be “=”, “!=”, “<” , “>”, “<=” or “>=”.
  For definition of static attribute filters see Step 5: Attribute filter settings

example:

http://host:port/rde_server/res/<RDE-ID>/recomm/<NAME-OF-TEMPLATE>
/sid/<SESSION-ID>?item=<ITEM-ID>
&<FILTER-PARAMETER>=
(conc.item.<attribute_name>=<value>)and(conc.item.<attribute_name>=<value>)

- Core name of item parameters: global item parameters that are transferred independently from the recommendation request, for example the name of an image directory. The RDE Server ignores those parameters and simply passes them on. They can be attached by "&<global_parameter><nr>" (nr has to be 1, 2 or 3).

example:

http://host:port/rde_server/res/<RDE-ID>/recomm/<NAME-OF-TEMPLATE>
*/sid/<SESSION-ID>?item=<ITEM-ID>&**<global_parameter><1|2|3>***

- Rule type: The rule type indicates the column "weight" within the file resultrules.csv. Default value is "0" which means the recommendations will be based on online rules. If you want the system to display manually set rules, please indicate here the type you have previously set for those rules (within column "weight"). Additionally, please mark „not learning" for manually added rules.

Please note: the following table explains the existing rule types:

Rule type	Learning type	Description
0	Online learning	Realtime based online rules
1	Offline learning	product to product learning rule (flatrules)
2		actual not in use
3	Offline learning	category-based product to product learning rule
4	Offline learning	category top seller will be filled up if there are not enough results

Table 23: online and offline rule types

- Learning or not learning: default setting is "learning". This configuration makes sure that the RDE always learns in real time on the basis of the customer behaviour. The setting can be chosen for product-to-product recommendations within RDE | Recommendations module only.

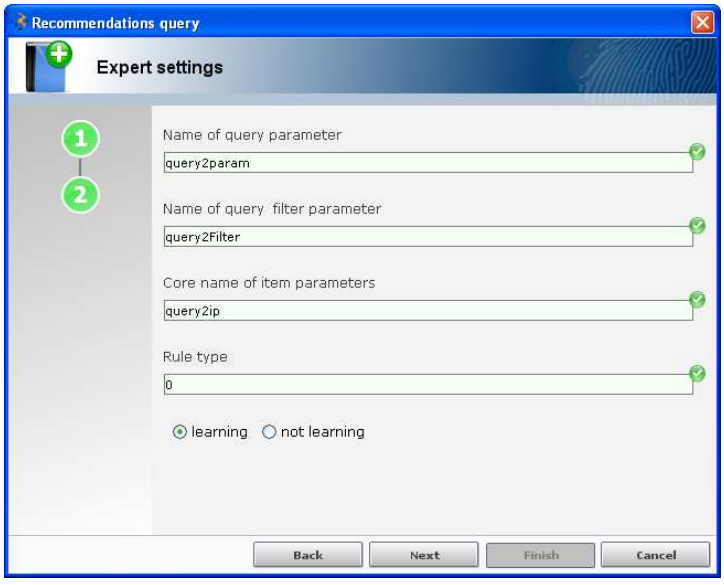


Figure 30: recommendation query - expert settings

Step 4: Fallback settings

Additionally, it is possible to configure fallback rules which will be displayed if not enough recommendations can be calculated. The fallbacks are either from the complete product range or only from one category. In this case, the category has to be committed using the parameter name defined in Fallback setting (see Figure 31). Duplicates will be removed in advance.

Please note: In order to display global top seller or category top seller, at least the following configuration has to be set within the "configuration" area (therefore see Figure 22):

re.algorithm.topseller.items = true

Optional top seller settings can be done to define top seller calculation mode or time (see Table 15).

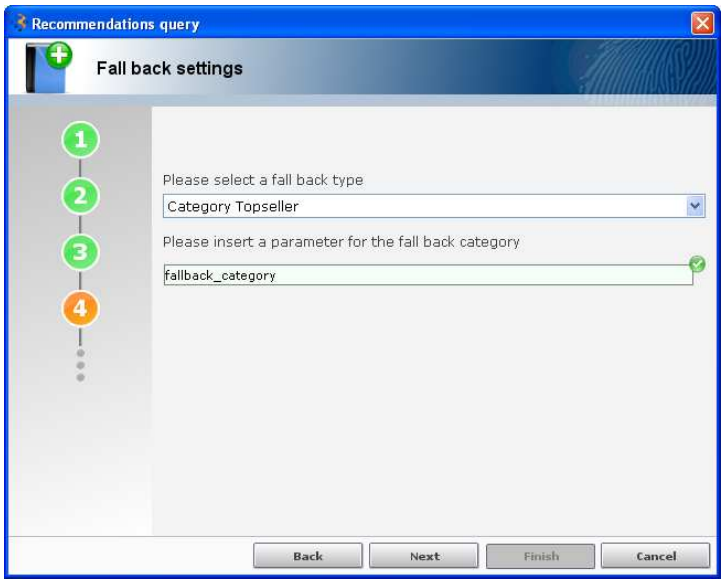


Figure 31: parameter name for category fallback

Step 5: Attribute filter settings

Depending on the chosen recommendation type and origin, in a next step the recommendations can be customized on the basis of so called business rules.

There are three different types of business rules, the "premise-to-conclusion filter rules" and the "conclusion filter rules" for either only one value or a comma separated list of values. Figure 32 shows examples of all types.

Premise-to-conclusion filter

The "premise-to-conclusion" filter rules are used to set up filters for the relation between an input product and the corresponding output products, for example the rule that the price of the recommended products always has to be higher than the price of the premise product.

Conclusion filter

The "conclusion filter", in contrary, only has an impact on the resulting recommendations. Figure 32 depicts, for example, how product recommendations can be limited (excluded) on the basis of categories. According to the settings in the figure, products from the category Cat1 will be excluded from the recommendations. This rule applies to all recommendations, no matter what the input information (premise) is.

There are several types of business rules, e.g. filters on the basis of categories, products, prices or other product attributes like colours. All filters can be initialized with fix values and can be linked through the following operators:

- Logical operators: and, or, xor (case sensitive)
- Compare operators: =, !=
- Numeric compare operators: <=, >=, <, >
- List operators: in, not in (e.g.: (conc.item.param1 not in (blue, red, yellow))

Please note: Numeric compare operators can only be used with numeric columns within items.csv. Possible columns are param1 to param20. Therefore, the columns have to be defined as numeric within configuration menu (property re.itemdata.columns.numeric, Table 14). Additionally, numeric operators will not have any effect on categories.

The RDE Client verifies the input for syntax but not for semantic errors (e.g. exclusion of conditions). Incorrect conditions will be ignored during execution.

Please note that only the concatenation of exactly two conditions is possible. A condition itself could consist of two filter rules. With it, any nesting could be created with a proper handling of parenthesis.

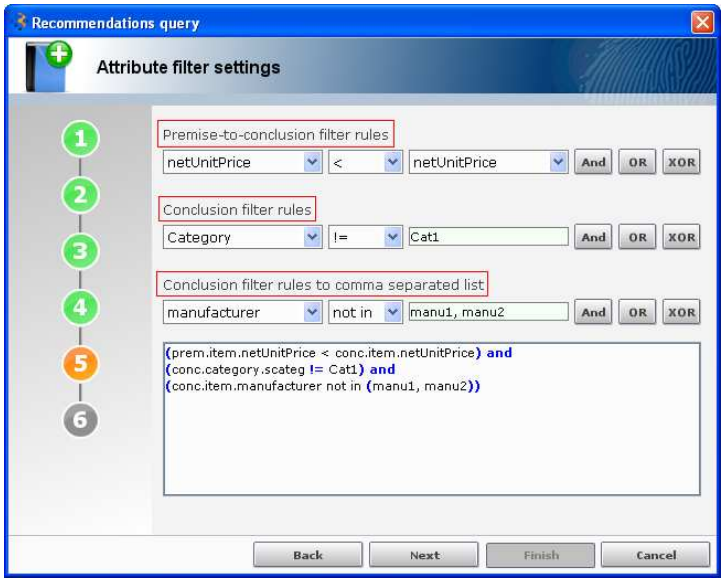


Figure 32: attribute filter settings

examples:

- Product conditions

((prem.item.netUnitPrice < conc.item.netUnitPrice) and
(conc.item.manufacturer not in (manufacturer1, manufacturer2)))

Recommended products have to be costlier than the input product and must not be from manufacturer 1 or manufacturer 2.

- Category conditions

((prem.category.scateg != conc.category.scateg) and (conc.category.uid != Cat1))

Recommended products have to be from other categories than the input product and must not from category Cat1.

Please note: There are some important differences between conclusion filter rules “category” and “target category”:

- Category: recommendations are only from the committed category
- Target category: recommendations are from the committed category and all sub categories (within the category hierarchy)

Step 6: Product filter settings

These settings are used to create a so called black list. There are two possibilities to define the black list: dynamic and static filters.

Dynamic product filter

Dynamic product filters can simply be attached with “&” to the template URL only if necessary.

http://host:port/rde_server/res/<RDE-ID>/recomm/<NAME-OF-TEMPLATE>/sid/<SESSION-ID>?item=<ITEM-ID>
&<DYNAMIC-FILTER-NAME>=<ITEM-ID_1>,<ITEM-ID_2>,<ITEM-ID_N>

Static product filter

Within the static product filter list the products will be completely excluded from the recommendations. Therefore, it is necessary to type all single product IDs, which then will be added or removed from the black list with the buttons “plus” (Figure 33-1) and “minus”.

In order to add a complete black list, there is a possibility to upload a text file with all items that should be excluded (Figure 33-2). This file should have only one column without header or separator.

example:
item1
item2
…

Please note: Dynamic product filters are stronger than static filters. This means the static black list will be ignored as soon as a dynamic filter exists.

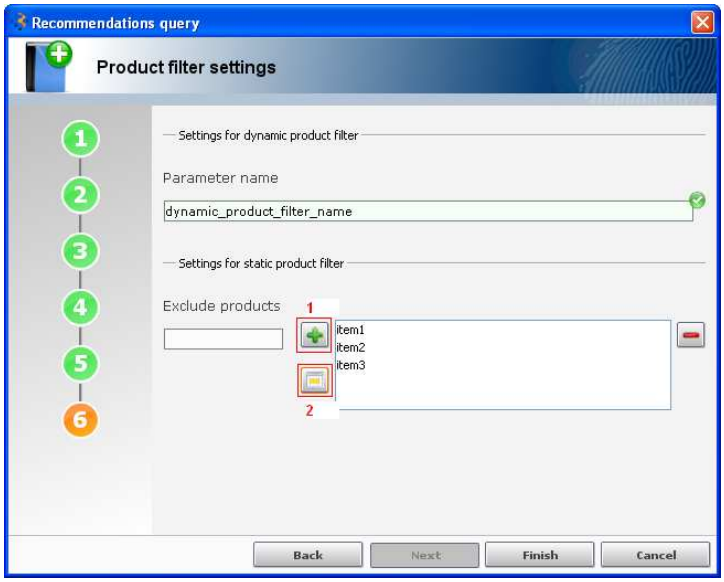


Figure 33: product filter settings

Result: Template URLs

After the successful saving of the recommendation templates, the corresponding recommendation functionality is available immediately. Provided that there are recommendation rules and product master data, recommendations can be called from the recommendation engine using HTTP requests. Differences between templates for RDE | Recommendations module (Chapter 6.3.3) and RDE | Newsletter module (Chapter 7.7.3) are described within the specific chapters.

Additionally, all existing templates can be exported and imported.

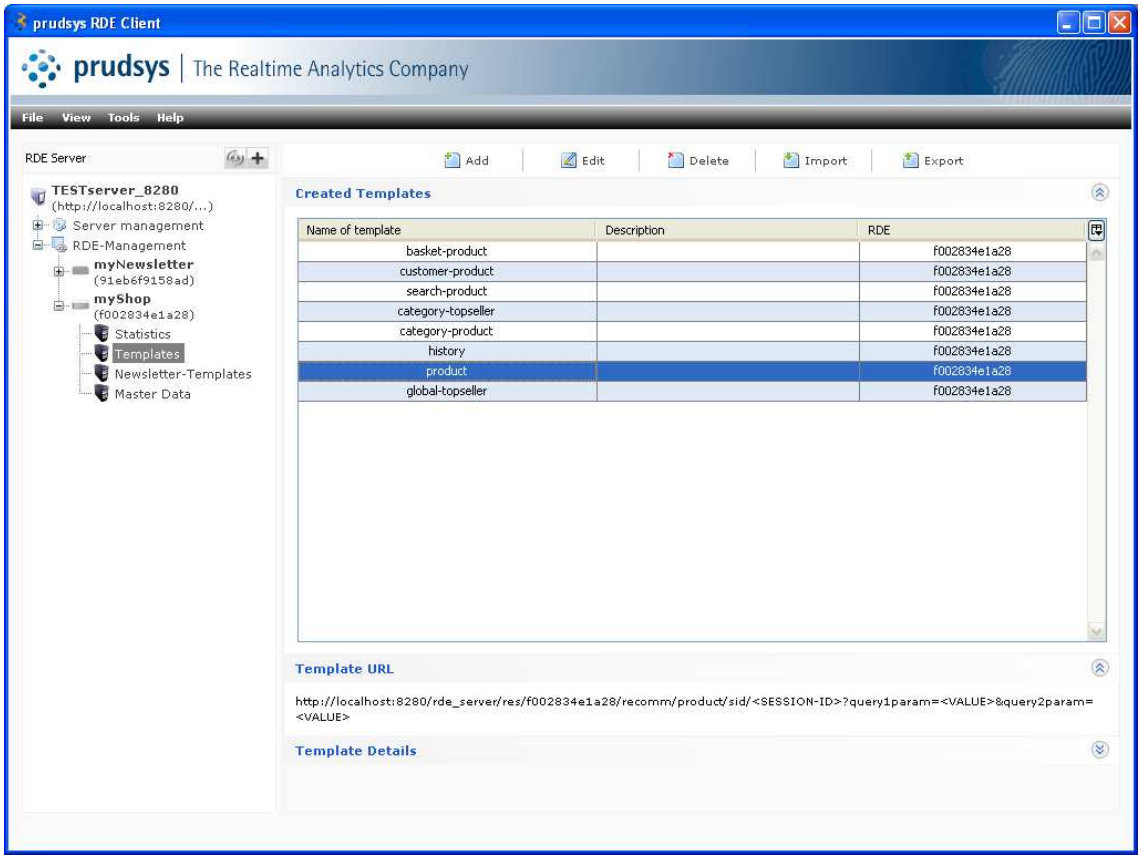


Figure 34: template overview

5.10.5 Add queries – basket to product recommendation

This template provides the same setting options like the product to product recommendation template. For more information see the previous Chapter 5.10.4:

- basic settings (page 76)
- recommendation return value (page 77)
- expert settings (page 77)
- fallback (page 79)
- attribute filter settings (page 79)
- product filter settings (page 81)

5.10.6 Add queries – search term to product recommendation

This template provides the same setting options like the product to product recommendation template without fallback options. For more information see the previous Chapter 5.10.4:

- Step 1: basic settings (page 76)
- Step 2: recommendation return value (page 77)
- Step 3: expert settings (page 77)
- Step 4: attribute filter settings (page 79)
- Step 5: product filter settings (page 81)

5.10.7 Add queries – customer recommendation

This template provides the same setting options like the product to product recommendation template. For more information see the previous Chapter 5.10.4:

- Step 1: basic settings (page 76)
- Step 2: recommendation return value (page 77)
- Step 3: expert settings (page 77)
- Step 5: fallback (page 79)
- Step 6: attribute filter settings (page 79)
- Step 7: product filter settings (page 81)

Only step 4 differs within template and will be described in the following section.

Step 4: Expert settings for user recommendations

The RDE Server provides a possibility to define extended user recommendations. Simple user recommendations ignore the following settings.

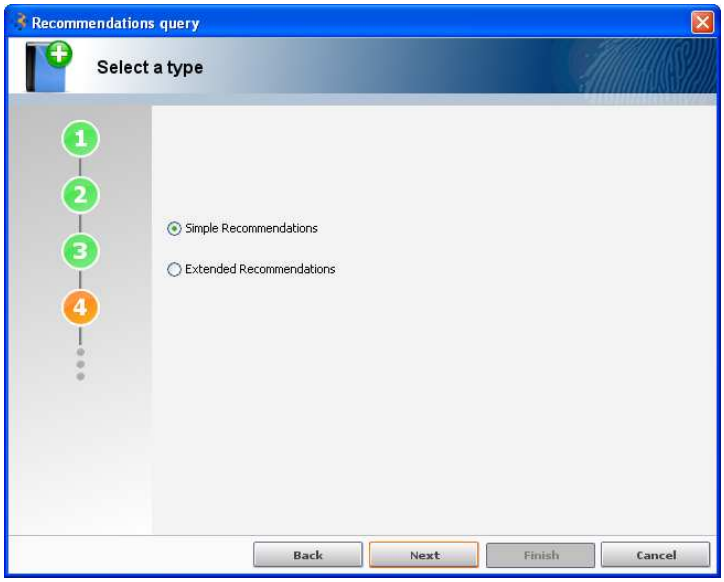


Figure 35: extended user recommendations (1)

After choosing the extended recommendations mode, four different options are able to be selected: clicks, baskets, orders and/or searches. The recommendations will be calculated based on the selections in the given sequence. This sequence can be changed by drag and drop the panel to the desired position.

Extended user recomm differences

Please note: Up to RDE Server version 2.11.3, the quantity of all single extended recommendations defines the number of recommendations which will be displayed within the template. This option overrides the setting “maximal number of recommendations” within basic settings (page 76).

Within newer versions, the setting “maximal number of recommendations” defines the recommendation limit overall, i.e. this limit overrides the total number of all extended recommendation quantities. Means each single quantity of extended type will be limited to that “maximal number of recommendations”. All in all the extended quantities could be set higher than “maximal number of recommendations” but the maximal number displayed will be according to basic settings. The special case “maximal number = -1” continue along the behaviour of versions up to 2.11.3.

The following example shows three user recommendations. The first two recommendations are based on the last five orders of the specific user. The last recommendation is based on the last three clicks.

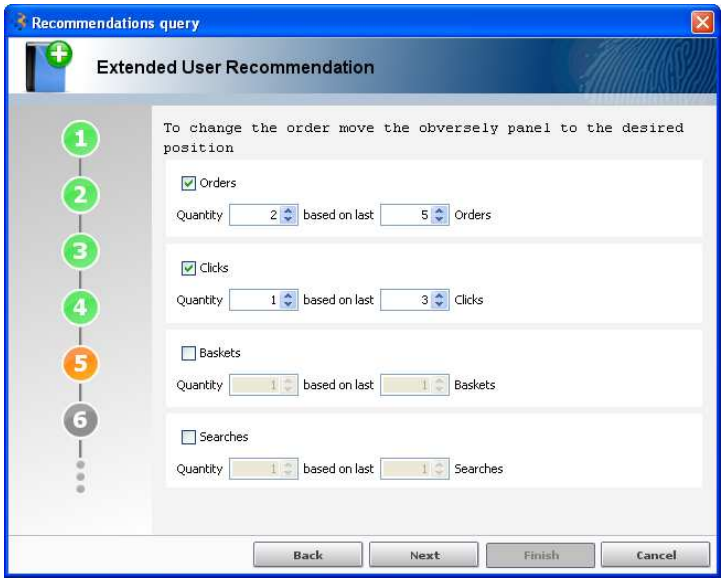


Figure 36: extended user recommendation (2)

The extended user recommendation output contains two new fields.

The field “basedOn” describes on what action the recommendation is based on:

- c – clicks
- s – searches
- b – baskets
- o – orders

The field “reason” shows why the product was recommended. It will be filled with one or more product-IDs or “topseller”.

Example output of an extended user recommendation:


```
{
    "netUnitPrice": "30.99",
    "onlineFlag": "1.0",
    "reward": "30.99",
    "reason": "topseller",
    "param1": "white",
    "product_nr": "p2k1",
    "basedOn": "o",
    "quantityUnit": "quantity",
    "UID": "p2k1",
    "manufacturer": "manufacturerA",
    "name": "cupboard",
    "brand": "brandB",
    "masterUID": "",
    "quantity": "100.0",
    "imageURL": "image"
 }
```

Please note: User recommendations are based on the historic and current user behaviour. Therefore, the following configuration has to be set within the "configuration" area (therefore see Figure 22):

re.translog = true

re.translog.readTranslogs = true

5.10.8 Add queries – global top seller and category top seller

This template provides the same setting options like the product to product recommendation template without fallback options. For more information see the previous Chapter 5.10.4:

- Step 1: basic settings (page 76)
- Step 2: recommendation return value (page 77)
- Step 3: expert settings (page 77)
- Step 4: attribute filter settings (page 79)
- Step 5: product filter settings (page 81)

Please note: In order to display global top seller or category top seller, at least the following configuration has to be set within the "configuration" area (therefore see Figure 22):

re.algorithm.topseller.items = true

Optional top seller settings can be done to define top seller calculation mode or time (see Table 15).

5.10.9 Add queries – global top category

This template only provides the basic setting options (page 76).

Please note: In order to display global top seller or category top seller, at least the following configuration has to be set within the "configuration" area (therefore see Figure 22):

re.algorithm.topseller.items = true

Optional top seller settings can be done to define top seller calculation mode or time (see Table 15).

5.10.10 Add queries – category to top category

This template provides the basic setting options (Step 1: page 76) and special expert settings:

Step 2: expert settings

- Name of query parameter: input parameter which has to be transferred during the request for a recommendation (in this case: category). The name of the queries can be set at will.
- Hierarchy level: The hierarchy level has to be set for these top categories, which should be calculated for only one part of the hierarchy tree (defined within categoryhierarchy.csv, Table 6). Level 1 means one level under the passed category.

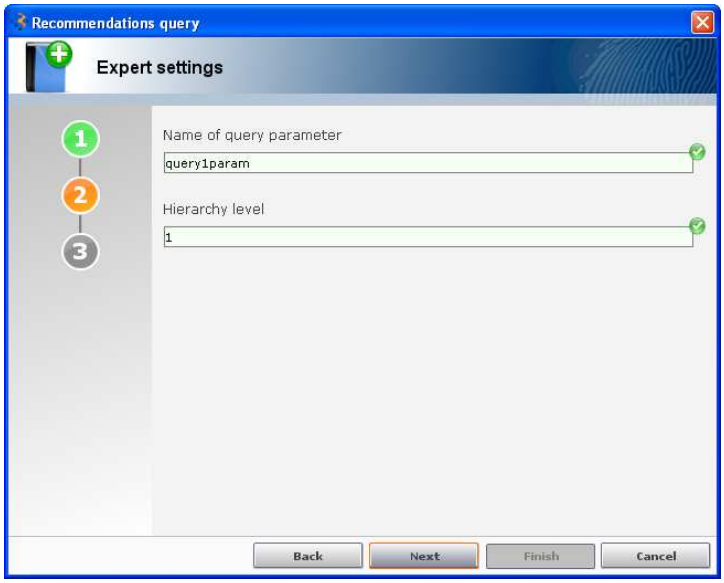


Figure 37: recommendation query - expert settings

Please note: In order to display top categories, the following configuration has to be set within the “configuration” area (therefore see Figure 22):

re.algorithm.topseller.items = true

5.10.11 Add queries – Transaction history

This template provides the basic and recommendation return value setting options like the product to product recommendation template. For more information see the previous Chapter 5.10.4:

- Step 1: basic settings (page 76)
- Step 2: recommendation return value (page 77)

Step 3: select user history

This template provides products from a user transaction history based on selectable types (see Figure 38).

