Nan21 e-Business Suite User Manual

Version 0.9+

Table of Contents

1 Introduction

1.1 Overview

Nan21 e-Business Suite is an integrated business application with Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM) functionalities for small and mid-size companies.

Main application modules:

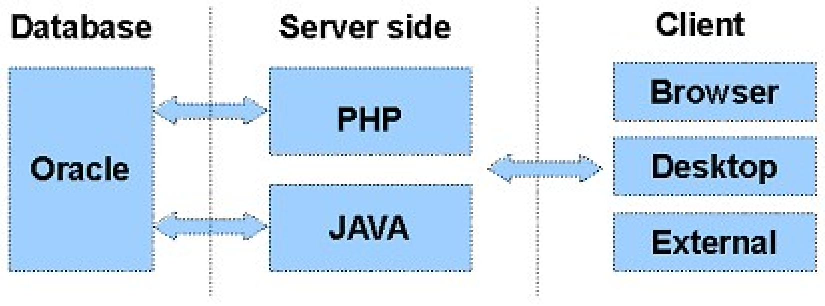
- Material Management: Product master-data, stocks and inventory movements.
- Business partners. Customers, vendors, third parties involved in commercial relationships with your company.
- Sales: Pricing, Contract and Order management.
- Distribution: Transports and parcels, mainly for parcel delivery companies, delivery life-cycle management
- Financial: includes Invoicing, Expenses, Payments and Accounting.
- Project Management: Projects and issues.
- Administration: Application and organization configuration and maintenance.

1.2 Architecture overview

Architecture is a standard 3 tier approach, with database, server-side controller running in a web-server and a client. Multiple server-side controllers and clients are available, users can select the formula which they like most.

Client can be one of the provided user interfaces, a third party application which needs to exchange data with N21EBS, custom code written by users/integrators, etc.

Application business logic is exposed to clients by the controllers based on HTTP requests, response data available in JSON or XML format.



Architecture overview

1.2.1 Database

The database is used for the data storage and also implements business logic in PL/SQL packages, therefor N21EBS is not database independent, is available only with Oracle database.

1.2.2 Middle tier

There are two middle tier implementations, one using Php and the other one built on Java.

Although both implementations have the same functions yet, we recommend to consider for production mode the Java implementation as, in time, this will be the reference core.

From clients perspective both implementations work the same way, a set of request parameters performs the same business logic function.

1.2.3 Client

Multiple user interfaces are planned to be released, for the moment there is released a browser based one built on Extjs AJAX library, the second one is an experimental desktop application running on Adobe Air using the same code base as the browser one.

Note:
The Java middle tier and the Adobe Air desktop client are available starting with version 0.9 which is desired to be the first beta release.

2 Installation

This chapter contains basic information about installing Nan21 e-Business Suite.

It assumes you have already setup the infrastructure necessary to install N21EBS or you have the knowledge to do so. If you do not have experience with some of the required technologies, we suggest you to read the documentation from the publisher in order to understand the basic concepts.

2.1 Before you install

Before you install N21EBS, setup the necessary infrastructure required by N21EBS:

- Install Oracle database 10g, 11g or XE. If you need help on this topic, visit Oracle Documentation at http://www.oracle.com/technology/documentation/index.html
- In case you want to use N21EBS with the PHP middle tier, setup an Apache web server and activate PHP. If you need help on this topic, visit Apache website at http://httpd.apache.org and PHP Documentation at http://www.php.net/docs.php
- For the Java based middle tier you have to install a web server able to execute Java servlets. Tomcat: http://tomcat.apache.org/ , JBoss: http://www.jboss.com/

2.2 Install database objects

Steps to be performed:

1. Create Oracle user.
2. Setup schema objects.
3. Import data to initialize database.
4. Tune database

Note:
If you need help on using Oracle SqlPlus, visit product documentation center at:
http://www.oracle.com/technology/docs/tech/sql_plus/index.html

1) Connect with SqlPlus to Oracle with a user which has the necessary rights to create database users and grant roles.

Execute: /Database/create_user.sql

When prompted, enter the schema name which will be the owner of the database objects. By default it is NAN21 with password NAN21

You can change later the database user password.

2) Connect with SqlPlus to Oracle with the new user:

SQL>connect NAN21/NAN21@XE; (i.e. : user/password@Service_Name)

If at first step you choose to give a different name to the database schema as the default one, use your user-name and password.

Execute: /Database/setup_db.sql

Note: During object creation there might appear errors because certain objects are referenced and are not yet created. After all objects are created the schema is recompiled and should not be invalid objects.