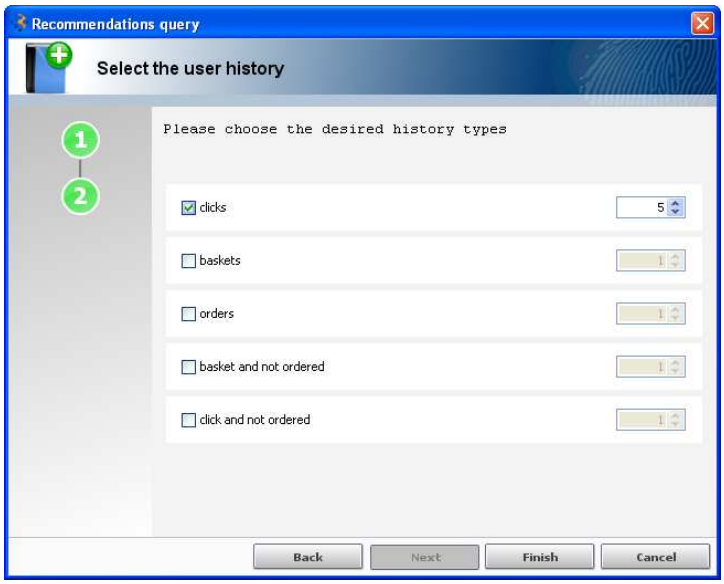


Figure 38: template transaction history

Please note: The template only provides products based on the user transaction history. These products are no recommendations.

The amount of chosen history types overwrites the value of “maximal number of recommendations” from Step 1 (Basic settings).

These history types will be shown with the field “reason” within the output:

- clicked product (reason “clicked”)
- product added to the basket (reason “basket”)
- ordered product (reason “ordered”)
- product added to the basket but not ordered (reason “basketAndNotOrdered”)
- clicked but not ordered product (reason “clickedAndNotOrdered”)

The transaction history will be analysed in the following way:

- First: latest session information from the user
- Second: translog data from the user

Please note: In order to use transaction log data, the following configuration has to be set within the “configuration” area (therefore see Figure 22):

re.translog = true

re.translog.readTranslogs = true

5.10.12 Add queries – category to sorted product list

This template provides the basic and recommendation return value setting options like the product to product recommendation template. For more information see the previous Chapter 5.10.4.

- Step 1: basic settings (page 76)

Additionally, the number of recommendations per page has to be defined.

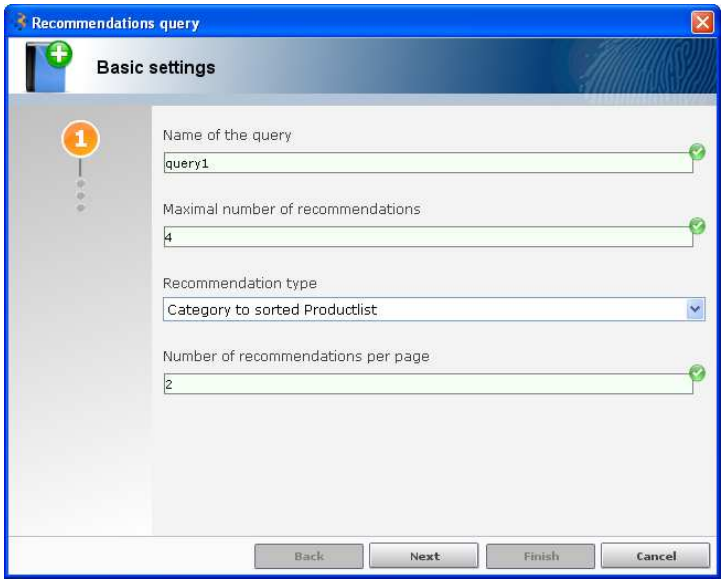


Figure 39: template category to sorted list

Please note: The options “maximal number of recommendations” and “number of recommendations per page” only have effects on recommendations if the resulting URL is modified in the following way:

- standard template URL:

http://host:port/rde_server/res/<RDE-ID>/recomm/<TEMPLATE_CATEGORY_TO_SORTLIST>/sid/<SESSION-ID>?category=<VALUE>

- extended template URL

*http://host:port/rde_server/res/<RDE-ID>/recomm/<TEMPLATE_CATEGORY_TO_SORTLIST>/sid/<SESSION-ID>?category=<VALUE>**&page=<NUMBER>***

- Step 2: recommendation return value (page 77)
- Step 3: expert settings (page 77)

Step 4: sort base of user transactions

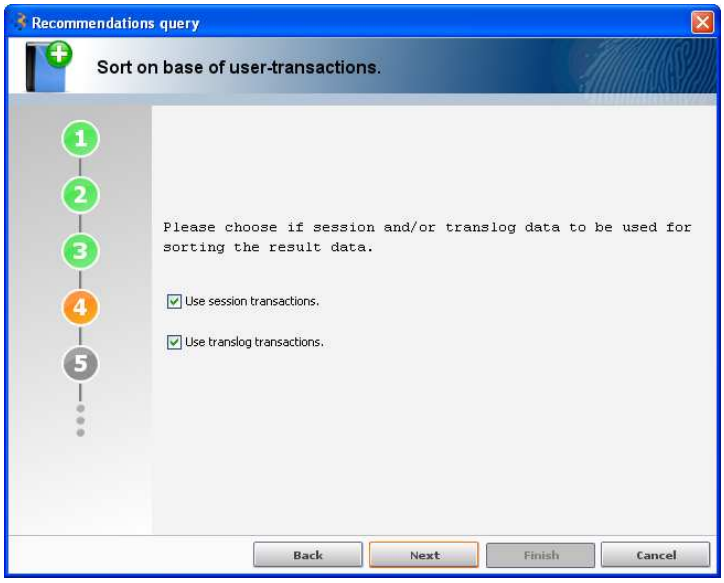


Figure 40: template category to sorted list - sort base

The sort criteria depend on session-based user transactions and/or translog information. If both options are used, the transaction data will be analysed in the following way:

- First: latest session information from the user
- Second: translog data from the user

Please note: In order to use translog data, the following configuration has to be set within the “configuration” area (therefore see Figure 22):

re.translog = true

re.translog.readTranslogs = true

Step 5: fall back settings

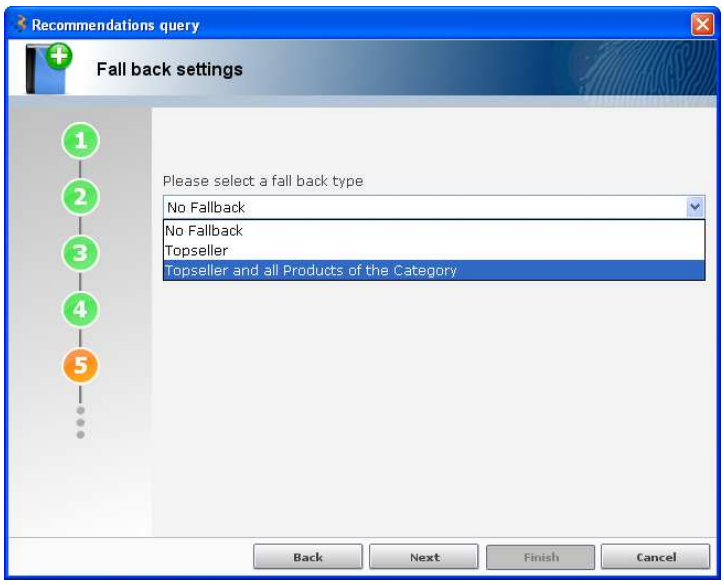


Figure 41: template category to sorted list - fall back

- No fall back: recommendations based on transaction data or session progress within the transmitted category
- Top seller: category top seller
- Top seller and all products of the category: if there are not enough category top seller, the RDE Server fulfils with remaining products of the chosen category

Please note: In order to display global top seller or category top seller, at least the following configuration has to be set within the “configuration” area (therefore see Figure 22):

re.algorithm.topseller.items = true

Optional top seller settings can be done to define top seller calculation mode or time (see Table 15).

- Step 6: attribute filter settings (page 79)
- Step 7: product filter settings (page 81)

5.11 Testing the prudsys RDE installation

There are different possibilities to test the correct installation of the prudsys RDE which will be explained in the following chapters.

5.11.1 Server start

Whenever the RDE Server is started, all important information concerning the Server start is written into an RDE specific log file (re.log). With the help of this information, the proper installation and configuration of the different RDE instances can be checked.

The log file can be found within the path /<RDE-DIR>/re_server/re_<RDE-ID>/logs/relog/ and opened directly or can be checked within the user client (see Figure 23).

Algorithm Parameters

The following general and RDE | Recommendation specific (Chapter 6.2) algorithm parameters appear at the end of the log file and show the current settings which have been created within the configuration menu (see Chapter 5.8).

Note: Some special main parameters determine other ones, e.g. the parameter controlGroupModule appears only if the property re.statistic.controlgroup was set.

Parameter within the logfile	Name within configuration menu	Default value	Description
Learning algorithm parameters			
conditional	re.alg.conditional	true	Main learning type: true – conditional false – unconditional
condSubtype	re.alg.condSubtype	DP	Conditional learning type: 0 – DP, 1 – TD
dpVersionSubtype	re.alg.dpVersionSubtype	simplified	Type of DP-version of unconditional reinforcement learning: 0 – full version 1 – simplified version
betaType	re.alg.betaType	fixed	Type for probability distribution update: 0 – constant beta (re.alg.beta) 1 – parameter beta will be adapted in real time with beta as exponent
beta	re.alg.beta	0.05	Learning rate of probability distribution, the higher the faster the rules will change
scaleCuDP	re.alg.scaleCuDP	0.5	Scaling between unconditional and conditional probabilities
controlGroupLearning DP	re.alg.controlGroupLearning DP	false	True: enables the usage of control groups to estimate the transition probabilities
transProbFilePath	re.alg.transProbFilePath	path	/re_<RDE-ID>/data/transprobs.csv
delayedLearning	re.alg.delayedLearning	false	True: enables delayed learning and disables online update.
algorithmType	re.alg.algorithmType	2	0 – MC on-policy, 1 – MC off-policy, 2 – SARSA, 3 – Q, 4 – SARSA(λ), 5 – Q(λ)

Parameter within the logfile	Name within configuration menu	Default value	Description
gamma	re.alg.gamma	0.0	Interval [0..1] Discount Rate, weight of future rewards; the lower the more "conservative"
alphaType	re.alg.alphaType	1	0 – rule update with static alpha ("fixed") 1 – parameter alpha will be adapted in real time ("mean")
alpha	re.alg.alpha	0.7	Interval [0..1] Learning rate, the higher the faster rules will change
clickReward	re.alg.clickReward	0.05	Interval [0..1] Reward for clicks
omega	re.alg.omega	0.01	Interval [0..1] Regularization of reward
useCategoryHierarchy	re.alg.useCategoryHierarchy	false	True: include category hierarchy into learning
useMultilevelPrecond	re.alg.useMultilevelPrecond	false	True: multilevel preconditioning is included into learning
suppUpdate	re.alg.suppUpdate	0.2	Support update; algorithm which cleans up the recommendation result rules
Recommendation selection parameters			
sortType	re.alg.sorttype	DP	sort criteria for online learning algorithms: 0 – support, 1 – confidence, 2 – lift/reward, 3 – clicks, 5 – P-Version, 6 – DP-Version
minProbCountDP	re.alg.minProbCountDP	20	Defines the minimum probability count of DP version
cacheDP	re.alg.cacheDP	true	True: caching for DP version is enabled to enhance the performance
activeSetPercentNew	re.alg.activeSetPercentNew	0.2	Percentage rate of new recommendation rules, if a new active set is calculated
policyType	re.alg.policyType	epsilon-greedy	Method for top seller calculation: 0 – greedy 1 – epsilon-greedy 2 – softmax
epsilon/tau	re.alg.epsilon	0.1	Interval [0..1] The higher the value the more randomized recommendations are shown

Parameter within the logfile	Name within configuration menu	Default value	Description
useActiveSet	re.alg.useActiveSet	false	True: active set is enabled Note: an “active set” is a temporal set of recommendation rules for every product which will not be changed until a timeout or a number of events occur
General			
noRecsForSelectedItems	re.alg.noRecsForSelected Items	false	True: a recommendation will not be displayed any longer if it was added to the basket
debug	re.alg.debuglevel	0	Debug level: 0 – debug is disabled, 1 – default debug, 2 – extensive debug

Table 24: algorithm parameter within re.log

Performance Indicators

Another possibility is to check certain performance indicators which also appear within the re.log file. In order to check those indicators, the data has to be pre-processed. To activate this action, the following property has to be set within the RDE Client:

re.analysis.settings.usepreprocessing = true

Afterwards, the RDE Server has to be re-started. Within the re.log file, the following statistical indicators, grouped in three areas, will appear.

Note: This is only a partial extract.

Parameter within the logfile	Description
Transaction Statistics	
nTransact	Number of transactions in transaction data
nTransactItem	Total number of transactions with reference to item class
Item Statistics	
nItems	Total number of items in item data (items.csv)
nItemTransact	Total number of items with reference to transaction class
Catalog Statistics	
nCatalogs	Total number of categories in category data
nTaxRelationships	Number of edges in the allocated category hierarchy (categoryhierarchy.csv)

Table 25: statistical key figures within re.log

5.11.2 Server information

The RDE Server provides some requests to query information about itself and installed RDE applications.

*http://host:port/rde_server/admin/**status***

The shown table provides information about the whole RDE Server, i.e. free memory. Please note: Users need permissions to request the RDE Server status (therefore see Table 2).

*http://host:port/rde_server/**serverinfo**/version*

The current RDE Server version is shown.

*http://host:port/rde_server/**serverinfo**/reqiredClient*

The required RDE Client version is shown.

*http://host:port/rde_server/**serverinfo**/res/<RDE-ID>*

The shown table provides specific RDE application information about attributes (i.e. RDE-ID, name) and count of every uploaded data (i.e. count items). These should be checked on the basis of own data files.

5.11.3 Sending Test Requests

The following requests are the basic requests for default shop integration. These should be tested in any cases. There are also extended test request within Chapter 5.12 (Integration of an RDE into an online shop).

URL specification

Please note:

- RDE IDs, template name and session IDs must not contain the characters ‘/’, ‘\’, ‘?’, ‘&’, ‘#’ and ‘:’. These are forbidden for URLs in general.
- Within parameter item ID, all characters are valid, but ‘&’, ‘?’ and ‘=’ have to be quoted.
- Quantity information always has to be declared in whole numbers (integer).

Recomm Request

*http://host:port/rde_server/res/<RDE-ID>/**recomm**/<NAME-OF-TEMPLATE>/sid/<SESSION-ID>?item=<ITEM-ID>*

Basket Request

*http://host:port/rde_server/res/<RDE-ID>/event/**basket**/sid/<SESSION-ID>*
?itemids=<ITEM-ID_1>,<ITEM-ID_N>
&quantities=<QUANTITY_1>,<QUANTITY_N>

Order Request

*http://host:port/rde_server/res/<RDE-ID>/event/**order**/sid/<SESSION-ID>*
?itemids=<ITEM-ID_1>,<ITEM-ID_N>
&quantities=<QUANTITY_1>,<QUANTITY_N>

If the RDE Server has been installed properly, sample requests can be sent using any browser.

Note: The request parameters “itemids” and “quantities” both fit for one or more products (for basket and order requests).

There are three methods to check the test requests: translog files, session logging and request logging. All alternatives are described below.

Translog files

If the writing of translg files has been activated (Table 17), those files can be used to check the test requests. On the basis the columns “basket” and “order” and their correlation, the actions can be checked.

Basket	Order	Description
0	0	Recommendation
1	0	Basket
0	1	Order

Table 26: correlation between columns

Note: Translog files are exported only after a restart of the system.

Session logging

The session logging can also be used to test the functionalities of the RDE Server. Therefore the session-ID can be used to check the exact call of events.

The session-ID has to be added first (Figure 42). After every request, the “refresh” button has to be pressed to see the effect of the request. Therewith the RDE events can be checked step by step.

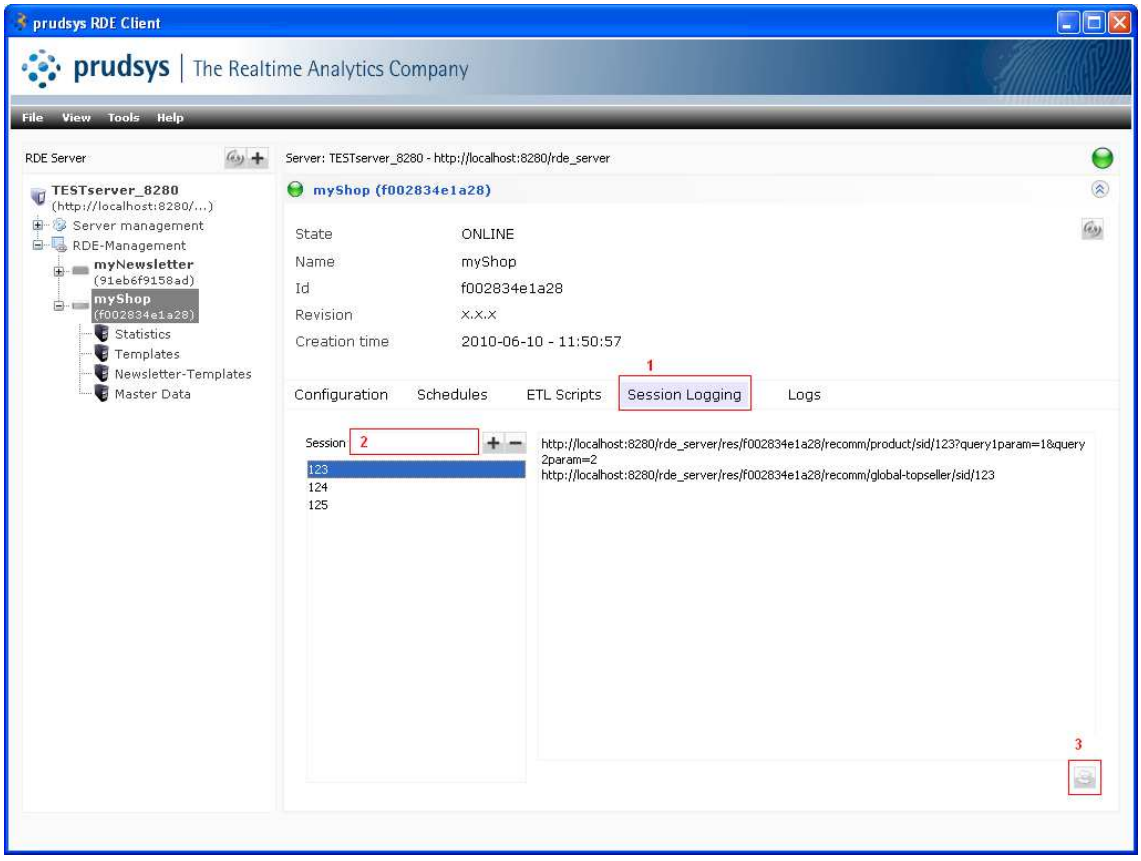


Figure 42: session logging

Request logging

The request logger is used to log every request. It is comparable with the apache access log. The log file is located within the directory re_server/re_<RDE-ID>/logs/requestlog.

Column	Value	Description
1	Incoming time	When the request arrived
2	Outgoing time	When the processing of the request is finished
3	Time elapsed	Difference between start time and end time in 1/100 sec
4	Method name	Name of the method (GET, PUT, POST, DELETE)
5	URL	The requested URL
6	Response state	The response status code

Table 27: request logging

Note: The request logger is disabled by default and should not be enabled in live mode, because the log file will expand very fast in case of the huge amount of requests.

The request logger can be enabled within the logs area of the RDE Client.

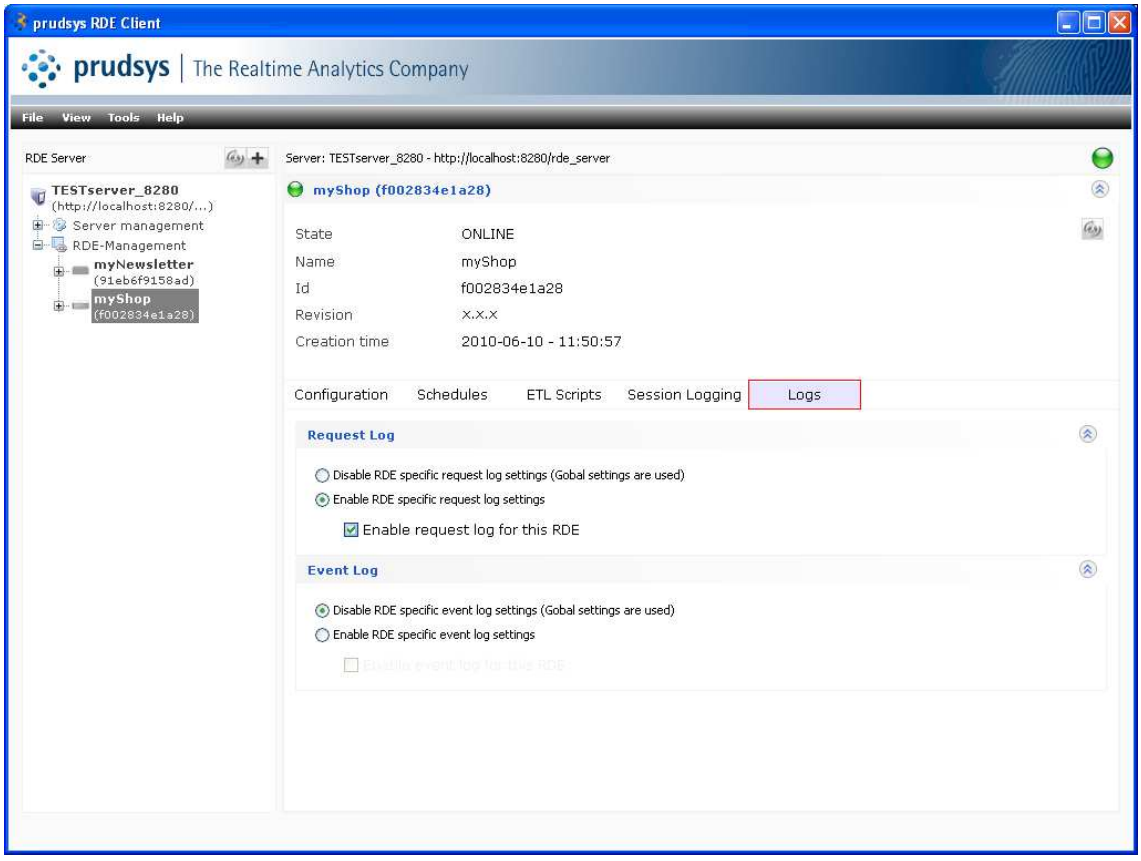


Figure 43: enable request logging

Enable request logs

Alternatively, the following POST request has to be sent e.g. with curl:

curl –X POST --digest –u username:password http://host:port/rde_server/admin/logs/loglevel/DEBUG/logger/com.prudsys.requestLogger

Disable request logs

To disable the request logger, the following POST request has to be sent:

curl –X POST --digest –u username:password http://host:port/rde_server/admin/logs/loglevel/WARN/logger/com.prudsys.requestLogger

Get request logs

A complete list of all available request logs of a specific RDE can be requested via browser:

http://host:port/rde_server/admin/logs/requestlog/res/<RDE-ID>/list

To get only a single day, this request has to be sent via browser:

http://host:port/rde_server/admin/logs/requestlog/res/<RDE-ID>/day/<DAY>

Please note: The required format of <DAY> is YYYY-MM-DD.

Additionally, users can specify the number of latest entries by declaring the parameter "last":

http://host:port/rde_server/admin/logs/requestlog/res/<RDE-ID>/day/<DAY>?last=<LAST_ENTRIES>

Delete request logs

To delete all request logs, the following DELETE request has to be sent e.g. with curl:

curl –X DELETE --digest –u username:password
http://host:port/rde_server/admin/logs/requestlog/res/<RDE-ID>/list

To delete a request log for only one specific day, the following DELETE request has to be sent e.g. with curl:

curl –X DELETE --digest –u username:password
http://host:port/rde_server/admin/logs/requestlog/res/<RDE-ID>/day/<DAY>

Please note: Users need permissions to get or delete request logs (therefore see Table 2).

5.12 Integration into an online shop

5.12.1 Tracking events

Whereas the requests described in Chapter 6.4 are using the so called “recomm” event for tracking as well as the return of recommendations, the following events “basket” and “order” are only used to support the tracking of the customers’ purchase behaviour. The enhanced “recommit” event provides more detailed information about actually displayed recommendations. Therefore, learning behaviour of the RDE will be improved.

These tracking events are indispensable for the learning mechanisms of the recommendation engine and have to be integrated into the shop frontend.

Please note:

- RDE IDs, template name and session IDs must not contain the characters ‘/’, ‘\’, ‘?’, ‘&’, ‘#’ and ‘:’. These are forbidden for URLs in general.
- Within parameter item ID, all characters are valid, but ‘&’, ‘?’ and ‘=’ have to be quoted.
- Quantity information always has to be declared in whole numbers (integer).

Basket event

Request example for the basket event:

http://host:port/rde_server/res/<RDE-ID>/event/basket/sid/<SESSION-ID>
?itemids=123,456,789&quantities=3,1,1

In this example, three units of product 123 and one unit of the products 456 and 789 have been put once into the shopping cart.

Order event

Request example for the order event:

http://host:port/rde_server/res/<RDE-ID>/event/order/sid/<SESSION-ID>
?itemids=123,456,789&quantities=3,1,1

In this example, three units of product 123 and one unit of the products 456 and 789 have been ordered.

Note: The request parameters “itemids” and “quantities” both fit for one or more products (for basket and order requests).

Recommit event

Sometimes not all recommendations delivered by the RDE are actually displayed in the shop. For example, there may be applied external filters before the final recommendations are displayed. In order to inform the RDE about the effectively displayed recommendations, the recommit event needs to be called:

http://host:port/rde_server/res/<RDE-ID>/event/recommit/sid/
<SESSION-ID>?itemids=<ITEM-1>,...,<ITEM-N>&includeNewRules=<true|false>

The recommit event always applies to the last recommendation event of the session requested by the RDE. For *includeNewRules=false*, the submitted recommendation items must be a subset of the recommendations delivered by the previous recommendation event. This case applies to the previous example of external filters. Note: the recommit event is not required if all recommendations of the RDE have been displayed.

The option *includeNewRules=true* is more powerful. In this case, also external recommendations like manual ones, which have not been delivered by the RDE, can be committed. The RDE includes this information into its learning.

Control transmission

If the event has been transmitted successfully, no message will be generated. In order to control the correct transmission, the translog file within the directory re_server/re_<RDE-ID>/translog can be checked. This file is updated at every restart of the recommendation engine. For more information about the translog file, see Table 18.

5.12.2 Special events

Besides to the recommendation (recomm) and tracking events (basket, order) which are necessary for the standard functionalities of the RDE, there are additional events for the realization of extended functionalities (e.g. personalized recommendations).

The following special events can be used to exchange information with the prudsys recommendation engine.

User to session

The event usertosession assigns a user ID to a session. This event can be called at any moment during a running session. Nevertheless, it is recommendable to execute the request only after the user has been identified (via cookie, log-in, etc.)

*http://host:port/rde_server/res/<RDE-ID>/event/**usertosession**/sid/<SESSION-ID>/userid/<USER-ID>*

Channel to session

The event channel to session identifies the source of a click, whether it was a recommendation or a newsletter click.

*http://host:port/rde_server/res/<RDE-ID>/event/**channeltosession***
/sid/<SESSION-ID>/channelid/<CHANNEL-ID>

Update one attribute for one item / banner

The event updateItemAttribute allows the modification of a product or banner value (both defined by request parameter “item”) at runtime without having to change the product/banner master data, for example to update the availability status of a product/banner (online flag).

http://host:port/rde_server/admin/res/<RDE-ID>/event/
updateItemAttribute*?****item****=<ITEM-ID>&****attribute****=<ATTRIBUTE>*
*&****value****=<NEW-VALUE>*

http://host:port/rde_server/admin/res/<RDE-ID>/event/
updateItemAttribute*?****item****=<BANNER-ID>&****attribute****=<ATTRIBUTE>*
*&****value****=<NEW-VALUE>*

example:

http://host:port/rde_server/admin/res/<RDE-ID>/event/
updateItemAttribute*?****item****=1&****attribute****=onlineFlag&****value****=0*

Update several attributes for one item / banner

There is also a possibility to update several attributes for one product or banner.

http://host:port/rde_server/admin/res/<RDE-ID>/event/
updateItemAttributes*?****item****=<ITEM-ID>*
*&****attributes****=<ATTRIBUTE_1>,<ATTRIBUTE_N>*
*&****values****=<NEW-VALUE_1>,<NEW-VALUE_N>*

http://host:port/rde_server/admin/res/<RDE-ID>/event/
updateItemAttributes*?****item****=<BANNER-ID>*
*&****attributes****=<ATTRIBUTE_1>,<ATTRIBUTE_N>*
*&****values****=<NEW-VALUE_1>,<NEW-VALUE_N>*

example:

http://host:port/rde_server/admin/res/<RDE-ID>/event/
updateItemAttributes*?****item****=1&****attributes****=onlineFlag,netUnitPrice&****values****=0,35.50*

Note: Several attributes and values will be separated by commas.

Using the column netUnitPrice, please take into account that the decimal places have to be separated from the pre-decimal places by a decimal point (English format).

Update one attribute for several items / banners

Additionally, there is a possibility to modify one attribute for several products or banners.

http://host:port/rde_server/admin/res/<RDE-ID>/event/
updateItemsAttribute*?****items****=<ITEM-ID_1>,<ITEM-ID_N>*
*&****attribute****=<ATTRIBUTE>*
*&****values****=<NEW-VALUE_1>,<NEW-VALUE_N>*

http://host:port/rde_server/admin/res/<RDE-ID>/event/
updateItemsAttribute*?****items****=<BANNER-ID_1>,<BANNER-ID_N>*
*&****attribute****=<ATTRIBUTE>*
*&****values****=<NEW-VALUE_1>,<NEW-VALUE_N>*

example:

http://host:port/rde_server/admin/res/<RDE-ID>/event/
updateItemsAttribute*?****items****=1,2,3,4&****attribute****=onlineFlag&****values****=0,1,1,0*

Controlling updates

To control the modifications, the following request has to be sent:

http://host:port/rde_server/admin/res/<RDE-ID>/items/online/<ITEM-ID>

http://host:port/rde_server/admin/res/<RDE-ID>/banners/online/<BANNER-ID>

Note: Users need permissions to update items or banners (therefore see Table 2).

CSV update

In order to update complete new products, product categories, categories, category hierarchy or content, the RDE Server provides special requests. These requests are HTTP PUTs which can be called e.g. with curl.

*curl -X PUT -T "file path" -H "Content-Type: text/csv" --digest -u username:password "http://host:port/rde_server/admin/res/<RDE-ID>/**<TYPE>**/insertOnlineData?separator=%7c&**updateData**=true|false"*

TYPE can be the assumed the following specifications:

- items
- categories
- itemcategories
- banners
- categoryhierarchies

example:

*curl -X PUT -T "file path" -H "Content-Type: text/csv" --digest -u username:password "http://host:port/rde_server/admin/res/<RDE-ID>/**items**/insertOnlineData?separator=%7c&updateData=true|false"*

The parameter “updateData” can adopt the values true or false:

- True: New product data will be inserted and existing data will be updated row-wise
- False: New product data will be inserted, but existing data will not be updated

The data updated at runtime only persist as long as the prudsys RDE is active.

Please note: For mode “updateData=true”, only still existing columns can be filled through the upload process. This means the columns have to be identically to the columns from items.csv (offline upload), even if there are no new values.

Attention: Missing columns will be deleted (they will be overwritten with “no value”)!

Column-wise CSV update

Special case – column-wise CSV update only for items and banners with updating data records (“updateData=true”).

The optional parameter “updateMode=column_wise” can be set. With this mode, the data records will be updated column-wise. This means, only columns with new values have to be uploaded. Therefore, missing columns will not be deleted.

Please note: column pid is required.

example:

*curl -X PUT -T "file path" -H "Content-Type: text/csv" --digest -u username:password "http://host:port/rde_server/admin/res/<RDE-ID>/items/insertOnlineData?separator=%7c&updateData=**true**&**updateMode=column_wise**"*

Updated data can be displayed by the following requests

http://host:port/rde_server/admin/res/<RDE-ID>/items/online/<ITEM-ID>

http://host:port/rde_server/admin/res/<RDE-ID>/banners/online/<BANNER-ID>

http://host:port/rde_server/admin/res/<RDE-ID>/categories/online/<CATEGORY-ID>

Note: Users need permissions to update and view CSV files (therefore see Table 2).

Set control group

The event setcontrolgroup allows the external assignment of a session to a group. The RDE supports multiple control groups, being "0" always the group which receives recommendations. If there is only one control group, its index is "1". In the case of multiple control groups, their indices depend on their application mode. Normally, the RDE handles the assignment of control groups automatically, but the event setcontrolgroup (using authentication) allows the external configuration of control groups.

*http://host:port/rde_server/admin/res/<RDE-ID>/event/**setcontrolgroup**/sid/<SESSION-ID>/<CONTROLGROUP-ID>*

Note: Users need permissions to set the control group (therefore see Table 2).

Get control group

The event getcontrolgroup returns the group the current session has been assigned to (no matter if the assignment was executed automatically by the RDE or manually via setcontrolgroup).

*http://host:port/rde_server/res/<RDE-ID>/event/**getcontrolgroup**/sid/<SESSION-ID>*

RDE status

The event RDE status tests if the RDE has been initialized and started correctly. Results are either RE_STATUS_ONLINE or RE_STATUS_OFFLINE. If the RDE Server is still running, the HTTP code is always 200.

*http://host:port/rde_server/res/<RDE-ID>/event/**restatus***

RDE ping

The event RDE ping tests like "RDE status", but returns are different:

- RDE is online (HTTP code 200 and return string "RE_STATUS_ONLINE")
- RDE is offline (HTTP code 503 and return string "RE_STATUS_OFFLINE")
- RDE is not available (HTTP code 404)

*http://host:port/rde_server/res/<RDE-ID>/event/**ping***

Save models

The event save models stores the current recommendation rules and transition probabilities.

*http://host:port/rde_server/admin/res/<RDE-ID>/event/**savemodels***

Latest files will be stored within directory data/re_server/re_<RDE-ID>/data as resultrules_out.csv and transprobs.csv.

Note: Users need permissions to save current rules (therefore see Table 2).

Banner / category clicked

The events banner clicked and category clicked show whether a banner or a category was clicked during a session.

*http://host:port/rde_server/res/<RDE-ID>/event/**bannerclicked**/sid/<SESSION-ID>/bid/<BANNER-ID>*

*http://host:port/rde_server/res/<RDE-ID>/event/**categoryclicked**/sid/<SESSION-ID>/cid/<CATEGORY-ID>*

Last clicked

The event last clicked shows a history of all last clicked products or categories within one session as a JSON object. The request can be called afterwards the following recommendations or event:

- product-to-product recommendation
- category-to-banner recommendation
- customer-to-banner recommendation
- category clicked event

*http://host:port/rde_server/res/<RDE-ID>/event/**lastclicked**/sid/<SESSION-ID>*

Example: One last clicked product and two last clicked categories are shown.


```
[{
    "lastClickedProducts": [
      {
        "netUnitPrice": "34.0",
        "onlineFlag": "1.0",
        "reward": "34.0",
        "SKU": "1126",
        "strikeOutPrice": "0.0",
        "type": "",
        "product_nr": "1126",
        "quantityUnit": "",
        "UID": "1126",
        "description": "",
        "manufacturer": "",
        "name": "",
        "quantity": "0.0",
        "URL": "index.php?cl=details&anid=1126",
        "imageURL": "out/pictures/1/1126_p1.jpg"
      }
    ]},
  {
    "lastClickedCategories": [
      {"UID": "8a142c3e4143562a5.46426637"},
      {"UID": "8a142c3e49b5a80c1.23676990"}
    ]}]
```

Optional parameters can be set independent of each other in the following way:

Parameter	Value	Description
type	product | category, default: both	What type of clicks should be shown
last	[0, ∞], default: all	Number of last shown products or categories
duplicates	[true, false], default: false	True: multiple clicked products or categories will be shown separately
split	[true, false], default: true	False: JSON object will not be split into products and categories (like seen above), but together within last clicked elements

Table 28: optional parameters for last clicked event

The following example request only shows the last two categories with duplicates:

http://host:port/rde_server/res/<RDE-ID>/event/lastclicked/sid/<SESSION-ID>
*?**type**=category&**last**=2&**duplicates**=true*

6 prudsys RDE | Recommendations

6.1 Introduction to RDE | Recommendations Module

The prudsys RDE | Recommendations module enables the optimum use of cross- and up-selling potential in e-commerce, telemarketing and high street shopping. It generates recommendations for appropriate products and content based on actual user behaviour. This ensures a high level of personal relevance.

Your benefit is a high acceptance rate for recommendations, longer browsing time in the shop and increasing turnover.

Business scenarios

The prudsys RDE | Recommendations module enables comprehensive optimisation and sales process personalisation at all customer touch points, without manual intervention. The range of solutions covers product recommendations, the personalisation of content in online shops and newsletters, social search and the personalised ranking of product lists (e.g. search results pages). So, for example, the module will analyse enquiry sequences to show the order in which products are usually sold and the order in which content is usually viewed.

With the real-time learning function, the module uses the ongoing interaction with the user to learn about user behaviour and uses this information to make recommendations in real-time.

How it works

The RDE | Recommendations module bases its recommendations on an evaluation of historic transaction data and on real-time learning from the ongoing interaction with users and visitors.

Historic data is not absolutely necessary but it has the advantage that it improves the quality of recommendations. If information about previous transactions is available, the module will not only tell you what products were viewed or sold together but it will also indicate other deciding factors in a purchase.

6.2 Configuration parameters for RDE | Recommendations

After creating a new RDE application (Chapter 5.2), the specific configuration parameters for RDE | Recommendations can be set comfortably using the menu “configuration” like described within Chapter 5.8).

6.2.1 Live mode properties (real time learning)

Overview

There are many parameters to control the real time learning algorithms. However, in most cases none of them needs to be changed. The main real time learning approach of the RDE is based on reinforcement learning (RL). Due to the fact that RL algorithms are very complex in detail, we only introduce some basic terms required for better understanding the RDE parameters.

RL calculates an action-value function that assigns the expected reward to all products and their recommendations throughout the remaining session. Within the RDE, it is stored by recommendation rules and – in some cases – additional data structures. There are two parts of RL: selecting the best recommendations and online learning.

Best recomms

First of all the RDE needs to select the best recommendations, i.e. those ones which have the highest action values. This process is called “exploit” mode. On occasion, some new recommendations have to be tested by the RDE. This process is called “explore” mode. The right interaction between exploit and explore is defined by the policy selection algorithm. The RDE offers two standard approaches: epsilon-greedy and softmax policy.

Online learning

The online learning is the heart of RL. First of all, we distinguish between “unconditional” and “conditional” RL. Unconditional RL means that the algorithm does not directly learn from its recommendations but analyzes all transactions no matter what recommendations have been displayed. That is, the probabilities are considered as being not conditioned by the recommendations. This process follows the typical data mining philosophy also used in almost all state-of-the-art recommendation engine algorithms. The problem of finding the best recommendations is mainly considered as an analysis task.

In contrast, the standard prudsys RDE approach is the conditional RL where all recommendation calculations take into account the actually delivered recommendations, too. This is a completely new approach which directly follows the RL theory and considers the calculation of recommendations as a real control problem.

Conditional or un-conditional

Notice that technically both approaches are based on the RL framework, especially solving the Bellman equation. Obviously, conditional RL is more complex than unconditional and requires a clean integration of the RDE Server into the online application. Especially, the recommit event always needs to be called if some recommendations delivered by the RDE have not been displayed. The unconditional RL delivers a lower quality of recommendations but it is more simple and robust. The central parameter *re.alg.conditional* defines whether the conditional or unconditional RL version has to be used.

DP or TD

There is another important distinction between Dynamic Programming (DP) and Temporal-Difference (TD) learning. DP uses the whole transition model including the transition probabilities and rewards which are estimated by the RDE in order to calculate the action-value function. The TD algorithm directly updates the action-value function and hence requires less data to be stored.

Both approaches have their advantages and disadvantages. The prudsys RDE parameters *uncondSubtype* and *condSubtype* define whether DP or TD version is used for unconditional and conditional RL, respectively. DP version is configured by default.

As a result we operate with 4 basic RL versions of the RDE: unconditional DP (also called P version), unconditional TD (former standard algorithm), conditional DP (current standard version) and conditional TD.

Property	Value	Description
module.recommendations. recommitall	true | false default: false	All displayed recommendations will be transmitted to the RDE Server
rde.service.recommendation. globalSortType	0 – support, 1 – confidence, 2 – lift/reward, 3 – clicks, 5 – P-Version, 6 – DP-Version (default)	Global sort type for rule reduction during import process
re.alg.activeSetModule	-1 (default) | [0..integer_max]	-1: disabled otherwise: number of detail views until a new action set is calculated

Property	Value	Description
re.alg.activeSetPercentNew	[0.0 .. 1.0] default: 0.2	Percentage rate of new recommendation rules, if a new active set is calculated
re.alg.activeSetSize	[0..integer_max] default: 10	Number of recommendation rules within action set
re.alg.activeSetTimeout	-1 | [0..integer_max] default: 3600	-1: disabled otherwise: timeout in seconds until a new action set is calculated
re.alg.algorithmType	0 – MC on-policy, 1 – MC off-policy, 2 – SARSA (default), 3 – Q, 4 – SARSA(λ) 5 – Q(λ)	Reinforcement learning algorithm type
re.alg.alpha	[0..1] default 0.7	Learning rate of reward update, the higher the faster rules will change
re.alg.alphaType	0 | 1 default: 1	Type for online rule update: 0 – rule update with constant alpha (re.alg.alpha) 1 – parameter alpha will be adapted in real time with alpha as exponent
re.alg.beta	[0..1] default: 0.05	Learning rate of probability distribution, the higher the faster the rules will change
re.alg.betaType	0 | 1 default: 0	Type for probability distribution update: 0 – constant beta (re.alg.beta) 1 – parameter beta will be adapted in real time with beta as exponent
re.alg.cacheDP	true | false default: true	True: caching for DP version is enabled to enhance the performance
re.alg.clickReward	[0..1] default: 0.05	Reward for clicks
re.alg.conditional	true | false default: true	Main learning type: true – conditional false – unconditional
re.alg.condSubtype	0 | 1 default: 0	Conditional learning type: 0 – DP 1 – TD Note: property re.alg.conditional = true is required
re.alg.uncondSubtype	0 | 1 default: 0	Unconditional learning type: 0 – P 1 – TD Note: property re.alg.conditional = false is required
re.alg.controlGroupLearningDP	true | false default: false	True: enables the usage of control groups to estimate the transition probabilities

Property	Value	Description
re.alg.delayedLearning	true | false default: false	True: enables delayed learning and disables online update. Recommendation rules will be updated additionally while deactivating the RDE application. Rules will only be adapted by purchase data from registered users. This method tends to prevent down selling.
re.alg.dpVersionSubtype	0 | 1 default: 1	Type of DP-version of unconditional reinforcement learning: 0 – full version 1 – simplified version
re.alg.epsilon	[0..1] default 0.1	Exploration rate: the higher the value the more randomized recommendations are shown
re.alg.gamma	[0..1] default 0.0	Discount rate, weight of future rewards; the lower the more "conservative"
re.alg.lambda	[0..1] default 0.9	Propagates learning to past rules; the higher the faster is the learning
re.alg.minProbCountDP	[0..integer_max] default: 20	Defines the minimum probability count of DP version
re.alg.newAlphaStep	[0..∞], default: 20	Number of steps until alpha is constant for all states Note: properties alphaType = 1 is required
re.alg.noDoubleClickStats	true | false default: false	True: the statistic count a recommendation only once per session, even if it is clicked several times
re.alg.noRecsForSelectedItems	true | false default: false	True: a recommendation will not be displayed any longer if it was added to the basket
re.alg.omega	[0..1] default 0.01	Regularization parameter of reward
re.alg.penalType	0 | 1 default: 0	Type of penalisation of unconditional learning: 0 – recommendations will be penalised 1 – all products will be penalised Note: Only for unconditional TD
re.alg.policyType	-1 – random 0 – greedy, 1 – epsilon-greedy (default), 2 – softmax 3 – auto softmax	Policy type to select recommendations: For epsilon-greedy, re.alg.epsilon defines the exploration rate. For softmax, the same parameter defines the temperature. Auto softmax automatically determines the right temperature parameter in order to achieve an exploration rate also given by re.alg.epsilon.
re.alg.pVersionSubtype	0 | 1 default: 0	Algorithm type of P-Version: 0 – full version 1 – simplified version
re.alg.scaleCuDP	[0..1] default: 0.5	Scaling between unconditional and conditional probabilities

Property	Value	Description
re.alg.sorttype	0 – support, 1 – confidence, 2 – lift/reward, 3 – clicks, 5 – P-Version, 6 – DP-Version (default)	sort criteria for online learning algorithms
re.alg.transProbFilePath	data/transprobs.csv	Path of the file for DP learning transition probabilities
re.alg.useActiveSet	true | false default: false	True: active set is enabled Note: an “active set” is a temporal set of recommendation rules for every product which will not be changed until a timeout or a number of events occur
re.statistics.rec.acc.window.size	[0..∞] default: 200	The order value generated through previously recommended items will only be considered within this period of actions

Table 29: configuration parameters - online learning

RDE specific log file

Note: To check these settings within the RDE Server, users can control the re.log file in the directory re_server/re_<RDE-ID>/logs/re.log (therefore see Chapter 5.11.1).