3) Execute: /Database/data/init_minimal.sql

4) Depending on your database installation there might be necessary to set some system parameters. Check with your database administrator.

For a newly installed Oracle XE change the following system parameters: PROCESSES to 100 and SESSIONS to 115

```
alter system set PROCESSES=100 scope=SPFILE;
alter system set SESSIONS=115 scope=SPFILE;
```

2.3 Install middle tier

There are two middle tier implementations available, you can choose one of them or use both.

Our recommendation is to use the Java one as this is the reference implementation for production releases, and perhaps will be more feature rich. The Php implementation is used mainly for fast in-house development.

2.3.1 PHP implementation

Before installation check the php.ini file and enable php_oci8, uncommenting the following line: extension=php_oci8.dll

If you encounter difficulties setting up the Oracle enabled php environment there is a very good article on this topic:

http://www.oracle.com/technology/pub/notes/technote_php_instant.html

Create a folder named n21eBusinessSuite in your php document root directory.

Copy the content of /ServerPhp from the N21EBS download to the folder created above.

1. Edit file /ServerPhp/properties_oracle.php

There you have to change the connection information according to your Oracle database setup and user definition. Look for the following code:

```
$_n21_tns_["NAN21"] = array(
            "DB_TNS" => "(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=toshiba2)
(PORT=1521))(CONNECT_DATA=(SID=XE)))"
```

```
            ,"DB_USER" => "NAN21"
            ,"DB_PASSWORD" => "NAN21");
```

It is a standard TNS descriptor as Oracle defines it in tnsnames.ora

Change the values for HOST, PORT and SID according to your database installation.

Check Oracle documentation for your database version if you are in doubt on how to set them.

Change the value for DB_USER and DB_PASSWORD to the values specified when you created the application database user.

2. Edit file /ServerPhp/properties.php

Set the working mode for the built-in basic access right check: enabled or disabled.

```
define('DO_DC_PERMISSION_CHECK', 'Y' );
```

Specify the logging level

```
define('LOG_LEVEL', 3 );
```

2.3.2 JAVA implementation

2.4 Install client

2.4.1 Browser based client

The same user interface code can be used either with the Java or Php middle tier. However, due to browser security restrictions the user interfaces must be loaded from the same domain as the XHR(Xml Http Request) calls are sent to perform data processing. The same domain means to be the same protocol, server name and port.

Thus, if you want to to use a certain middle tier and you have installed it to be accessible for example from the following url: http://mydomain:80/n21eBusinessSuite/, then the user interfaces must be loaded in browser from the following address: http://mydomain:80

1. **If you choose to use the browser based user interface with Php middle tier**

   You have already installed the middle tier to the following folder:

   YOUR_PHP_DOCUMENT_ROOT/n21eBusinessSuite/ServerPhp

   Copy the folder /ClientExtjs from the downloaded package to your n21eBusinessSuite root directory.

   Now you should have :

   YOUR_PHP_DOCUMENT_ROOT/n21eBusinessSuite/ServerPhp containing the php scripts for the server-side processor

   and

YOUR_PHP_DOCUMENT_ROOT/n21eBusinessSuite/ ClientExtjs the scripts for the user interfaces.
Edit file n21eBusinessSuite/ClientExtjs/index.html
Set the value to 'php' for the following parameter:

```
CFG_DEPLOYMENT = 'php';   // values: {php,java}
if (CFG_DEPLOYMENT == 'php') {
  ..... other parameter values...
}
```

Set the proper values for the subsequent parameters in the php branch.