Rule growth

The limitation of recommendation result rules per products enables a better control of the memory load of the server. The support update is an algorithm which cleans up the recommendation result rules, which means older rules will get a lower weight over the time so that the most current and up to date rules will be increased.

Settings for rule growth / adjustment		
Property	Value	Description
re.rule.adjustment.enable	true | false default: false	limitation of rules for every product
re.rule.adjustment.rulesPerItem	<amount> default: 20	maximum amount of rules for very product
re.alg.suppUpdate	<amount> default: 0.2	support update, only for unconditional TD version

Table 30: configuration parameters - rule growth / adjustment

6.2.2 Hierarchical Learning

Hierarchical learning

In order to speed up the learning and to find more recommendations, the prudsys RDE allows to calculate rules based on hierarchy levels. These algorithms are called hierarchical reinforcement learning and can improve recommendations in many aspects.

There are two version of hierarchical RL implemented. The first version, selected by the parameter *re.alg.useCategoryHierarchy*, directly calculates and stores the hierarchical

rules. The second version, selected by *re.alg.useMultilevelPrecond*, implicitly uses hierarchies for multilevel preconditioning. In that case, no hierarchical rules are calculated and stored.

Online learning – Hierarchical Methods		
Property	Value	Description
re.alg.useCategoryHierarchy	true | false default: false	True: category rules are included into recommendation rules
re.alg.categoryHierarchyType	0 – new rules have support value 0 (default) 1 – new rules have support value 0.95	This parameter applies to option re.alg.newCategRulesPerStep
re.alg.useMultilevelPrecond	true | false default: false	True: multilevel preconditioning is included into learning, which is required for the next options
re.alg.multilevelMaxParentSize	[0, ∞] default: 100	Maximum number of children a category node may posses. Used for internal preprocessing. Note: property re.alg.useMultilevelPrecond = true is required.
re.alg.multilevelPrecondLevels	[2, ∞] default: 2	Number of levels used for multilevel preconditioning. Note: property useMultilevelPrecond = true is required.
re.alg.multilevelWeightedProlong	true | false default: false	True: weighting (by number of actions) will be used in prolongation operator. Note: this option is more time consuming. Note: property useMultilevelPrecond = true is required.
re.alg.newCategRulesPerStep	[0, ∞] default: 0	Number of new recommendations per rule that are added, if an item has not enough recommendations. 0 – parameter will be ignored, otherwise: if re.alg.categoryHierarchyType is true, category rules are used for this option. Note: property re.alg.useMultilevelPrecond = true is required and categoryHierarchyType overwrites useMultilevelPrecond

Table 31: configuration parameters - hierarchical method

6.2.3 Banner configuration

Banners

If banner recommendations are active, the file reitems.csv within the data directory will be filled as well. The column “type” with value “b” indicates a banner.

If the RDE generates banner rules (rule type 1), the file resultrules.csv contains new banner and category rules with prefix _b_<bid> or _c_<cid>.

Settings for banner calculation		
Property	Value	Description
re.recommendations.banner	true | false default: false	True: enable banner recommendations
re.recommendations.banner.ruletype	0 | 1 default: 1	0 – generate no banner rules, 1 – generate banner rules for each banner and category during start of the RDE Note: property re.recommendations.banner = true is required.
re.recommendations.banner.defcategory	<default category>	Default category for recommendation template customer-to-banner, if there is no last clicked category

Table 32: configuration parameters – banner

6.3 RDE | Recommendations templates

6.3.1 Basic settings

These basic setting options are described in detail within Chapter 5.10.1.

6.3.2 RDE | Recommendations: output settings

If JSON or JSONSIMPLE output is chosen, after a successful request a file will be delivered which can be opened with any editor.

- JSONSIMPLE: Format for recommendations which have been created with an older version of the RDE Server (lower than revision 2.0). In this case, no combined requests (several different requests within one template) can be created.

JSON SIMPLE: specific example

```
[
  {
    "netUnitPrice": "30.99",
    "onlineFlag": "1.0",
    "reward": "30.99",
    "param1": "white",
    "product_nr": "p2k1",
    "quantityUnit": "quantity",
    "UID": "p2k1",
    "manufacturer": "manufacturerA",
    "name": "cupboard",
    "brand": "brandB",
    "masterUID": "",
    "quantity": "100.0",
    "imageURL": "image"
  }
]
```

- JSON: Combined requests can be created

JSON: general example

```
[
  {
    "query1": []
  },
  {
    "query2": []
  },
  {
    "global": {
      "controlgroup": "false"
    }
  }
]
```

JSON: specific example

```
[ {"globTops": [
  {
    "netUnitPrice": "30.99",
    "onlineFlag": "1.0",
    "reward": "30.99",
    "param1": "white",
    "product_nr": "p2k1",
    "quantityUnit": "quantity",
    "UID": "p2k1",
    "manufacturer": "manufacturerA",
    "name": "cupboard",
    "brand": "brandB",
    "masterUID": "",
    "quantity": "100.0",
    "imageURL": "image"
  }]},
{"global": {
"controlgroup": "false"
}}]
```

- HTML
- TEXT

Character encoding:

- UTF-8
- ISO-8859-1

Name of Velocity template:

- If HTML or TEXT output is chosen, a velocity template has to be specified which will be filled with the recommendation returns from the RDE and delivered as HTML page or TEXT output to the sender. There are pre-defined one standard HTML template (default.vm) and one TEXT template (defaultTEXT.vm) within the directory /templates. Customized templates can be stored within the same directory.
- The templates have to be written in a special syntax. For more information see Apache Velocity Project.

JavaScript callback function name:

- If the recommendations should be delivered as JavaScript function, the name of the function has to be indicated. The return format should be JSON.

```
if((typeof <function_name>) == 'function')
{<function_name>
([{"<template_name>":[{"netUnitPrice":"29.99","onlineFlag":"1.0",
"reward":"14.0","SKU":"item123","strikeOutPrice":"69.99","product
_nr":"123","netPurchasePrice":"15.99","quantityUnit":"unit","UID"
:"051640000003","description":"Table","manufacturer":"","name":"T
able
black","brand":"","masterUID":"","URL":"http://<url>","imageURL":
"http://<imageurl>"}]},{"global":{"controlgroup":"false"}}]); }
```

Escape characters:

- This is normally not necessary, as most browsers already escape special characters, i.e. special characters are mostly displayed correctly.

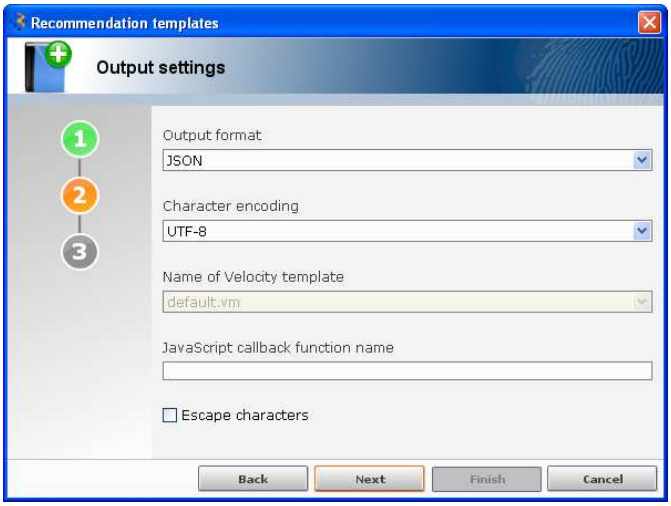


Figure 44: output settings

6.3.3 Add queries

In this final step, recommendation templates will be created. It is described in detail within Chapter 5.10.3.

Template URL

If a recommendation template is created with the RDE Client, the corresponding HTTP request named “Template URL” is displayed within the window (see Figure 45). This request has to be sent to the RDE Server in order to query the corresponding RDE.

Note: The resulting template URLs differ between the modules RDE | Recommendations and RDE | Newsletter. Newsletter template URLs will be described in detail within Chapter 7.7.3.

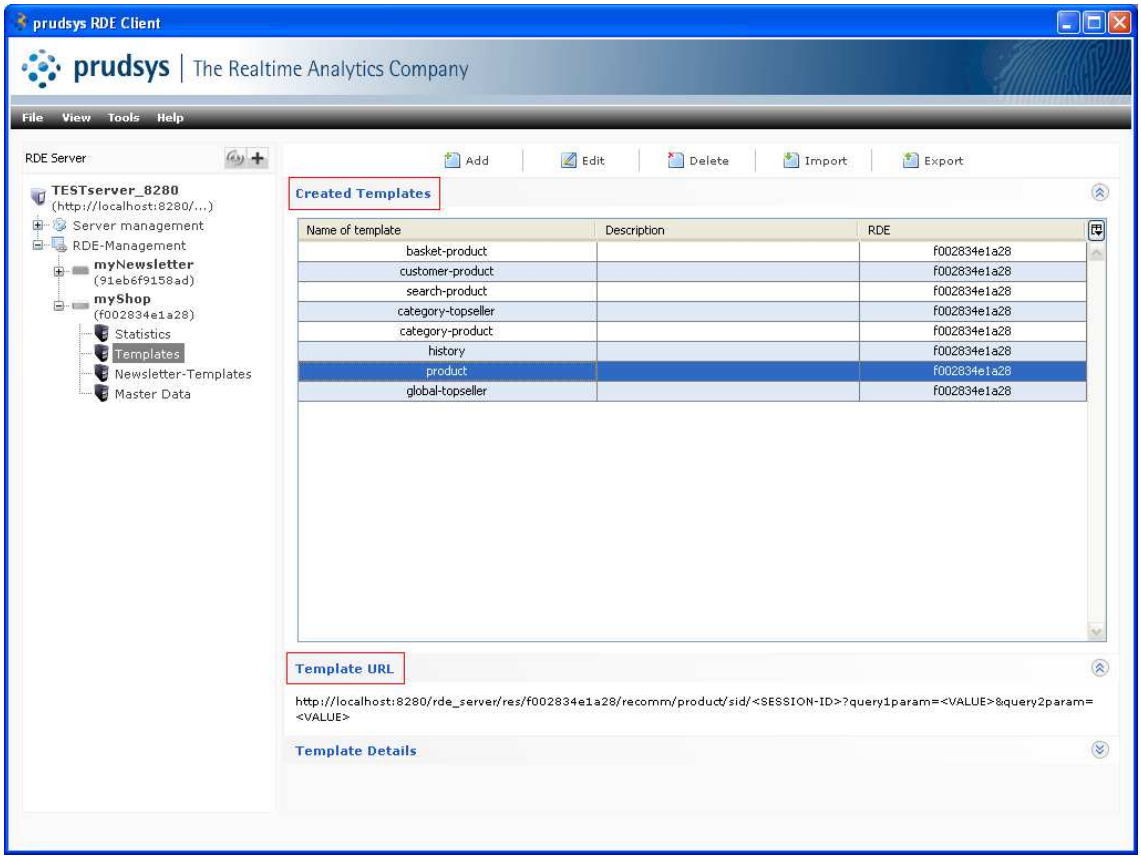


Figure 45: templates – overview

6.4 Requesting recommendations

Before the requests can be tested, the RDE application has to be activated. Afterwards, the test calls can be sent directly via HTTP requests in order to test the correct functioning of the settings.

There are differences in personalized and anonymous recommendations, which are described in the followings section.

Personalized recomm

Personalized recommendations always consist of two separate requests. The first request is used to assign the user to a session, for example when he/she logs in with his/her user data. Afterwards, personalized recommendations can be calculated and displayed on the basis of his/her user profile and historic purchase data.

Without the request user to session, the recommendations will be calculated only based on the latest session information.

*http://host:port/rde_server/res/<RDE-ID>/event/**usertosession**/sid/<SESSION-ID>/userid/<USER-ID>*

*http://host:port/rde_server/res/<RDE-ID>/**recomm**/<NAME-OF-TEMPLATE>/sid/<SESSION-ID>*

Note: Templates should have the input customer (therefore see Table 21).

Anonymous recomm

Recommendations for anonymous users consist of only one request and are always structured like the following two examples:

*http://host:port/rde_server/res/<RDE-ID>/**recomm**/<NAME-OF-TEMPLATE>/sid/<SESSION-ID>?item=<ITEM-ID>*

Note: The parameter “item” is the previously chosen name for the query parameter and can be set at will (therefore see Figure 30).

Depending on the chosen recommendation origin and name, ITEM-ID could be a single product, a product list, a category or a search term. Dynamic parameters can be attached by the “?<parameter>=<value>”. Several filter parameters can be combined with “&”.

6.5 RDE | Recommendations Statistics

The client allows the creation of individual statistic dash boards with customized tables and charts, using all available data logged by the RDE server.

To support a quick start, four default worksheets are provided through the RDE Client:

- Year statistic: number of transactions (nTrans), orders (nOrd) and shopping carts (nBask)
- Month statistic: number of transactions, orders and baskets
- Year statistic: revenue (rev), revenue of accepted recommendations (revRecAcc)
- Month statistic: revenue, revenue of accepted recommendations

Note: All short cuts are described within Table 33.

Step 1: Creating a new worksheet (Figure 46)

- title
- colour
- comments

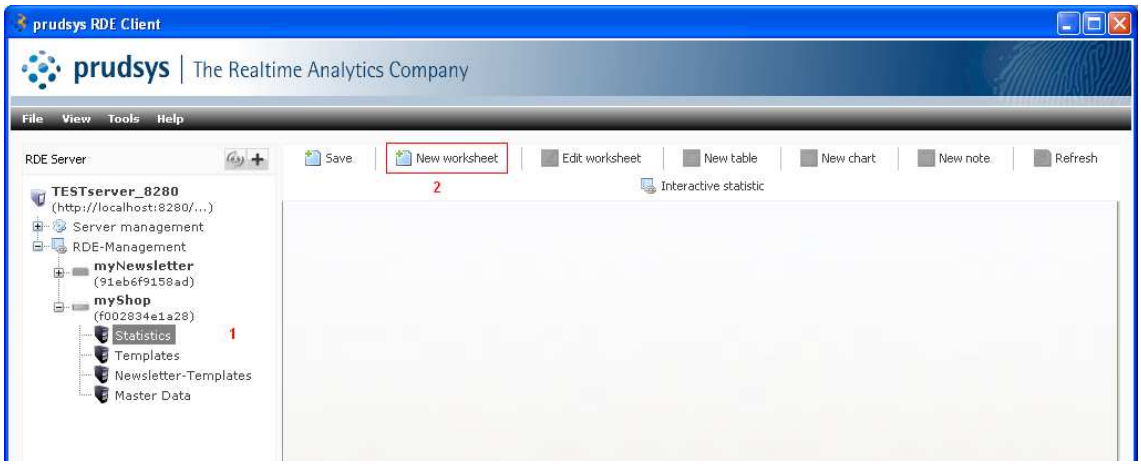


Figure 46: statistic overview

Step 2: Creating a table, chart or comment

- press the corresponding button
- change settings of all objects with a right-click
- save settings

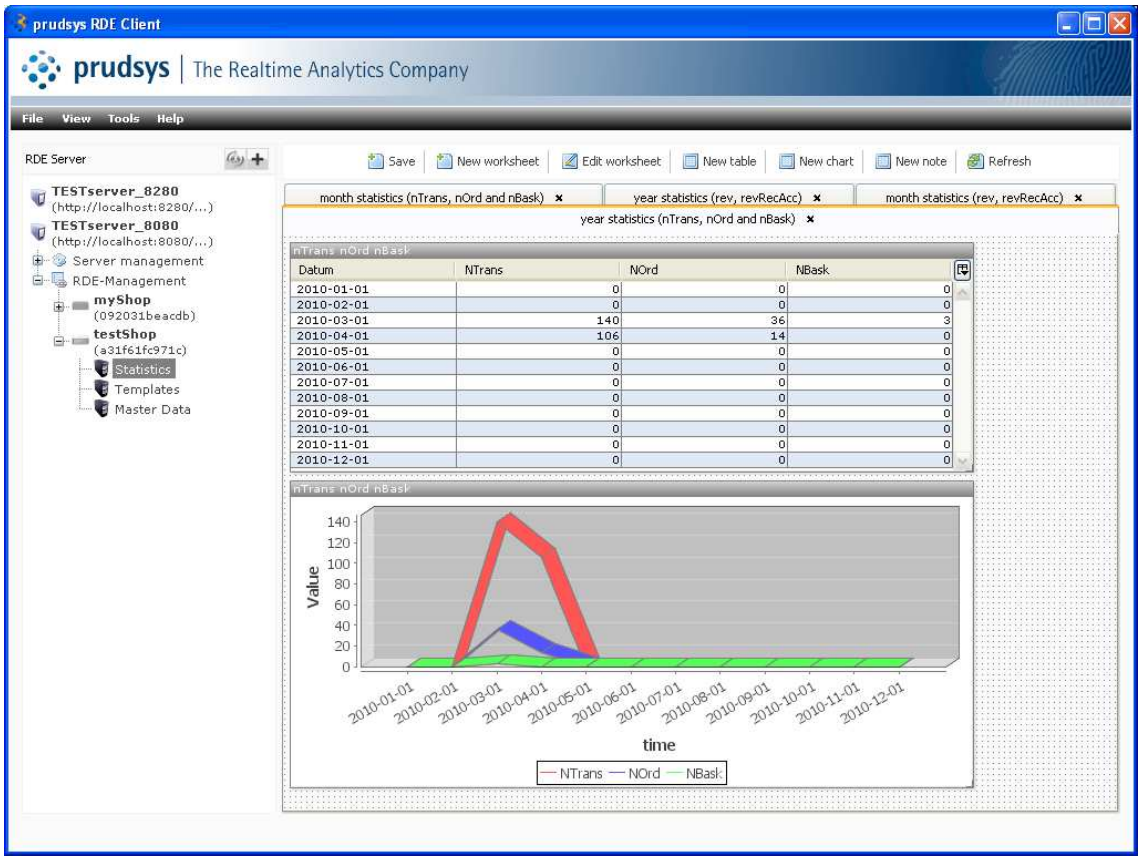


Figure 47: statistic worksheet

Chart and table settings:

- main settings: name, font family, font size
- column definition:
  - time base: day, week, month, year
  - key performance factors, which are described in the following table
- extended settings: four different chart and table types are available

Figure 48: selection of chart types

key performance factor	description
NBask	number of shopping carts
NItems	number of calls of product views
NOrd	number of orders
NRec	number of displayed recommendations
NRecAcc	number of accepted recommendations
NRecItem	number of recommended products
NTrans	number of sessions
ctrl_nBask	number of shopping carts (control group)
ctrl_nItems	number of calls of product views (control group)
ctrl_nOrd	number of orders (control group)
ctrl_nTrans	number of sessions (control group)
ctrl_rev	revenue (control group)
ctrl_rev2	squared revenue (control group)
ctrl_revRecAcc	revenue of accepted recommendations (control group)
inRules	number of valid online result rules after RDE activation
nItemsOrdered	number of ordered products
nProd	number of products Note: nProd is only filled if this configuration parameter is set: *re.analysis.settings.usepreprocessing = true*
path	path
rev	revenue
rev2	squared revenue
revRecAcc	revenue of accepted recommendations

key performance factor	description
text	text
time	time

Table 33: statistical key performance factors

Interactive statistic

Additionally to the statistic worksheets within the client, the RDE Server provides an interactive statistic overview which shows all relevant figures. The site can be found using the button “Interactive Statistic” (Figure 47) or via the following link (authentication via user name and password is required):

http://host:port/rde_server/admin/res/statistic/interactive?locale=en|de

Note: Users need permissions to view the interactive statistic (therefore see Table 2).

This site allows the comparison of all figures from the A/B test, which means the key performance factors of the different groups can be compared directly. For this purpose, there are three main columns: the control group results, the recommendation group results and the difference between the two groups.

The performance factors can be sorted by any column and filtered by year, month, week, day, hour or a defined time span. Additionally, all statistic values can be exported in CSV file format and processed with other programs (e.g. Excel).

The following factors are tracked separately for the control and the recommendations group.	
performance factors	Description
Visits	number of sessions
Recommendations	number of received recommendation queries
Clicked Recommendations	number of clicked recommendations
CTR	click through rate
Baskets	number of shopping carts
Orders	number of executed orders
Order Value	gross order value
Order Value Recomm.	gross order value generated through previously recommended items
Basket / Visit	baskets per session
CRO	orders per session
Avg. Order Value	average gross order value
Value / Visit	average gross order value per session

Table 34: statistical key figures (1)

Order value recomm The statistical key figure "order value recommendation" will be influenced by the property re.statistics.rec.acc.window.size which can be configured within the RDE Client. This parameter defines the maximal number of steps between accepting a recommendation and purchasing the recommended product. Later purchases will not be counted within the statistics.

Direct comparison of control and recommendations group	
performance factors	description
Δ Recommendations	difference in the number of recommendation requests
Δ Baskets	difference in the number of shopping carts
Δ CRO	difference in the number of orders per session
Δ Avg. Order Value absolute	difference in the average absolute gross order values
Δ Avg. Order Value percentage	difference in the average relative gross order values (percentages)
Δ Avg. Value / Visit absolute	difference in the average absolute gross order values per session
Δ Avg. Value / Visit percentage	difference in the average relative gross order values per session (percentages)
Δ Avg. Value / Visit percentage 95% confidence interval	95% confidence interval of delta average order value per session in percent

Table 35: statistical key figures (2)

6.6 Live System

After the successful integration into an online shop and start up, the functionality of the prudsys RDE can be checked via test orders, using the same requests as for the test system (see Chapter 5.11.3). The session-ID of a test order can be used to check the translog files. At the same time, the daily statistic file can be compared to the translog files.

Differences translog / statistics

Please note that the translog files only store learning requests, i.e. product-to-product recommendations. Within the statistic files, also non learning requests are stored. Furthermore only known products, which exist within the master data file, will appear within the translog files. On the contrary recommendations, baskets and orders in combination with unknown products will appear within the statistic files. Therefore, there might be differences between the translog files and the statistics files.

Additionally, the statistics and translog files can be used to check the relation between control and recommendation groups. Within the files, the relation between those groups has to be the same as in the configuration menu in the client.

7 prudsys RDE | Newsletter

7.1 Introduction to RDE | Newsletter Module

RDE | Newsletter

The RDE | Newsletter module automatically generates a newsletter with personalised content for each recipient (e.g. with product recommendations, personalised content, etc.). Base for these recommendations are surfing and purchasing behaviour in the online shop, information about historic shopping baskets and the clicking pattern in the electronic newsletter. The benefits are high newsletter click-through rates, relevant recommendations and increased turnover.

Business scenarios

The prudsys RDE | Newsletter module enables a personalised approach with each individual customer without the need for manual intervention. To this end, both profile information and historical transaction data are analysed. If this information is not available, the system switches to top seller mode, which is continuously optimised in real-time based on the customer's reaction. So the sender can reduce cancellation rates, increase subscriptions and improve purchase conversion. The RDE | Newsletter module can also be used to determine personal content for conventional mailings in paper format.

How it works

The RDE | Newsletter module contains all the functions needed to display personalised recommendations and content as an electronic newsletter and as print mailing. The personalised recommendations are created using the same methods as the RDE | Recommendations module, i.e. by the evaluation of historic transaction data and by real-time learning. When the receiver opens the newsletter, the personalised newsletter content is displayed dynamically and in real-time. The use of intelligent newsletter templates means that the existing mailing system and process can be used furthermore.

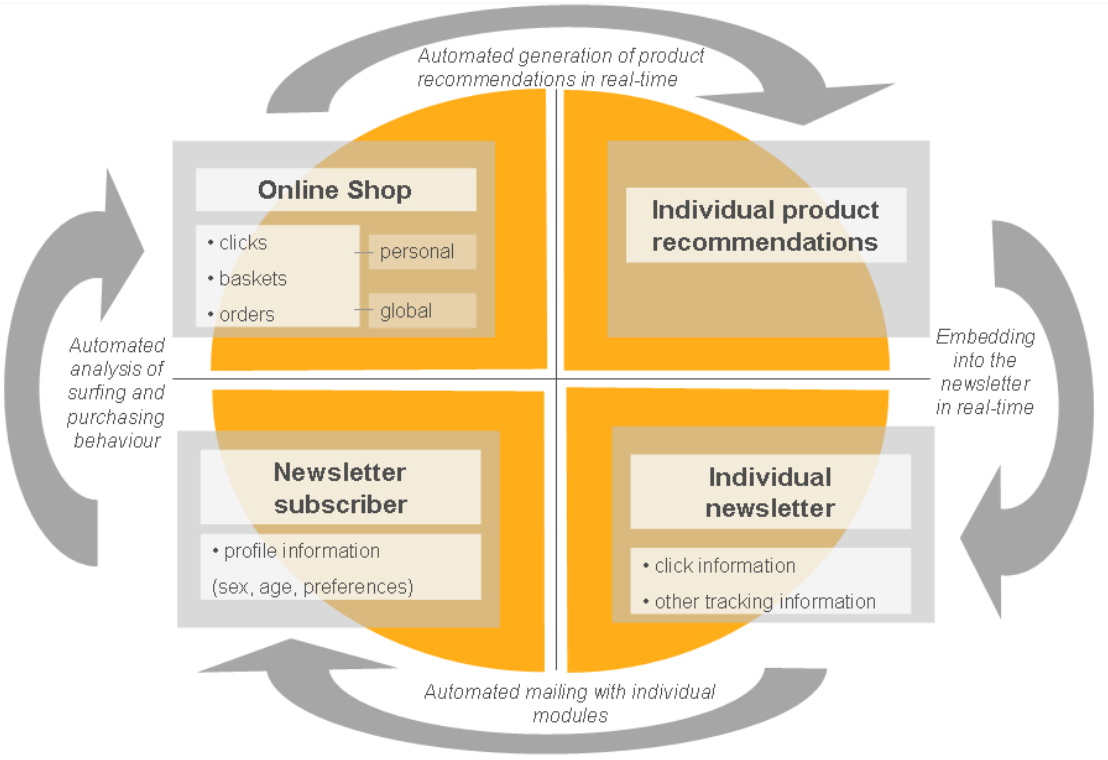


Figure 49: correlation between Recommendations and Newsletter module

Personalized recomms

Personalized recommendations can not only be displayed in the online shop but also within electronic mailings and newsletters. The personalized offers are based on the individual historic orders, the clicking behaviour within the mailing and the continuative clicking behaviour within the online shop of every single recipient. The product recommendations of the prudsys RDE are compared with already displayed product offers and calculated on the basis of the every individual customer profile.

The following image provides an overview of the required data and their processing, provided that both the prudsys RDE | Recommendations for online shops and the prudsys RDE | Newsletter are running simultaneously:

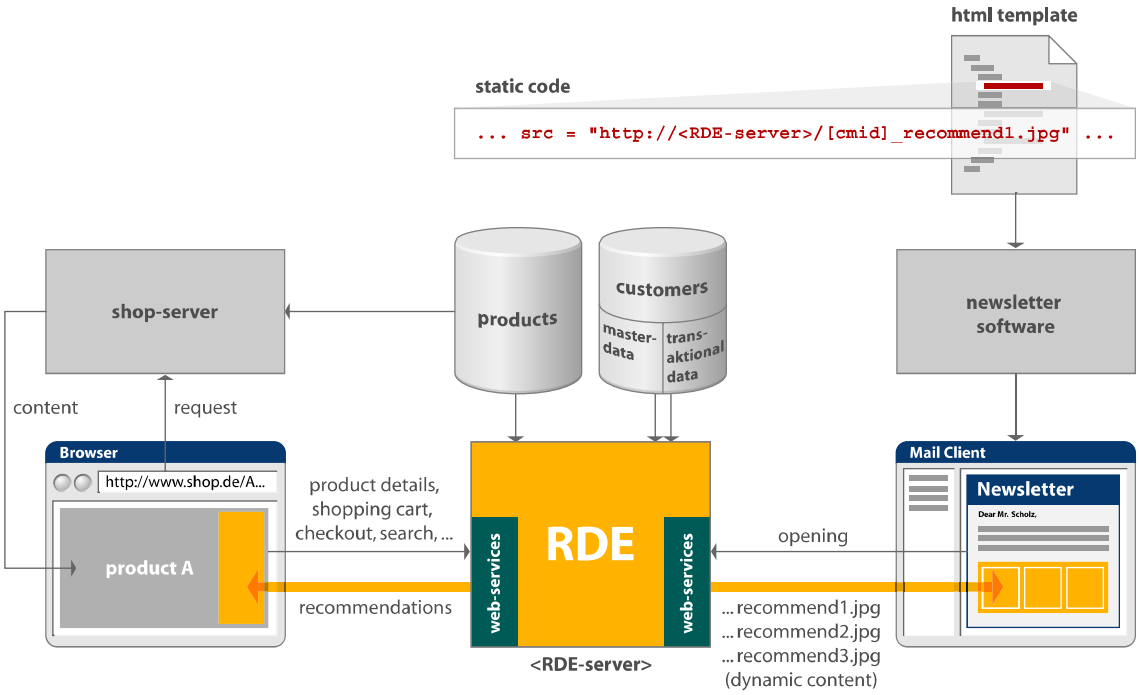


Figure 50: data migration within shop and newsletter RDE

Multipart message

Please note that personalized recommendations can only be displayed in “multipart messages“, i.e. HTML mailings which are able to download content dynamically.

Within the template of the HTML newsletter, predefined requests for the recommendations have to be integrated. The recommendations will be calculated by the prudsys RDE | Newsletter not before the mailing is opened by the recipient. The static request serves as a placeholder who will be replaced by product information like product name, price, image link and product link.

The required email addresses can be extracted from the shop data base, the data warehouse or the data base of the mailing programme and then transferred to the prudsys RDE | Newsletter module.

Redirect

When the electronic mailing or newsletter is opened by the recipient, the customer profile is transferred in real time to the prudsys RDE Server which also calculates and displays the product or content recommendations in real time. The placeholders within the HTML mailing are replaced via redirect. This makes sure that the displayed product information is always up to date. Products which have already been sold out will not be displayed.

The key for the transaction data is always the recipient's email address or the customer-ID if there is a unique customer-ID within all involved systems.

Success tracking

The success of the personalized newsletter recommendations can be measured in the same way as in the online shop. It is necessary to define a control group which receives only static recommendations. The size of the control group is optional, but it is recommended to choose a control group size of at least 10% of all newsletter recipients.

The module prudsys RDE | Newsletter offers reporting functionalities and displays, for example, the key performance indicators "number of email openers" and "number of accepted recommendations".

7.2 Creating an RDE | Newsletter application

After starting the RDE client, a new RDE application has to be created (therefore see Chapter 5.2). In the first step, there is no differentiation between an RDE | Recommendations, an RDE | Newsletter or other applications. The differences between the various products are carried out in further setting steps.

Same or different servers

The RDE | Newsletter application can be installed on the same server as other RDE applications, because they (e.g. the RDE | Recommendations application) will transfer data to the Newsletter application. Alternatively, the RDE | Newsletter application can be installed on a separate server. The necessary steps to install a new server are described within Chapter 2).

7.3 Configuration parameters for RDE | Newsletter

7.3.1 General settings

After the creation of a new RDE | Newsletter application, the following settings have to be carried out via the RDE client menu "Configuration" (therefore see Figure 22).

General settings newsletter application		
Property	Value	Description
module.newsletter.statistics.enable	True | false default: false	True: enable the newsletter statistic
module.newsletter.textimage.targettype	gif | png default: gif	Image file format
newsletter.base.image.url	http://imageserver/ yourshop/images/ ${UID}.png or ${imageURL}	basic image link within the shop: either server path or column within an items.csv file
newsletter.base.link.url	<empty>	basic product link within the shop, will be provided through newsletter templates

General settings newsletter application		
Property	Value	Description
newsletter.base.textimage.filename	UID	file name of the text image: if the hosting is carried out internally, the file will be named after the indicated column (Note: UID corresponds to the column pid)
newsletter.base.textimage.url	<textimageURL>	link to the product text image (intern and extern) example: http://${host}:${port}/rde_server/res/${reid}/newsletter/textImage/${prop_newsletter.filename.column} Special case – column pid: http://${host}:${port}/rde_server/res/${reid}/newsletter/textImage/${**UID**}
newsletter.datafile	items.csv	name of the product data file for creation of the text images within data directory
newsletter.diff.strategy	MEMORY | FLATFILE | MIX	mode for text image generation
newsletter.filename.column	<column_from_ items.csv>	column for text image naming (internal hosting)
newsletter.internal.hosting.textimage	True | false default: true	true: text images will be generated by the RDE false: text images will be provided external
newsletter.imagetemplate	<name>.jrxml or <subsection>:<name>. jrxml	name of the iReport template

Table 36: configuration parameters - newsletter general

Several iReports

Note: There can be more than one newsletter image template, e.g. for different languages. In this case, the templates have to be signed with a unique subsection and separated by a pipe ("|").

Example: There are two image templates for languages German (DE) and English (EN).

DE:newsletter_image_de.jrxml|EN:newsletter_image_en.jrxml

Additional settings for newsletter master data items.csv		
Property	Value	Description
newsletter.csv.charset	UFT-8	character set within items.csv
newsletter.csv.comment	#	character for marking comments
newsletter.csv.delimiter	|	separator
newsletter.csv.hasHeader	true | false	existing / not existing header
newsletter.csv.mode	Excel	mode for reading CSV files
newsletter.csv.recorddelimiter	\n	data set separator, here [Enter]
newsletter.csv.trim	true	spaces will be removed
newsletter.datablocksize	1000	number of pages of a pdf file generated by the report template
newsletter.storagetype	MD5	storage type for text images

Table 37: configuration parameters - newsletter items

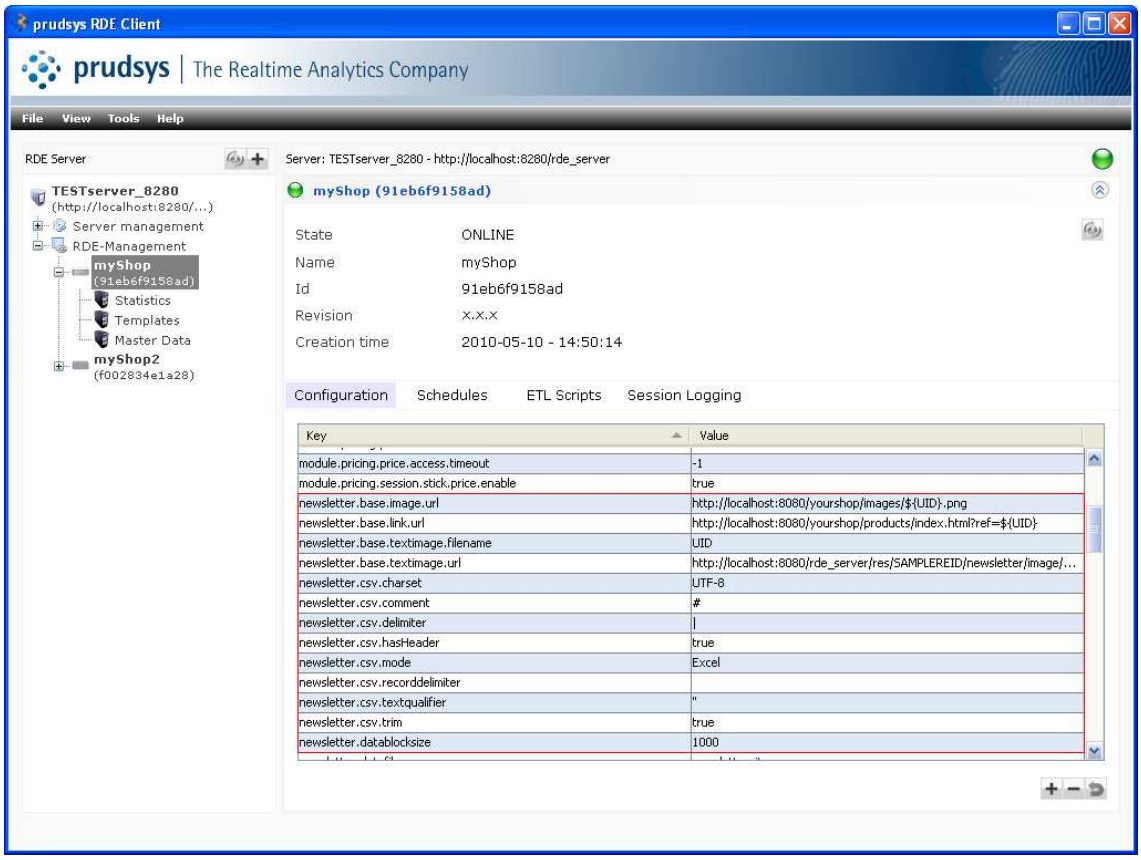


Figure 51: newsletter configuration

7.3.2 Fallback Settings

Automated fallback

If a determined newsletter address is not yet known and can not be identified within the online shop, there will be no or not enough transaction data. In this case, fallback recommendations can be displayed. Fallback recommendations can be calculated automatically or defined manually. This means, certain products are defined that will always be displayed if there are not enough personal recommendations for a specific customer.

For more information about fallback settings, see Chapter 5.8.2.

Manual fallback

The standard default settings allow the definition of up to five manual fallback recommendations. If these are not enough, new settings can be added through a click on the button “plus” (see Figure 52). The denominator should have the same pattern as the standard fallback property

newsletter.fallback.recommendation.<number> (see Figure 53).

Please note: Automated top sellers are stronger than manual fallbacks, i.e. automated top sellers will be always displayed first if they are enabled.

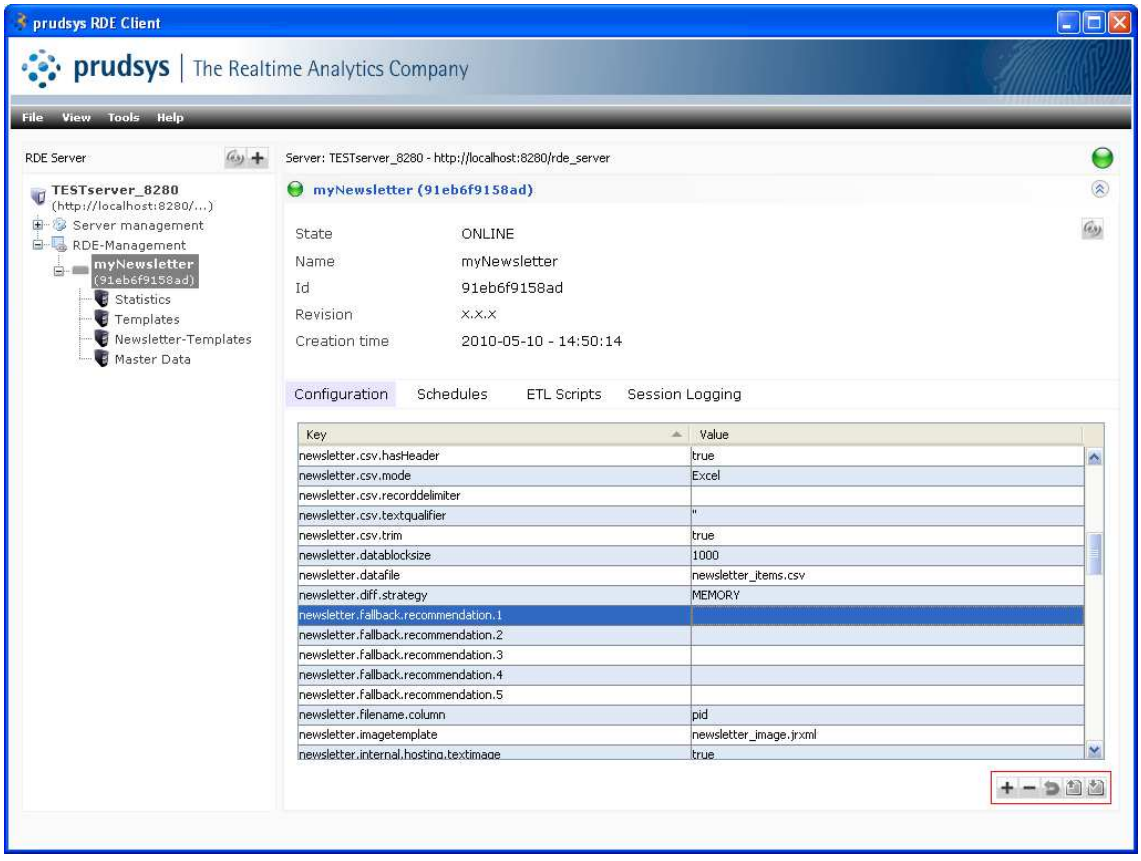


Figure 52: add property for newsletter fallback (1)

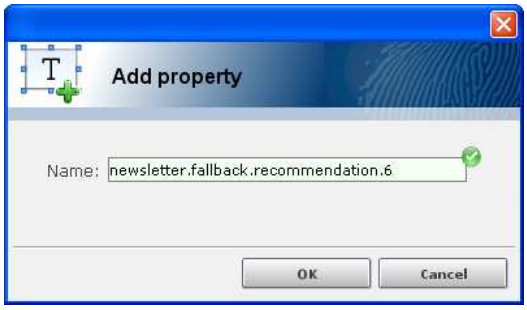


Figure 53: add property for newsletter fallback (2)

7.4 Required data

7.4.1 Using the RDE | Recommendations Data

To be able to calculate and display recommendations, the RDE | Newsletter application needs the product data, online recommendation result rules and transaction data. In most cases, there is already an RDE | Recommendations application. In this case, the necessary data files can be imported directly from the application via HTTP requests.

Product master data

The complete list of all products within the online shop (the same file the RDE | Recommendations application uses, see Chapter 5.5.1):

- items.csv

These columns are required for basic recommendation calculation: product-ID (pid), price (netUnitPrice) and reward. Furthermore, these columns are required for display the recommendations within the newsletter: URL and imageURL.

Online rules

Recommendation result rules which have been calculated through the offline learning process of the RDE | Recommendations application

- resultrules.csv

Please note: In case of using a conditional algorithm (*re.alg.conditional = true*), the data file transprobs.csv has to be uploaded as well.

All files (items.csv, resultrules.csv, transprobs.csv) can be uploaded via upload process within the RDE Client (Chapter 5.6.1) or via Kettle script (Chapter 5.6.2) into the directory

/<RDE_DIR>/data/re_server/<RDE-ID>/data.

Transaction data

The RDE | Newsletter application needs transaction log files to calculate personalized recommendations (Chapter 5.5.5). The transaction log files can be imported from the existing shop application or have to be provided through the customer.

- userproducttransactions.csv (option 1)

or

- translog files (option 2)

Two upload options

There are two options to upload required transaction data to the RDE Server. With option 1, only one HTTP request will be executed. This request collects all necessary files within one run. This option is used favourably for files < 10 MB and periods < 90 days.

With option 2, the files can be uploaded individually. Only latest files will be sent to the server. This option is importantly faster with large files and periods.

Option 1: user product transactions

If an RDE | Recommendations application already exists, the file userproducttransactions.csv can be generated through HTTP requests from the transaction log files (translogs) of the RDE | Recommendations application. If there is not any other application, this file has to be provided by the customer in the required format.

Two import alternatives

All three files (items, resultrules, userproducttransactions) can be imported from the RDE | Recommendations application via update request. There are two different import options.

Note: The RDE | Newsletter application must be deactivated during the execution of the following requests!

Alternative 1:

RDE | Newsletter application is running on the same server as the RDE | Recommendations application

http://host:port/rde_server/admin/res/<NL-RDE-ID>/newsletter/newsletterupdate/run?lastdays=<DAYS>&shop_reid=<SHOP-RDE-ID>

Alternative 2:

RDE | Newsletter application is running on a different server

http://host:port/rde_server/admin/res/<NL-RDE-ID>/newsletter/newsletterupdate/run?server=<IP-SHOP-SERVER>&port=<PORT>&appname=rde_server&shop_reid=<SHOP-RDE-ID>&login=<LOGIN>&password=<PASSWORD>&lastdays=<DAYS>

The call “lastdays“ is not limited and can be set at will, depending on the required date for the transaction data.

Please note: Users need permissions to import the required data (therefore see Table 2).

Option 2: individual import of translog files

The translog files have to be imported to the RDE | Newsletter application. This step is equal for applications running on different or same servers.

To get the translog files from RDE | Recommendations application, the following requests have to be sent via browser to the server:

- List all translog files:

http://host:port/rde_server/admin/res/<NL-RDE-ID>/translogs/

- Get one translog file as zip file:

http://host:port/rde_server/admin/res/<NL-RDE-ID>/translogs/translog.<DATE>

Note: The required format of <DATE> is yyyymmdd.

- Get the latest translog file from the currently running process:

http://host:port/rde_server/admin/res/<NL-RDE-ID>/translogs/dump

- Get all existing translog files as zip files:

http://host:port/rde_server/admin/res/<NL-RDE-ID>/translogs/archive

- Get the translog files of the last <days>:

http://host:port/rde_server/admin/res/<NL-RDE-ID>/translogs/archive ?lastdays=<DAYS>

After downloading the translog files from RDE | Recommendations application, the files can be simply transferred to the RDE | Newsletter application by copy and paste into the translog directory.

Please note: Users need permissions to download translog files (therefore see Table 2).

It is important to set the following settings to calculate newsletter recommendations based on translog files:

re.translog.readTranslogs = true

re.translog.readNLastDays = <amount of last days>

For more information about these translog file settings, see Table 17. To activate these settings, the RDE application has to be reactivated.

7.4.2 Special Data for the RDE | Newsletter application

Real time delivery

In order to provide a proper display of the newsletter, the product description and prices have to be integrated in form of images, which means those images have to be created separately. This creation is necessary because emails only load images when they are opened. But text will not be reloaded. As the personalized recommendations are calculated and delivered in real time (when the corresponding email is opened by the receiver), it is necessary to create those images in real time as well.

In order to create the text images, the master data file has to be uploaded via RDE Client or Kettle file. The product data for the newsletter can be provided separately, but it is also possible to use the file items.csv from RDE | Recommendations application.

Note: Users need permissions to upload image data (therefore see Table 2).

Information within text image

Within the text image, any information can be rendered. If the data is provided separately, the file must contain at least the product-ID and all information which should be displayed within the newsletter. In most cases these information are name, description, price and/or strike out price.

The data could contain the complete catalogue but also specific parts, for example to focus only on seasonal products.

The data described above is used to generate the necessary text images with the help of Jasper-Report. Content, colours and format can be chosen at will. The images will be created on the basis of the template and the data from the file items.csv.

The following image shows a text image example:

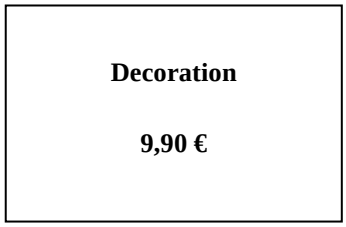


Figure 54: example of a text image

7.4.3 Jasper-Report

A Jasper-Report is a template, which is the basis for generating text images. Per default, a template newletter_image.jrxml is provided together with the application. This template can be adapted at will. Alternatively, new reports can be created.

Report software

In order to create new reports, it is recommended to use the open source software iReport, version 3.5.3 (download at http://jasperforge.org/projects/ireport).

Together with the program, some tutorials are provided which give a good introduction into the creation of templates with iReport.

Finally, the report has to be transferred to the server. Therefore, the new .jrxml file has to be copied into the following directory:

/data/re_server/<NL-RDE-ID>/newsletter

Alternatively, the report can be transferred via curl order:

curl -X POST -T "<directory of the .jrxml file>" --digest -u username:passwort http://host:port/rde_server/admin/res/<NL-RDE-ID>/newsletter/imagetemplate/newsletter_image.jrxml

Please note: Users need permissions to upload the image template (therefore see Table 2).

7.5 Master Data Update

Depending on the required actuality of the master data, it is possible to update the master data in determined time intervals. Transaction data can be transferred to the prudsys RDE Server once per day. As most newsletters are sent more seldom, it is recommended to update the transaction data only weekly or directly before the distribution of a new newsletter campaign.

Additionally, it is possible to define which data form the transaction files should be taken into account for the calculation of personalized recommendation. It is possible to define the number of past days that should be taken into account to avoid that personalized recommendations are based on orders that have been carried out a long time ago. This makes sure that newsletter offers are always based on the most recent recommendation rules.

7.6 Generating Text Images

To create text images, the items.csv is separated into different parts. Through the software Jasper-Report, PDF files will be created. The single pages are rendered into text images, which will be put into the following directory:

/data/re_server/<NL-RDE-ID>/newsletter/images

It is possible to generate GIF or PNG images. The desired option has to be configured within the configuration area (therefore see Figure 22):

module.newsletter.textimage.targettype = gif | png

The default value is GIF.

The PDF files are automatically put into the following directory:

/data/re_server/<NL-RDE-ID>/newsletter/tmp

Generation

In order to execute the HTTP requests, the login data of the server is required.

Please note: Users need permissions to generate text images and control the generation process (therefore see Table 2).

- Start text image creation:

http://host:port/rde_server/admin/res/<NL-RDE-ID>/newsletter/createtextimages/run

- Control process:

http://host:port/rde_server/admin/processes

- Stop process:

http://host:port/rde_server/admin/processes/<PROCESS-ID>/cancel

Note: If the Tomcat is simply shut down, problems will occur at restart. Therefore, the Tomcat should be stopped using “Cancel”.

7.7 RDE | Newsletter templates

The output format of the prudsys RDE | Newsletter application is always text. Product images, text images and product links are delivered as dynamic elements via redirects personally for every addressee when newsletter receiver opens the mailing.

Therefore, newsletter templates are used to define different recommendation types like described within Table 21. They can be created in the same way like shop recommendation templates. A template contains all necessary information needed for the generation of recommendations. The RDE Server is able to parse this information and send corresponding queries to the recommendation engine, as well as to process the recommendation returns and deliver it to the newsletter opener.

A wizard supports the creation of any template in only 3 steps.

7.7.1 Basic settings

These setting options are described in detail within Chapter 5.10.1.

7.7.2 Newsletter settings

Base link URL:

Link URL into the web shop

Text image subsection:

Indicate a subsection if you are using several iReports. This subsection has to be set within the configuration menu („newsletter.imagetemplate“ in Table 36). If there is only one iReport, this field may stay empty.

Channel URL parameter:

Any name can be chosen as channel parameter. The name will be sent to the RDE together with the recommendation request and is required for the newsletter statistics.

Additional URL Parameter:

Any number of parameters can be set. The parameters will be sent to the RDE together with the recommendation request. The RDE ignores those parameters but they will appear in the RDE output for further processing and tracking.

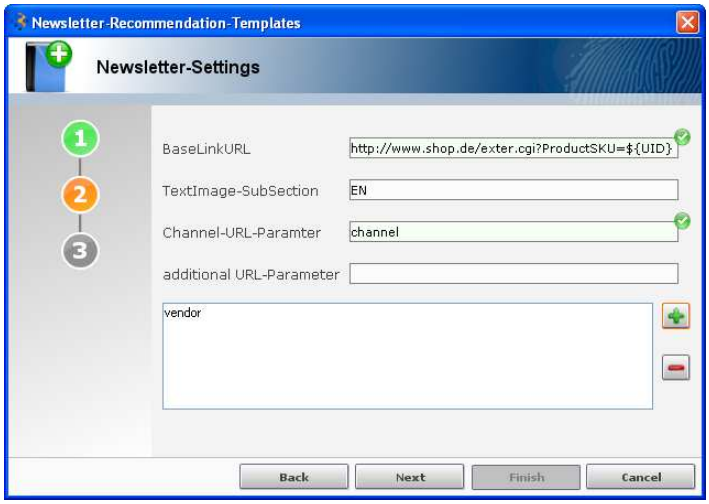


Figure 55: newsletter settings

7.7.3 Add queries

In this final step, recommendation templates will be created. It is described in detail within Chapter 5.10.3

Template URL

If a recommendation template is created with the RDE Client, the corresponding three HTTP requests are displayed within the window (see Figure 56). These requests have to be sent to the RDE Server in order to query the corresponding RDE.

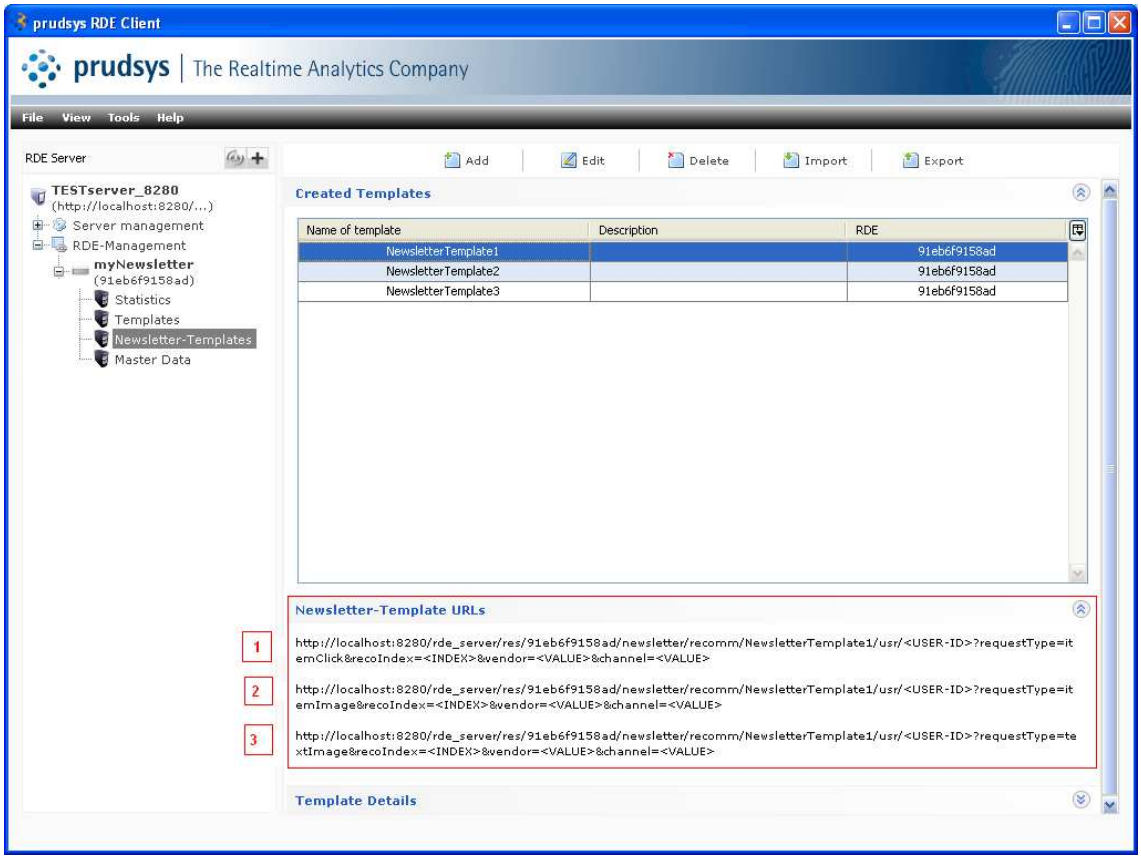


Figure 56: newsletter template URLs

7.8 Requesting Newsletter Recommendations

Before the requests can be tested, the newsletter application has to be activated. Afterwards, the test calls can be sent directly via HTTP requests in order to test the correct functioning of the settings.

Basically, a newsletter recommendation consists of three URLs. They only differ in the parameter “requestType”:

- Product URL

*http://host:port/rde_server/res/<NL-RDE-ID>/newsletter/recomm/<NAME-OF-NEWSLETTER-TEMPLATE>/usr/<USER-ID>?requestType=**itemClick**&recoIndex=<INDEX>&vendor=<VALUE>&channel=<VALUE>*

- Product image

*http://host:port/rde_server/res/<NL-RDE-ID>/newsletter/recomm/<NAME-OF-NEWSLETTER-TEMPLATE>/usr/<USER-ID>?requestType=**itemImage**&recoIndex=<INDEX>&vendor=<VALUE>&channel=<VALUE>*

- Text image

*http://host:port/rde_server/res/<NL-RDE-ID>/newsletter/recomm/<NAME-OF-NEWSLETTER-TEMPLATE>/usr/<USER-ID>?requestType=**textImage**&recoIndex=<INDEX>&vendor=<VALUE>&channel=<VALUE>*

The parameter “recoIndex” describes the index of queried recommendations within the newsletter. The first recommendation has to be signed with *recoIndex=0.*

The parameter “vendor” is an additional parameter and will not be interpreted by the newsletter RDE.

The parameter “channel” is a previously chosen name and can be set at will (therefore see Chapter 7.7.2). The transferred channel is used for statistical analyses (therefore see Chapter 7.10.2).

To test the complete newsletter configuration, the following options should be tested:

Personalized recommendations

- The user is identified and his/her ID is put into the request (user-ID from the file userproducttransactions.csv/translog file). Every user should get different recommendations.
- An unknown user-ID is put into the request. The recipient should receive the fall back recommendations which have been set in advance. At recoIndex = 0, recommendation 1 should appear.

Non-personalized recommendations

- All other recommendation templates work like described in Table 21.

7.9 Integration into the newsletter

7.9.1 Newsletter HTML code

Three links are needed

The HTML newsletter template can be created as usual. It has to be taken into account that the prudsys RDE | Newsletter application does not deliver HTML design elements, but only the three links for the product, the product image and the text image.

The target link within the HTML template will be replaced by the corresponding product URL, image URL and text image URL when the recommendation is requested.

For example, the link to request the product URL within the template

http://host:port/rde_server/res/<NL-RDE-ID>/newsletter/recomm/<NAME-OF-NEWSLETTER-TEMPLATE>/usr/<USER-ID>?requestType=itemClick&recoIndex=<INDEX>&vendor=<VALUE>&channel=<VALUE>

is converted into the following URL (base link URL into the online shop, Chapter 7.7.2) when the newsletter is delivered

http://www.bspShop.de/exter.cgi?ProductSKU=0815-32&AAID=20235&CategoryName=234&vkn=KSHDIH&Bannerreference=CKAT

When this URL is called from within the newsletter, a client-wise redirect to a product detail view is executed.

The code for three recommendations within the template could look like the following example (the RDE differentiates between the recommendations with the help of their positioning):

Newsletter HTML code example

```
<html>
<head>
  <title>My new Newsletter</title>
</head>
<body text="#000000" link="#000000" vlink="#000000"
alink="#000000" leftmargin="0" topmargin="0" marginwidth="0"
marginheight="0">
<table>
<tr>
<td>
  <a href="http://<HOST>:<PORT>/rde_server/res/<NL-RDE-
ID>/newsletter/recomm/<TEMPLATE-NAME>/usr/<USER-
ID>?requestType=itemClick&recoIndex=0&vendor=vendor1&channel=nl">
    <img src="http://<HOST>:<PORT>/rde_server/res/<NL-RDE-
ID>/newsletter/recomm/<TEMPLATE-NAME>/usr/<USER-
ID>?requestType=itemImage&recoIndex=0&vendor=vendor1&channel=nl">
<br>
```

```
<img src="http://<HOST>:<PORT>/rde_server/res/<NL-RDE-
ID>/newsletter/recomm/<TEMPLATE-NAME>/usr/<USER-
ID>?requestType=textImage&recoIndex=0&vendor=vendor1&channel=nl">

  </a></td>

<td>

  <a href="http://<HOST>:<PORT>/rde_server/res/<NL-RDE-
ID>/newsletter/recomm/<TEMPLATE-NAME>/usr/<USER-
ID>?requestType=itemClick&recoIndex=1&vendor=vendor1&channel=nl">

    <img src="http://<HOST>:<PORT>/rde_server/res/<NL-RDE-
ID>/newsletter/recomm/<TEMPLATE-NAME>/usr/<USER-
ID>?requestType=itemImage&recoIndex=1&vendor=vendor1&channel=nl">
<br>

    <img src="http://<HOST>:<PORT>/rde_server/res/<NL-RDE-
ID>/newsletter/recomm/<TEMPLATE-NAME>/usr/<USER-
ID>?requestType=textImage&recoIndex=1&vendor=vendor1&channel=nl">

  </a></td>

<td>

  <a href="http://<HOST>:<PORT>/rde_server/res/<NL-RDE-
ID>/newsletter/recomm/<TEMPLATE-NAME>/usr/<USER-
ID>?requestType=itemClick&recoIndex=2&vendor=vendor1&channel=nl">

    <img src="http://<HOST>:<PORT>/rde_server/res/<NL-RDE-
ID>/newsletter/recomm/<TEMPLATE-NAME>/usr/<USER-
ID>?requestType=itemImage&recoIndex=2&vendor=vendor1&channel=nl">
<br>

    <img src="http://<HOST>:<PORT>/rde_server/res/<NL-RDE-
ID>/newsletter/recomm/<TEMPLATE-NAME>/usr/<USER-
ID>?requestType=textImage&recoIndex=2&vendor=vendor1&channel=nl">

  </a></td></tr>

</table>

</body>

</html>
```

7.9.2 Sample output

The following example shows how a newsletter can be built:

Figure 57: Lufthansa world shop newsletter

7.10 Newsletter statistic

7.10.1 Newsletter statistic synchronisation schedule

To update the newsletter statistics, a scheduler (Cron Job) that imports the translog files from the web shop has to be defined. The translog files from the shop will be compared to the newsletter transactions and accordingly analyzed.

One basic condition for a correct analysis of the newsletter recipients' behaviour within the web shop is the use of the following special events (therefore see Chapter 5.12.2):

- User to session
- Channel to session

These events provide the RDE with the user identity and channel information that is necessary for correct statistical analysis and reporting. The following property has to be set to save these events correctly:

module.newsletter.statistics.enable = true

The Cron Jobs can be defined within the menu „Schedules".

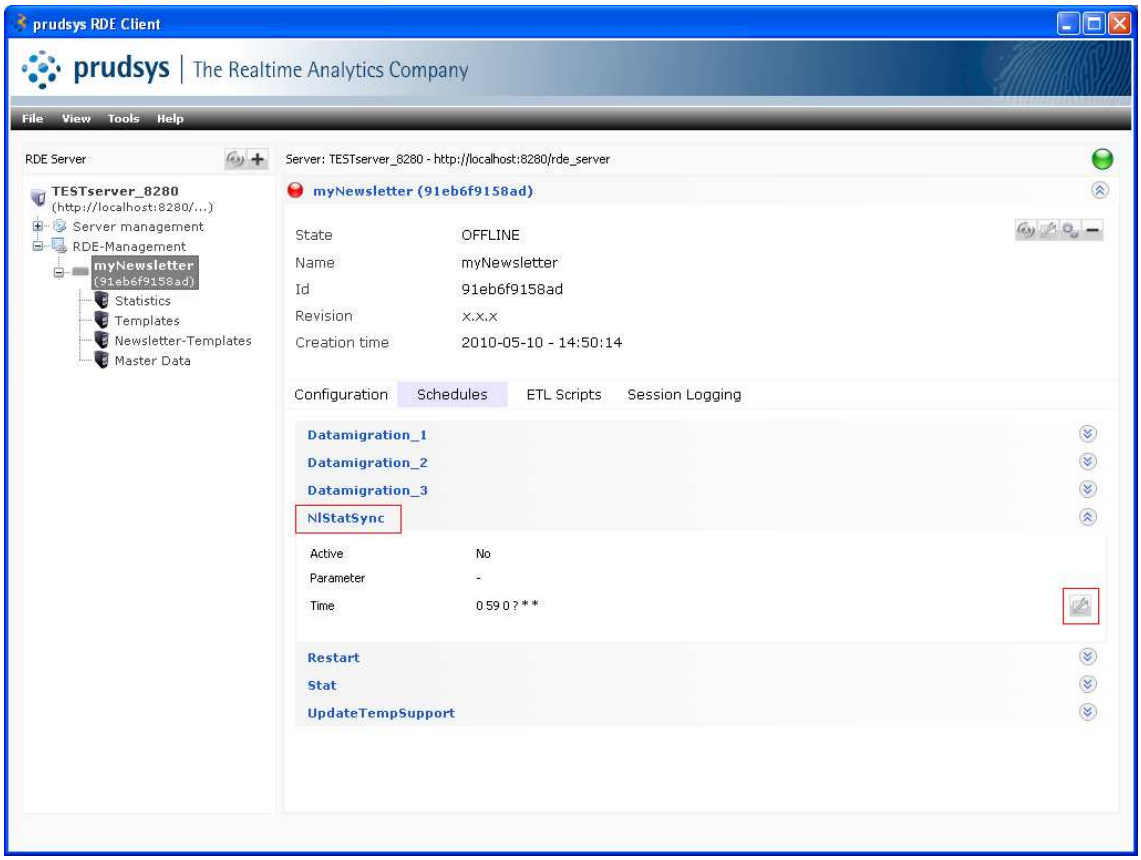


Figure 58: newsletter statistic schedule

The following parameters have to be set:

Job-Name

Any unique name.

Parameter

Path of the corresponding web shop RDE which provides the translog files.

The following syntax has to be used:

<user>@<protocol>:<translog-DIR|NLSyncResource>

- User: The userID needs to exist within the newsletter RDE as well as in the web shop RDE. It is necessary that the user is provided with all rights (like for example the user “system”).
- Protocol: HTTP (8080), AJP (8009) or local
- Indicate a translog directory (within the local file system) or the newsletter synchronization resource of the shop RDE.

Example:

system@local://<RDE-DIR>/rde/data/re_server/<RDE-ID>/translog

system@http://host:port/rde_server/admin/res/<RDE-ID>/newsletter/sync

Activate schedule

Active schedule: Yes/No

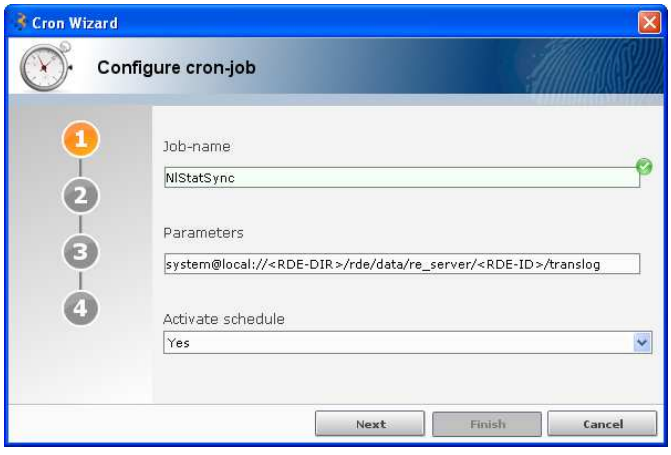


Figure 59: configure cron job for newsletter statistic

Time base

- Every (second, minute, hour, day, week, month) or once
- Time and date

Note: The Cron Job should be carried out shortly after the (in most cases daily) restart of the web shop RDE, as the translog files of the RDE are stored during restart. This is to make sure that the newsletter statistics is based on the most recent translog files.

Execution monitoring

After the successful execution of the newsletter statistics cron job a new line is created within the configuration menu:

module.newsletter.sync.lastFileName = translog.<LAST_DATE>

LAST_DATE refers to the date of the most recent RDE shop translog file available.

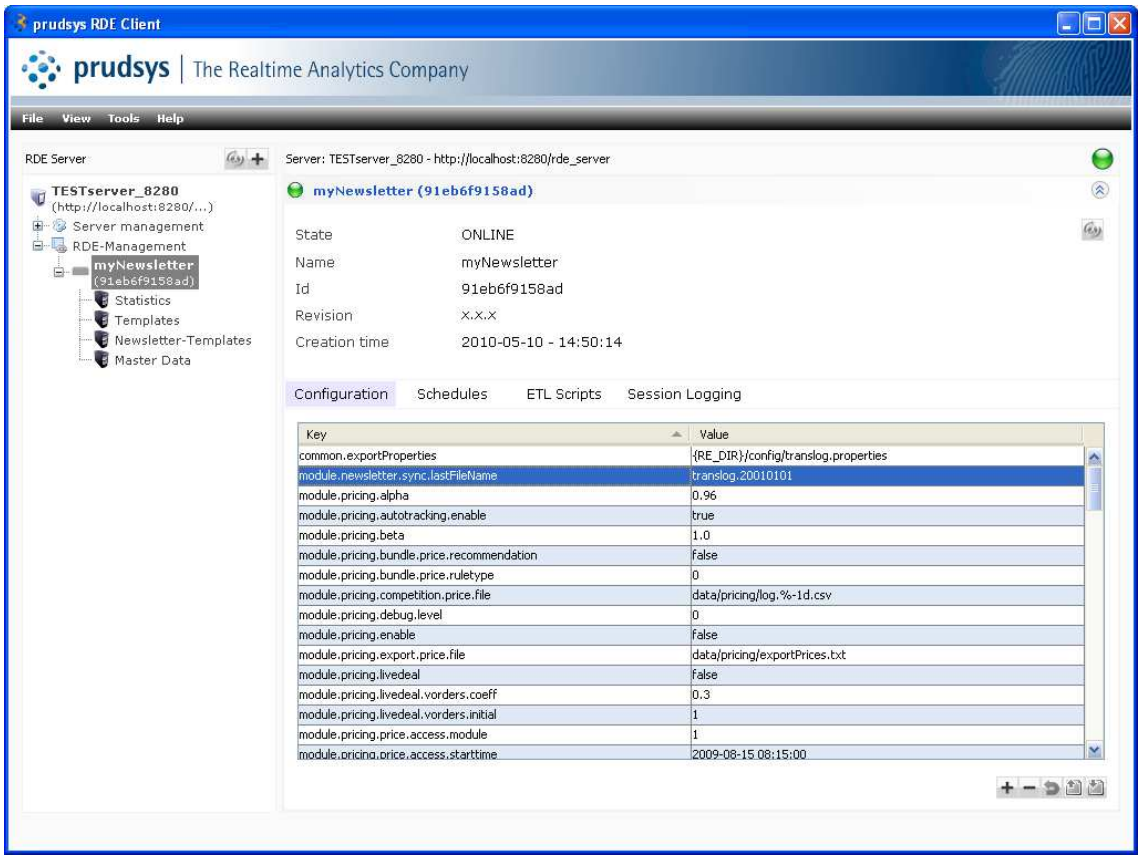


Figure 60: newsletter synchronisation

7.10.2 Statistic monitoring

The RDE Server provides an interactive statistic overview which shows all relevant key figures. The site can be found using the following link (authentication via user name and password is required):

http://host:port/rde_server/admin/res/newsletter/statistic/interactive?locale=en|de

Note: Users need permissions to view the interactive statistic (therefore see Table 2). Additionally, the configuration property has to be set:

module.newsletter.statistics.enable = true

This site allows the monitoring of all key figures. The statistic can be sorted by any column and filtered by year, month, week, day, hour or a defined time span. Additionally, all statistical values can be exported in CSV file format and processed with other programs (e.g. Excel).

Key figure	Description
channel	channel parameter that is sent together with the newsletter recommendation
nItemsViewed	number of newsletter openings
nItemsClicked	number of clicked recommendations within the newsletter
nBaskets	number of baskets created by newsletter recipients
nOrders	number of orders created by newsletter recipients
reward	total sales volume created by newsletter recipients
nItemsOrdered	total number of purchased products by newsletter recipients
nItemsOrderedRecomm	number of purchased newsletter recommendations
nBasketsRecomm	number of baskets which contain newsletter recommendations
nOrdersRecomm	number of orders which contain newsletter recommendations
rewardRecomm	total sales generated by newsletter recommendations only

Table 38: statistical key figures

8 prudsys RDE | Pricing

8.1 Introduction to RDE | Pricing Module

The prudsys RDE | Pricing module is the world's first tool for dynamic price setting and optimisation. It automates optimum pricing and includes product profitability calculation and sales control. The module will continuously update product prices, batch-wise or in real-time, to match changes in market and business conditions.

Your benefits

Your benefits are maximisation of earnings taking into account price and sales volume as well as optimized exhaustion of profit margins and price testing.

Business scenarios

The prudsys RDE | Pricing module enables real-time price optimisation on the basis of user acceptance at all touch points. Particularly in the web shop, where customer reactions to changed prices can be measured immediately, dynamic price optimisation leads to optimal use of the profit margin. In addition, bargain and white label shops are ideal for using dynamic pricing. Other possible uses come up in the area of modern scales and digital price tags in high street shopping as well as when it comes to price search engines.

Each product has its price. Today, this is only partly true. Rigidly maintaining a particular price over a long time period can make inroads into a retailer‘s earnings. Solutions employing algorithms enable a business to vary its prices continuously and to test price acceptance in real-time.

The following figure shows an exemplary business scenario.

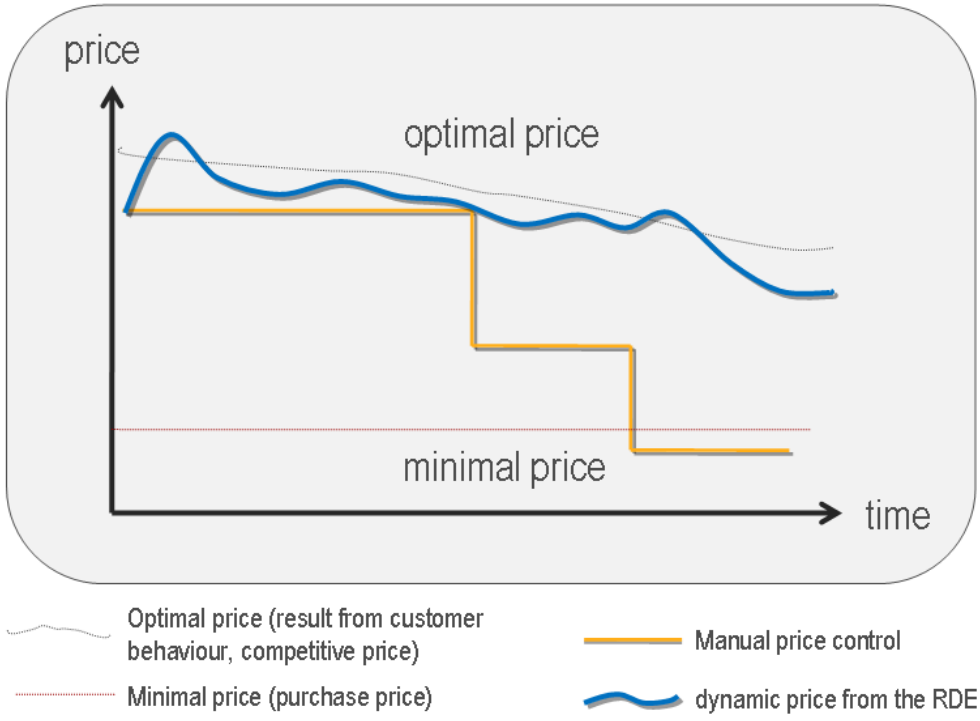


Figure 61: exemplary trend of price

Price optimum

prudsys RDE finds the optimum product price for a particular time period and also take into account the influence of numerous other factors. This method ensures that prices are adjusted in continuation in real-time to take into account the continuous changes in market and business conditions.

These are the most frequent factors influencing price adjustments:

- Customer purchasing behaviour
- Sales skills
- Value patterns for fresh products
- Product availability

Dynamic price optimisation is an ideal method for maximising turnover and earnings. You can also maximise more complex references. Prices which have remained practically unchanged for a long time can now be varied in a planned and targeted manner in order to provide valuable information about price elasticity.

Module functionality

The prudsys RDE | Pricing module analyses a variety of data including information about product value patterns, sales ranking, customer assessments and shelf availability to arrive at optimum product price. The module defines the optimum price for the current product and then creates separate price variations designed to increase turnover. Numerous parameters such as variance, upper price limit, lower price limit and time intervals between price changes can be defined by retailers themselves.

Dynamic price optimisation can apply several methods. The prudsys RDE | Pricing module is mainly based on reinforcement learning models and click-buy behaviour.

Re-inforcement learning

The reinforcement learning process receives information about the current status and its reward from the environment. On the basis of this information the process decides on what action (in this case increase or reduce price) to take next. The environment reacts to this action and the cycle begins again. Over time the process gathers more and more information about the environment.

Click-buy behaviour

Click-buy behaviour is the basis for changing the price. prudsys RDE records anonymous data regarding to clicks, changes to shopping carts and purchases. This method is based on the following assumption: if the willingness to pay does not change, the click-buy ratio also will remain constant. On the contrary changes to this ratio are considered to be an indication of a shift in the customer‘s willingness to pay. The method uses this change to adjust the price. This price adjustment can take place in real time, i.e. the prices are adjusted after every ratio change.

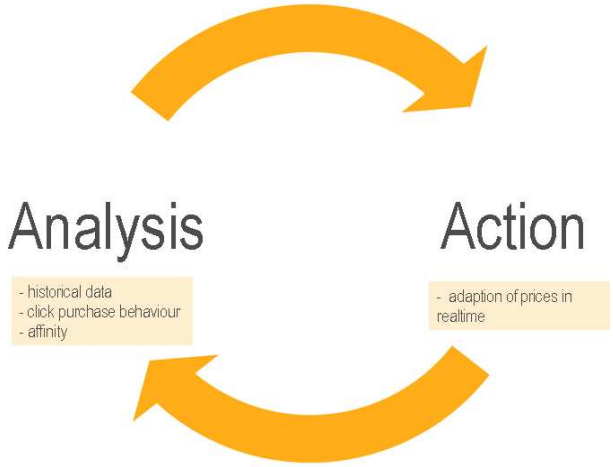


Figure 62: logic of price optimisation

8.2 Creating an RDE | Pricing application

An RDE application exists

If the RDE | Recommendations module is installed and an RDE application already exists, there is no need for a new creation. The RDE | Pricing module uses the same data files.

An RDE application does not exist

If there is no RDE application, a new one has to be created (therefore see Chapter 5.2). In the first step, there is no differentiation between an RDE | Recommendations, an RDE | Newsletter or other applications. The differences between the various products are carried out in further configuration steps.

8.3 Required data

Product master data

The prudsys RDE | Pricing module is able to use the following information which are defined within the columns of the items.csv file. For more information about product master data see Chapter 5.5.1.

Name	Description
pid	ID of the product
netUnitPrice	Sales price
reward	Reward
netPurchasePrice	Purchase price
reducedPrice	Reduced price, used for control group
minUnitPrice	The minimum product price
maxUnitPrice	The maximum product price
stock	Products in stock; will be calculated in real time: (quantity – order), there will be no recommendations for stock = 0

Table 39: important information for RDE | Pricing module

Please note: If there is no single reward data, the column netUnitPrice should be duplicated.

Obligatory data

In order to run the RDE application, the minimum required data is the product-ID and the reward. In order to use price bundles, the minimum required data is the netPurchasePrice.

Price limits

However, for the RDE | Pricing module price limits (columns maxUnitPrice and minUnitPrice) should be defined because otherwise the RDE could set new price limits to an unrealistic high level through its internal mechanisms of dynamic testing.

Price boundary types

The effect of the columns minUnitPrice and maxUnitPrice depends on the price boundary type defined by the configuration parameter priceBoundaryType. For the discount price boundary type, minUnitPrice defines the minimum and maxUnitPrice the maximum discount in percent, respectively. For the relative price boundary type, minUnitPrice defines the minimum price as percentage to the sales price and maxUnitPrice the maximum price as percentage of the sales price. The absolute price boundary type works similar, but specifies the total price boundaries.

The file has to be uploaded like described in RDE | Recommendations Manual, Chapter 5.6.

Competitive prices

In order to include competitive prices into the optimisation process, a file with the following columns has to be provided within the directory /re_<RDE-ID>/data/pricing.

Name	Description
pid	product-ID that belongs to items.csv
price	competitive price

Table 40: competitor‘s price file

As a consequence, the optimized prices will be additionally adapted to the competitive prices.

8.4 Configuration parameters for RDE | Pricing

All necessary configuration parameters for using the RDE | Pricing module can be set comfortably using the menu “Configuration” (Figure 22).

Please note: To activate the settings, the RDE application has to be deactivated and activated again (therefore see Chapter 5.3).

Pricing module – unit price optimisation		
Property	Value	Description
module.pricing.usePriceOptimization	True | false default: false	True: Enable module pricing
module.pricing.invokePriceOptTracking	True | false default: true	True: product clicks and purchases will be tracked automatically (if product recommendations exist) False: product clicks have to be tracked manually (if no product recommendations exist)
module.pricing.priceBoundaryType	0 | 1 | 2 default: 1	The type of price boundaries specified by the columns minUnitPrice and maxUnitPrice 0 – discount type (percentage) 1 – relative type (percentage) 2 – absolute type (recommended)
module.pricing.competPriceFile	data/pricing/ log.%-1d.csv	The path to the competition price file (keys are product-IDs)
module.pricing.exportPriceFile	data/pricing/ exportPrices.txt	The path to the pricing export file which will be stored during RDE deactivation process; comprised old price and optimised price
module.pricing.sessionStickPrice	True | false default: true	True: prices do not change within one session False: real time pricing, every click and order may change the price
module.pricing.priceAccessModule	[1..∞] default: 1	Price change after x clicks and purchases
module.pricing.priceAccessStartTime	<time>	Initial start time for price changes with format “yyyy-mm-dd hh:mm:ss”
module.pricing.priceAccessTimeout	-1 | [0..∞] default: -1	-1: disabled Otherwise: period of time in seconds to the next price change Note: if this property is enabled, clicks and purchases won't change prices until the RDE will be deactivated

Pricing module – unit price optimisation		
Property	Value	Description
module.pricing.pricingAgentName	<agent name>	Name of the agent to use or <empty> for default agent. Price optimisation: - ClickOrderAgent (default) - ClickOrderRegAgent Bundle optimisation: - DiscountAgent (default) - BundleDiscountAgent If not specified, the RDE will automatically select the right pricing algorithm depending on the price optimization type (single product, bundle, live deal, etc.). However, there are different algorithms available for each optimization type and the desired algorithm can be specified here. If the algorithm does not match the optimization type, an error message will be displayed during the activation of the RDE.
module.pricing.alphaP0	[0..1] default: 0.96	Expert setting: how clicks have an effect on price changes; the higher the stronger
module.pricing.betaP0	[0..1] default: 1.0	Expert setting: how purchases have an effect on price changes; the higher the stronger
module.pricing.debug	0 | 1 | 2 default: 0	0 – debugging disabled 1 – default debug level 2 – extensive debug level

Table 41: configuration parameters for unit price optimisation

Besides the unit price optimisation, prudsys RDE provides extended functionalities: price bundles and live deal. Therefore, the following properties can be activated.

Price bundle purposes

Price bundles serve two purposes:

- First: to recommend a bundled product to a specified product
- Second: to calculate a discount for the bundle. The discount is applied to the bundled product only, so the price bundling can be especially used for products with fixed prices in order to apply a virtual price optimization. Note that even when all product prices are fixed, bundle recommendations may be a useful recommendation type.

Pricing module - price bundles		
Property	Value	Description
module.pricing.bundlePriceRecommendation	True | false default: false	True: enable price bundle algorithm Note: the property module.pricing.usePriceOptimization has to be set to “true”
module.pricing.bundlePriceRuleType	-1 | [0..∞][*] default: -1	Default rule type of the price bundle algorithm -1: use all rules *: enable rule hash, i.e. bundle rules will be loaded at start of RDE and not change; otherwise rules are selected dynamically Example: 3* (rule type 3 used at the start of RDE to construct all bundle rules)

Table 42: configuration parameters for price bundles

Please note: The minimum required data for price bundles is the column netPurchasePrice.

Live deal

Product offers are called live deal if there are limited for a determined period of time. It is a special form of openly promoted sales. The price adapts according to the variable consumer demand.

Pricing module - live deal		
Property	Value	Description
module.pricing.liveDeal	True | false default: false	True: enable live deal algorithm Note: the property module.pricing.usePriceOptimization has to be set to “true”
module.pricing.liveDealVOrdersInitial	[1..∞] default: 1	Number of virtual purchases in the first hour
module.pricing.liveDealVOrdersCoeff	[0..∞] default: 0.3	Rise of virtual purchases per further hours

Table 43: configuration parameters for live deal

A/B testing

A/B testing functionalities can also be used for price recommendations. In this case, the control group is usually given the standard price or, if applicable, manually reduced prices. The reduced prices have to be defined within the column reducedPrice (items.csv).

For more information about A/B testing, see Chapter 5.8.3.

8.5 Integration of price recommendations

Two options of integration

There are two options to integrate price recommendations into online shops: either product recommendations already exist or not.

8.5.1 Price recommendation combined with product recommendation

If the RDE | Recommendations module is installed and an RDE application already exists, the price recommendations will be tracked by the request of recommendation templates.

Therefore the following property should be set:

module.pricing.invokePriceOptTracking = true

Note: To activate the setting, the RDE should be restarted.

To get price recommendations with auto tracking, the following requests should be sent to the RDE Server via web browser.

Changing prices with recomms

Step 1: Request the current price

The HTTP request for retrieving the current price of a single product:

http://host:port/rde_server/res/<RDE-ID>/pricing/recomm/sid/<SESSION-ID>?itemid=<ITEM-ID>

The HTTP request of a price bundle of a single product:

http://host:port/rde_server/res/<RDE-ID>/pricing/recomm/bundle/sid/<SESSION-ID>?itemid=<ITEM-ID>

Step 2: Request product recommendations and orders

The product-to-product recommendation template URL has to be sent to the RDE Server to get product recommendations (for more information about product recommendations, see Chapter 5.10.4). This request is a click on a specific product, so the RDE can track this event automatically.

http://host:port/rde_server/res/<RDE-ID>/recomm/<NAME-OF-TEMPLATE>/sid/<SESSION-ID>?itemid=<ITEM-ID>

Just as the recommendation event, the order request will be tracked automatically.

http://host:port/rde_server/res/<RDE-ID>/event/order/sid/<SESSION-ID>?itemids=<ITEM-ID_1>,<ITEM-ID_N>&quantities=<QUANTITY_1>,<QUANTITY_N>

Step 3: Result

In general, the price should be changed if step 1 is repeated.

8.5.2 Independent price recommendation

Naturally there is no need to track product clicks for price recommendations. But in the case with no product recommendation, it is necessary to track the products manually.

Therefore the following property should be set:

module.pricing.invokePriceOptTracking = false

Please note: To activate the setting, the RDE should be restarted.

To get price recommendations without auto tracking, the following requests should be sent to the RDE Server via web browser.

Changing prices without recomms

Step 1: Request the current price

This step is similar to the price request with auto tracking:

http://host:port/rde_server/res/<RDE-ID>/pricing/recomm/sid/<SESSION-ID>?itemid=<ITEM-ID>

Step 2: Click and order request

Click request for tracking the product:

http://host:port/rde_server/res/<RDE-ID>/pricing/event/click/sid/<SESSION-ID>?itemid=<ITEM-ID>

Order request for tracking the product:

http://host:port/rde_server/res/<RDE-ID>/pricing/event/order/sid/<SESSION-ID>?itemid=<ITEM-ID>&quantity=<QUANTITY>

Step 3: Result

In general, the price should be changed if step 1 is repeated.

9 prudsys RDE | Assortment Planning

The RDE | Assortment Planning module automates the planning process directly in the process pipeline. It provides process-linked forecasts and recommendations about goods and product quantities.

Your benefit

Dynamic adjustment of sales plans, accurate assessment of action effects and cost-optimised availability of goods.

Business scenarios

The module prudsys RDE | Assortment Planning can be used for precise forecasting of demand and sales quantities for new and listed individual products. Forecasts also ensure that goods are available at the optimum price, that inventories can be dynamically adapted to marketing plans and that the effectiveness of promotions can be assessed more accurately (advertising campaigns). By integrating the module into the real-time environment of the prudsys RDE, changes in sales patterns are quickly detected and the prognosis models are continuously and fully automatically adapted.

How it works

The RDE | Assortment Planning module uses historic ordering and sales data and takes into account a variety of real-time parameters (e.g. current ordering and sales information) to make valid forecasts about current or future planning values.

An RDE | Assortment Planning agent aims at solving assortment planning tasks in stationary trade. The main focus is calculation of sales volumes forecasts and, based on these results and other additional information like out-of-stock cost rates and product master data, determining the optimal order size or the optimal order time, respectively, for each individual product. Determining the optimal order size in case of fast-moving goods yields a planning process that maximizes the contribution margin of each individual product in the assortment and at the same time avoids out-of-stock situations. For slow-moving items the optimal order time, i.e. the optimal number of products in stock, is calculated by minimizing the inventory costs and out-of-stock costs and simultaneously avoiding out-of-stock situations.

The first step is the calculation of the forecast sales volumes for next periods for both types of output. This prediction provides the basis for further calculations like the optimal order size and is the basic output type the dynamic RDE | Assortment Planning package delivers. The offline agent simply takes historical sales volumes and calculates the forecasts depending on this data. These calculations can be performed by using weighted linear least squares regression techniques up to more sophisticated approximation methods, depending on the chosen agent. The historical sales volumes are the only information needed for determining these forecasts.

Offline Assortment Planning

Providing the sales volumes on a daily basis, an offline RDE | Assortment Planning agent will analyze them up to a certain number of past weeks defined by the user. The agent aggregates the daily data to be able to recognize the overall trend and determining future sales volumes. It disaggregates them accordingly to deliver forecasts for a user defined number of days examining the patterns of the distribution of sales volumes over the specific working days of one week.

Online Assortment Planning

The online RDE | Assortment Planning agent takes the distribution of sales volumes over the working days of a week for each product into account by establishing a forecast model for each separate working day.

If the user is able to provide more detailed information about the individual products like the purchase price, the sales price, the residual price, out-of-stock cost rates or even packaging measurements, the RDE | Assortment Planning agent can obtain further important quantities. For example, the optimal order size can be obtained by maximizing the expected contribution margin and the optimal order time in case of

slow-moving items is obtained by minimizing the overall costs accounting for out-of-stock costs and inventory costs. In general, the former optimization problem can be solved analytically involving quantities like the out-of-stock cost rate and the weighted average cost of capital besides the several types of price for a specific product. The latter is solved by numerical simulation since the desired quantities can not be obtained from the theoretical model explicitly.

The online variant of the RDE | Assortment Planning agent is able to perform all these calculations without the need of providing historical data once an adequate model has been learned.

Shelf allocation

If, additionally, shelf measurements of the local shop can be provided, the RDE | Assortment Planning agent is able to determine an optimal shelf allocation or even an optimal assortment in case of limited amount of space available on the shelves.

9.1 Configuration parameters for RDE | Assortment Planning

All necessary configuration parameters for using the RDE | Assortment Planning module can be set comfortably using the menu “Configuration” (Figure 22: configuration - overview) within the client. To activate the settings, the RDE application has to be deactivated and activated again.

Assortment Planning		
Property	Value	Description
module.disposition.useDisposition	true | false default: false	True: Enables the module RDE | Assortment Planning.
module.disposition.invokeDispositionTracking	true | false default: false	True: Enables the tracking for RDE | Assortment Planning. False: If the tracking is disabled, it has to be arranged manually.
module.disposition.dispositionAgentName	<agent name>	Name of the agent to use or <empty> for default agent

Table 44: configuration parameters for RDE | Assortment Planning

9.2 Required data

Learning method

The RDE Sever requires the following information about product transactions to complete the Assortment Planning learning method:

- time stamp
- product ID
- number of sold units

Processing method

Required data for processing the Assortment Planning module has to be provided within the product master data. For more information see Chapter 5.5.1.

Essential data for calculation of estimated distribution:

- product ID

Essential data for calculation of optimized purchase order quantity:

- product ID
- netUnitPrice
- netPurchasePrice
- residualPrice
- oosCost

9.3 Requesting Assortment Planning functionality

Execution

Assortment planning information for a single product can be requested via browser:

http://host:port/rde_server/res/<RDE-ID>/disposition/sid/<SESSION-ID>
?itemid=<ITEMID>?type=<TYPE>

The parameter type can reach one of the following values:

- 0: estimated distribution within the next seven days
- 1: optimized purchase order quantity within the next seven days

Manual tracking

Additionally, there is a possibility to track products manually if the automatic tracking is disabled (*module.disposition.invokeDispositionTracking=false*):

http://host:port/rde_server/res/<RDE-ID>/disposition/event/order/
sid/<SESSION-ID>?itemid=<ITEMID>&quantity=<QUANTITY>&price=<PRICE>

10 prudsys RDE | Data Cleansing

10.1 Introduction to RDE | Data Cleansing Module

RDE | Data Cleansing

The prudsys RDE | Data Cleansing module functions include real-time duplicate analysis during the consolidation of master data; this can be done in the batch mode if required. Other functions include product variant creation.

Your benefit is a more effective data cleansing and data merge.

How it works

This module performs pre-processing, real-time processing and post-processing using a unique combination of classic ETL technology combined with high-performance real-time Data Mining. With the aid of a visual designer you can create your own process sequences and link the results directly to your existing processes.

Use cases

The module covers the following use cases:

- Similarity search

Detection and, if required, deletion or merging, of similar data sets (e.g. similar products).

- Duplicate search

Duplicates are deleted in local batch mode (e.g. address data sets).

- Duplicate search in real time

The online duplicate search is called Duplicate Service. The basic functionality is the same compared to the local duplicate search, but additional functionalities allow the detection of duplicates in real time (e.g. for the detection of duplicates or similarities within login data in online shops to avoid frauds).

Handling

Note: Currently, the module RDE | Data Cleansing is handled via command line interface only. Therefore, curl syntax is used for the following descriptions.

10.2 Creating an RDE | Data Cleansing application

If an RDE application already exists, there is no need for a new creation. The RDE | Data Cleansing module uses the same data files.

If there is no RDE application, a new one has to be created via RDE Client (therefore see Chapter 5.2) or with curl:

curl -X POST --digest -u username:password
http://host:port/rde_server/admin/re_list/create?reid=<RDE-ID>

RDE-ID can be any alpha numeric string with maximal 12 characters.

10.3 Master data upload

Every duplicate search type requires the same master data import process.

Master data formats

Possible master data import file formats:

- csv
- jdbc

One file may contain an unlimited number of columns. However, all columns have to be defined within the search configuration file.

List of existing files:

curl -X GET --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/data

Output of a file

curl -X GET --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/data/<FILENAME>

Creation or update of a master data file

curl -X PUT -T "<INPUT_FILE>" --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/data/<INPUT_FILE>

The file will be stored in the following directoy:

<RDE-DIR>/data/re_server/<RDE-ID>/datacleansing/data

Deletion of an existing file

A file will be removed from the directory <RDE-DIR>/data/re_server/<RDE-ID>/datacleansing/data.

curl -X DELETE --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/data/<FILENAME>

10.4 Search configuration

10.4.1 Similarity and duplicate search configuration

This chapter describes all necessary settings for the duplicate search. The settings vary depending on the search method which will be used.

- source file settings
- output settings
- algorithm parameters
- statistic settings
- search rules and their weighting

Every search has to be set individually according to your input file.

Configuration parameter	Description
Input source configuration	Input source options, e.g. column mapping
Output source configuration	Output source options
SortOrder	Sort order for columns
IdColumn	Column for identification
TableId	Identification of specific information groups, e.g. specific data sources
DuplicateColumn	Algorithm classification for each column
DuplicateRule	Rule sets with thresholds for every column Note: number of thresholds should conform to number of columns

Table 45: duplicate search configuration parameters

As those setting can only be defined via xml files at the moment, please contact prudsys staff for configuration.

Create or update a search

curl -X PUT -T "<SEARCH_FILE>" -H "Content-Type:text/xml" --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/<SEARCH-TYPE>/<SEARCH-ID>

“Search type“ can be replaced by the following types:

- similaritysearch
- duplicatesearch

“Search ID” is a unique key for every search and can be selected at your will.

The search configuration file will be stored in the following directory:

<RDE-DIR>/data/re_server/<RDE-ID>/datacleansing/<SEARCH-TYPE>/<SEARCH-ID>

Attention: An existing search configuration with the same name will be overwritten.

Get a list of configured search actions

curl -X GET --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/<SEARCH-TYPE>

Request the configuration of a search action

curl -X GET --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/<SEARCH-TYPE>/<SEARCH-ID>

Delete an existing search action

curl -X DELETE --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/<SEARCH-TYPE>/<SEARCH-ID>

10.4.2 Duplicate service search configuration

The search configuration for duplicate service contains identical configuration parameters like describes within Chapter 10.4.1. However, differences with the upload process are describes within the following sections.

Creating or updating a search

curl -X PUT -T "<SEARCH_FILE>" -H "Content-Type:text/xml" --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/services/duplicates

Note: Compared to similarity and duplicate search, the search-ID can not be chosen at will (“duplicates“).

The search configuration will be stored within the following directory:

<RDE-DIR>/data/re_server/<RDE-ID>/datacleansing/services/duplicates

Note: There can only be one search configuration for duplicate service. An already existing search will be overwritten with anew upload process. Furthermore, the search configuration can not be deleted.

Search configuration output

curl -X GET --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/services/duplicates

10.5 Similarity and duplicate search process

Similarity and duplicate search are based on the same process which will be described in the following chapters.

Attention: Duplicate Service (real time duplicate search) is based on a different process which will be described separately in Chapter 10.6.

10.5.1 Starting the search

Start a new search

curl -X POST --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/
<SEARCH-TYPE>/<SEARCH-ID>/process

“Search type“ can be replaced by the following types:

- similaritysearch
- duplicatesearch

Monitor the search process

Request the process ID:

curl -X GET --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/
<SEARCH-TYPE>/<SEARCH-ID>/process

To monitor the search process:

http://host:port/rde_server/admin/processes/<PROCESS-ID>

To stop the search process:

*http://host:port/rde_server/admin/processes/<PROCESS-ID>/**pause***

To resume the search process:

*http://host:port/rde_server/admin/processes/<PROCESS-ID>/**resume***

To cancel the search process:

*http://host:port/rde_server/admin/processes/<PROCESS-ID>/**cancel***

10.5.2 Search results

The result of the search can be requested in the following way:

curl -X GET --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/
<SEARCH-TYPE>/<SEARCH-ID>/result

"Search type" can be replaced by the following types:

- similaritysearch
- duplicatesearch

Additionally, an output file output.csv will be stored in the directory <RDE-DIR>/data/re_server/<RDE-ID>/datacleansing/data.

Search result output

The output file contains, additionally to the initial input columns (which can be defined within the search configuration), the following result columns:

Column	Description
dupid	duplicate ID
isduplicate	1 – is a duplicate, 0 – is not a duplicate
recordid	record ID of the input file
matchavg	average duplicate match value
matchmin	minimal duplicate match value
matchmax	maximal duplicate match value
dupcount	number of records with duplicates
matchrule	applied rule within search configuration

Table 46: output of duplicate search

10.5.3 Search statistics

The search statistic file statistic.csv is created automatically within the directory

<RDE-DIR>/data/re_server/<RDE-ID>/datacleansing/<SEARCH-TYPE>/<SEARCH-ID>

and contains the default columns "dupid" and "recorded". Additional columns can be defined within the search configuration.

10.6 Duplicate service search process

Duplicate service is the real time duplicate search.

The processes “Creating an RDE application“ (Chapter 10.2), “Master data upload“ (Chapter 10.3) and “Search configuration“ (Chapter 10.4.2) are the same as described above.

There are some additional functionality which will be described within the following chapters.

10.6.1 Data import

Data sets will be read from the file input.csv and kept in memory for further processing. The search has to be deactivated before the data import.

Start data import

curl -X POST -T “<INPUT_SOURCE>" -H "Content-Type:text/xml" --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/services/duplicates/addrecords

A new file “datastore.csv“ with all imported data records will be generated within the following directory:

<RDE-DIR>/data/re_server/<RDE-ID>/datacleansing/services/duplicates

10.6.2 Activate and Deactivate a search action

During the activation, comparison data is read and indices are created. After activation, new data records can be read and compare with existing data in real time.

Activate duplicate search

curl -X POST --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/services/duplicates/activate

Deactivate duplicate search

The memory for comparison data will be emptied.

curl -X POST --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/services/duplicates/deactivate

Request activation status (active | not active)

curl -X GET --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/services/duplicates/active

10.6.3 Online Requests

An imported data set (xml) will be compared to the existing data. All existing rules or only a single rule can be applied.

Search for duplicates

curl -X POST -T "<NEW_RECORD>" -H "Content-Type:text/xml" http://host:port/rde_server/res/<RDE-ID>/datacleansing/services/duplicates/search

Attention: The output is carried out only via command line interface, no output file will be generated.

Search for duplicates with a specific rule set

curl -X POST -T "<NEW_RECORD>" -H "Content-Type:text/xml" http://host:port/rde_server/res/<RDE-ID>/datacleansing/services/duplicates/rulesets/<RULESET-ID>/search

10.6.4 Import data sets in real time

For the online duplicate search it is possible to add data sets in real time.

Attention: This is only possible if a search action has been defined beforehand.

Data sets

List of all data sets

curl -X GET --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/services/duplicates/records

The following parameters can be used to specify the data set request:

- Sort=<INDEX>

Index of the search environment which shall be used for the sorting

- Top=<INDEX>

Index of the first data set

- Bottom=<INDEX>

Index of the last data set

- Limit=<INDEX>

Maximum number of data sets.

The required parameters can be used in the following way:

http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/services/duplicates/records?<PARAMETER1>=<INDEX1>&<PARAMETER2>=<INDEX2>

Request a single data set

curl -X GET --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/services/duplicates/records/<RECORD-ID>

Create or update data sets

curl -X POST -T "<NEW_RECORDS_FILE>" -H "Content-Type:text/xml" --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/services/duplicates/records

Create or update a single data set

curl -X PUT -T "<NEW_RECORDS_FILE>" -H "Content-Type:text/xml" --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/services/duplicates/records/<RECORD-ID>

Delete a single data set

curl -X DELETE --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/services/duplicates/records/<RECORD-ID>

Rule sets

Rule sets are defined within the search configuration by default. Additionally, there is also a possibility to use new rules in real time.

List of additionally configured rule sets

curl -X GET "Content-Type:text/xml" --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/services/duplicates/rulesets

Output of one rule set with index

curl -X GET --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/services/duplicates/rulesets/<RULESET-ID>

Note: ruleset-ID can be chosen at will.

Create a new rule set

A rule set will be sent as xml file. The ruleset-ID has to be defined within this file.

curl -X POST -T "<NEW_RULESET>" -H "Content-Type:text/xml" --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/services/duplicates/rulesets

Update a rule set with index

curl -X PUT -T "<NEW_RULESET>" -H "Content-Type:text/xml" --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/services/duplicates/rulesets/<RULESET-ID>

Delete a rule set

curl -X DELETE --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/datacleansing/services/duplicates/rulesets/<RULESET-ID>

11 prudsys RDE | Scoring

11.1 Introduction to RDE | Scoring Module

The prudsys RDE | Scoring module enables real-time or batch-wise forecasting of customer and user behaviour based on individual master data. Forecasting can be used to predict customer choices in direct marketing transactions, to make individual customer forecasts, to identify fraud or to identify shopping cart abandonment.

Your benefit

Customized optimisation of marketing and sales actions, reducing shopping cart abandonments and fraud detection

Business scenarios

The prudsys RDE | Scoring module makes it possible to forecast certain user reactions based on profile information and behaviour patterns. A classic area of use is the optimisation of mailings with respect to the probability of reaction from the recipient.

Integrating scoring into the real-time world of the prudsys RDE results in additional, completely new solutions, including the forecasting and prevention of purchase cancellations, the detection of attempts to defraud and the classification of online shop visitors in new or returning customers.

How it works

The RDE | Scoring module uses individual master data (historic data or real-time data) to calculate and update decisioning models which are then used to forecast preset target parameters. The module procedures are able to calculate models before a planned action is implemented or in parallel with an action already running.

Integration into the RDE Server

The RDE | Scoring module supplement all other RDE modules at the best. It benefits from the information allocated through the RDE | Data Cleansing module. Furthermore, it completes the RDE | Recommendations module with score calculations for web shop applications.

Functionalities RDE | Scoring module

- Online scoring for common scoring models
- Online scoring self-learning scoring methods
- Offline scoring for common data sources (text file or database)
- Automatic or semi-automatic (interactive) model calculation

RDE | Scoring module uses the functionalities of RDE | Data Cleansing to provide table information.

Handling

Note: Currently, the module RDE | Scoring is handled via command line interface only. Therefore, curl syntax is used for the following descriptions.

11.2 Model management

Scoring models

The central component of the prudsys RDE | Scoring module are scoring models. The scoring models contain the knowledge about the master data in a condensed form and are required to predict a result for new data.

PMML

The prudsys RDE | Scoring module uses the PMML 4.0 standard to represent scoring models and is able to use scoring models produced by various programs.

Importing a scoring model:

To import a scoring model into the prudsys RDE | Scoring module the PMML 4.0 file has to be uploaded to the RDE server.

curl -X POST -T "<PMMLFILE>" --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/scoring/models

The request stores the model with an automatically selected model identifier and returns the model URL.

The following command imports and updates the scoring model with a selected model identifier.

curl -X PUT -T "<PMMLFILE>" --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/scoring/models/<MODEL-ID>

Listing existing scoring models:

curl -X GET –digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/scoring/models

Exporting a scoring model:

curl -X GET –digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/scoring/models/<MODEL-ID>

Deleting a scoring model:

curl -X DELETE –digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/scoring/models/<MODEL-ID>

11.3 Online scoring

Online scoring services

The prudsys RDE | Scoring module offers online scoring services. The online scoring service receives records through the web interface, processes the records and returns the prediction as a result.

The result can contain various classification numbers depending on the type of the scoring model. The most common are the predicted class and a score.

The online scoring services are each backed by a scoring model.

The online scoring services must be activated to perform scoring and deactivated to change the configuration.

Creating an online scoring service:

The service can either be created from a scoring model

curl -X POST --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/scoring/models/
<MODEL-ID>/newservice

or from scratch without referencing a scoring model at first

curl -X POST -T "<INPUTFILE>" --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/scoring/services

Both requests create a new service with an automatically selected service identifier and return the URL of the created service.

Listing existing services:

curl -X GET --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/scoring/services

Reading the service configuration:

curl -X GET --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/scoring/services/<SERVICE-ID>

Updating a service:

curl -X POST -T "<INPUTFILE>" --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/scoring/services/<SERVICE-ID>

This request updates a service with an exported service configuration. If the service does not exist a new service is created with the given service identifier.

Deleting a service:

curl -X DELETE --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/scoring/services/<SERVICE-ID>

Set scoring model:

curl -X DELETE –data "<MODEL-ID>" --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/scoring/services/ <SERVICE-ID>/modelid

Get scoring model:

curl -X GET --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/scoring/services/ <SERVICE-ID>/modelid

Configure result fields:

curl -X PUT -T "<INPUTFILE>" --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/scoring/services/ <SERVICE-ID>/resultfields

Result fields

The result fields are configured with an XML document describing each field. The "type" field determines the value of the returned field. Which identifiers are valid for the "type" field depends on the type of the scoring model and is described within Chapter 11.5.

```
<ResultFields>
    <ResultField type="score"/>
    <ResultField type="class"/>
    <ResultField type="scorerel" parameter="targetclass"/>
    <ResultField type="sourcefield">recordid</ResultField>
</ResultField>
```

Get result field configuration:

curl -X GET --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/scoring/services/ <SERVICE-ID>/resultfields

Activate a scoring service

Activate a service:

curl -X POST --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/scoring/services/ <SERVICE-ID>/activate

Deactivate a service:

curl -X POST --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/scoring/services/ <SERVICE-ID>/activate

Get activation status:

curl -X GET --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/scoring/services/ <SERVICE-ID>/active

Applying a scoring model

Process record:

The scoring services offer three methods to process a record.

curl -X POST -T "<RECORDDATA>" --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/scoring/services/<SERVICE-ID>/apply

The model is applied to the record and the prediction results are returned. The model does not change.

curl -X POST -T "<RECORDDATA>" --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/scoring/services/<SERVICE-ID>/learn

The model is updated from the record. It is not applied to the data so no application results are returned.

curl -X POST -T "<RECORDDATA>" --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/scoring/services/<SERVICE-ID>/learnapply

The model is updated from the record, applied to the record and the prediction results returned.

Some scoring models can not be modified so the “learn” and “learnapply” methods are not available for those scoring models.

11.4 Offline Scoring

Scoring tasks

Offline scoring is done through scoring tasks. A scoring task is a preset configuration which can be used to start an offline scoring process.

The configuration consists of a scoring model, the result field configuration and the configurations for input and output data.

The scoring task is preset to either perform the “apply”, “learn” or “learnapply” methods.

Creating a scoring task:

The task can either be created from a scoring model

curl -X POST --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/scoring/models/<MODEL-ID>/apply

curl -X POST --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/scoring/models/<MODEL-ID>/learn

curl -X POST --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/scoring/models/
<MODEL-ID>/learnapply

or from scratch without referencing a scoring model at first

curl -X POST -T "<INPUTFILE>" --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/scoring/tasks

Both requests create a new scoring task with an automatically selected task identifier and return the URL of the created task.

Listing existing tasks:

curl -X GET --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/scoring/tasks

Reading the task configuration:

curl -X GET --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/scoring/task/<TASK-ID>

Updating a task:

curl -X POST -T "<INPUTFILE>" --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/scoring/tasks/<TASK-ID>

This request updates a task with an exported task configuration. If the task does not exist a new task is created with the given task identifier.

Deleting a task:

curl -X DELETE --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/scoring/tasks/<TASK-ID>

Set scoring model:

curl -X DELETE –data "<MODEL-ID>" --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/scoring/tasks/<TASK-ID>/modelid

Get scoring model:

curl -X GET --digest -u username:password
http://host:port/rde_server/admin/res/<RDE-ID>/scoring/tasks/<TASK-ID>/modelid

Result fields

Configure result fields:

curl -X PUT -T "<INPUTFILE>" --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/scoring/tasks/<TASK-ID>/resultfields

Get result field configuration:

curl -X GET --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/scoring/tasks/<TASK-ID>/resultfields

Input source

Set input source configuration:

curl -X PUT -T "<DATASOURCE>" --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/scoring/tasks/<TASK-ID>/input

The sent data must be a data source configuration as described within Chapter 11.6.

Get input source configuration:

curl -X GET --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/scoring/tasks/<TASK-ID>/input

Output target

Set output target configuration:

curl -X PUT -T "<DATASOURCE>" --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/scoring/tasks/<TASK-ID>/output

The sent data must be a data source configuration as described within Chapter 11.6.

Get output target configuration:

curl -X GET --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/scoring/tasks/<TASK-ID>/output

Starting the scoring process

Starting the scoring process:

curl -X POST --digest -u username:password http://host:port/rde_server/admin/res/<RDE-ID>/scoring/tasks/<TASK-ID>/process

The scoring task will start a scoring process and return its URL. The progress and state of the process can be monitored via the returned URL.

11.5 Scoring model result fields

All scoring models at least support the following result fields

- score: score value
- source field: copies a field from the input record

Decision tree models

Decision tree models offer more details on a classification result. All result fields except for "score" and "class" accept a class name as parameter. If no parameter is given or left empty, the predicted class will be used.

- score: distribution of the predicted class in the result node
- class: predicted class
- scorerel: distribution of a class in the result node
- scoreabs: absolute number of class records in the result node
- totalrecordcount: total number of class records
- cumulativerecordcount: number of class records in all tree nodes with a higher score than the result node, including the result node
- gain: cumulated number of class records relative to the total number of class records
- partofpopulation: cumulated number of record relative to the total number of records

11.6 prudsys RDE | Data sources

prudsys RDE | Data sources are used in various places in the RDE server to configure access to files and database servers. The data source configurations are XML documents describing location, access parameter and format of data sources.

Data sources can be used for as input source and output target.

prudsys RDE | Data sources support read and write support of CSV files and JDBC databases.

CSV files

CSV files are widely used to transfer data between different systems. The files are basically text files with one record per text line and record fields separated by a single character.

Despite the simple structure there are many different variants of the CSV file format mostly differing in the used field separator and the quotation style for text and special characters.

The prudsys RDE | Data sources provide a wide variety of parameters to adjust to the used CSV file format but provides an automatic detection for the most common dialects.

```
<DataSource>
    <FileName>items.csv</FileName>
    <CSVOptions>
        <charset>UTF-8</charset>
        <useTextQualifier>false</useTextQualifier>
        <delimiter>|</delimiter>
    </CSVOptions>
</DataSource>
```

- delimiter: Field delimiter (Character)
- comment: Comment character (Character)
- escapeModeType: Escape mode for special characters
  - ESCAPE_MODE_BACKSLASH: Backslash escapes characters
  - ESCAPE_MODE_DOUBLED: The quotation character is doubled
- columnNameType: Column naming mode
  - COLUMN_NAME_AUTOMATIC_MODE: The mode is automatically detected
  - COLUMN_NAME_FIRST_LINE: The first line contains column names
  - COLUMN_NAME_CREATE: The columns are named field1.. fieldN
- skipEmptyRecord: Empty records are ignored (Boolean)
- trimWhitespace: Whitespaces before and after values are removed (Boolean)
- useComments: All characters on a line after a comment character are ignored (Boolean)
- charset: Input character encoding (String)
- testLines: Number of lines to use for automatic format detection (Integer)
- missingValues: List of strings which are used for missing values (String-List)
- textQualifier: Quotation character (Character)
- numberFormatClass: Class name of a custom NumberFormat Parser (String)
- numberFormatPattern: Pattern for a DecimalFormat Parser (String)
- numberFormatLocale: Locale identifier for numeric values (String).

Identifiers are formed Language_Country_Variant. Variant or country and variant can be omitted. In this case the default locale for the langue will be used. The identifier for language and country use the „ISO Language Code“ ISO-639 und „ISO Country Code“ ISO-3166.

Possible values are de, de_CH or es_ES_Traditional_WIN.

- useTextQualifier: Quotation character is used (Boolean)
- defaultColumnType: All columns are using this type by default
  - COLUMN_TYPE_AUTOMATIC_MODE: Column type is detected automatically

  - COLUMN_TYPE_CATEGORICAL_UNBOUNDED: String with dictionary, equal strings are assigned the same value
  - COLUMN_TYPE_CATEGORICAL_UNSTORED: String without dictionary
  - COLUMN_TYPE_NUMERIC_DOUBLE: Numeric double precision
  - COLUMN_TYPE_NUMERIC_FLOAT: Numeric single precision
  - COLUMN_TYPE_NUMERIC_INTEGER: Numeric integer
- columnMappings: List of column to name and type mappings
  - column: column number
  - name: column name
  - type: column type. See defaultColumnType

A column number has precedence over the column name if both are given.

JDBC database

JDBC is the most widely used standard to access database systems from the Java platform. The java programs access the database server with driver software that is supplied from the database manufacturer.

prudsys RDE | Data sources uses JDBC to read and write data from databases.

```
<DataSource>
    <TableName>customers</TableName>
    <JDBCOptions>
       <Driver>org.postgresql.Driver</Driver>
       <DriverURL>jdbc:postgresql://server/db</DriverURL>
       <Username>user</Username>
       <Password>password</Password>
       <DriverSource>pgsql.jdbc4.jar</DriverSource>
    </JDBCOptions>
</DataSource>
```

- Driver: JDBC driver class name
- DriverURL: JDBC URL
- Username: User name to use for JDBC connection
- Password: Password to use for JDBC connection
- DriverSource: URL to an external driver JAR file