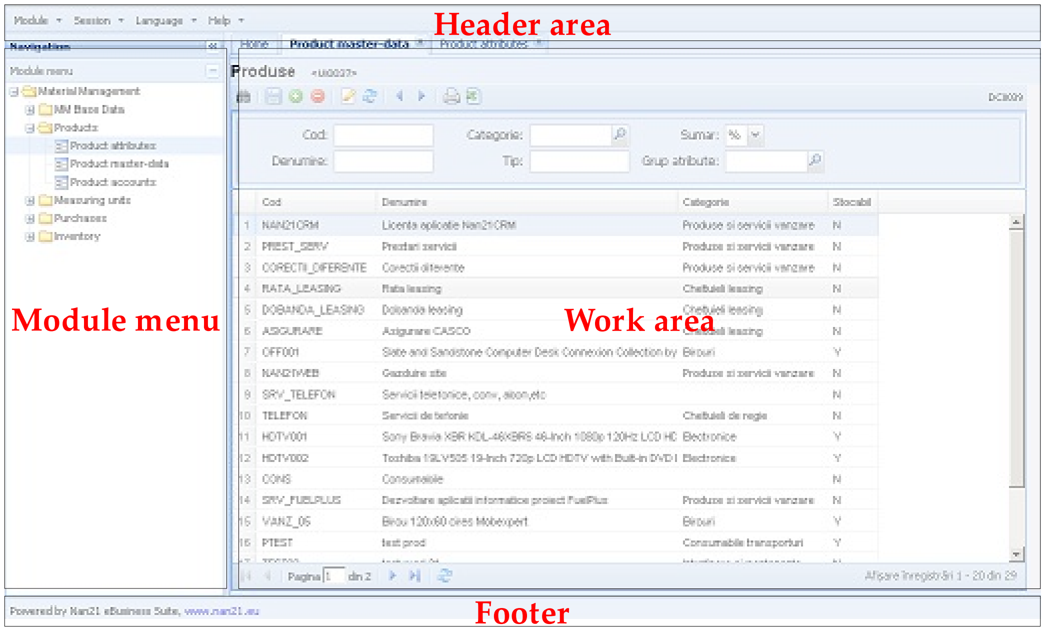
2. **If you choose to use the browser based user interface with Java middle tier**

2.4.2 Adobe Air desktop client

3 User interface fundamentals

3.1 User interface areas

The design of the screen consists of four areas: header area, module menu, work area and footer.



	Cod	Denumire	Categorie	Stocabil
1	NAN21CRM	Licenta aplicatie Nan21CRM	Produse si servicii vanzare	N
2	PREST_SERV	Prestari servicii	Produse si servicii vanzare	N
3	CORECTII_DIFERENTE	Corectii diferente	Produse si servicii vanzare	N
4	RATA_LEASING	Rata leasing	Cheltuieli leasing	N
5	DOBANDA_LEASING	Dobanda leasing	Cheltuieli leasing	N
6	ASIGURARE	Asigurare CASCO	Cheltuieli leasing	N
7	OFF001	Slate and Sandstone Computer Desk Connexion Collection by	Birouri	Y
8	NAN21WEB	Gazduire site	Produse si servicii vanzare	N
9	SRV_TELEFON	Servicii telefonice, conv, abon,etc		N
10	TELEFON	Servicii de telefonie	Cheltuieli de regie	N
11	HDTV001	Sony Bravia XBR KDL-46XBR5 46-Inch 1080p 120Hz LCD HD	Electronice	Y
12	HDTV002	Toshiba 19LV505 19-Inch 720p LCD HDTV with Built-in DVD	Electronice	Y
13	CONS	Consumabile		N
14	SRV_FUELPLUS	Dezvoltare aplicatii informatice proiect FuelPlus	Produse si servicii vanzare	N
15	VANZ_05	Birou 120x60 cires Mobexpert	Birouri	Y
16	PTEST	test prod	Consumabile transporturi	Y

Overview of Screen Area

3.1.1 Header area

Header area consists of a main menu in the left and user information in the right

Main menu items:

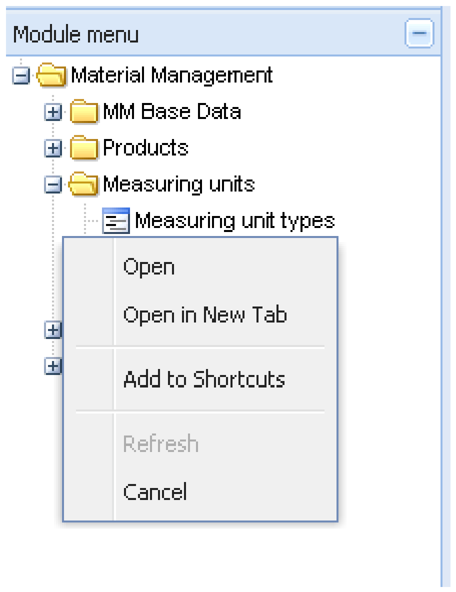
1. Module: Select application module and load the associated menu links in the Module menu navigation tree.
2. Session:
    - Log-in: Log-in if session expired or session has been locked
    - Log out: Logs the user off
    - Lock the working session
3. Language: Select the appropriate language
4. Help

3.1.2 Module menu

It consists of two navigation menu trees : Standard menus and Personal shortcuts.

The selection of a certain module from the Module menu item in Header area, refreshes the navigation menu tree links with the menu items associated to selected module.

Menu item level actions: Right clicking on a tree menu item the context pop-up menu is shown with the following functions:



Module menu pop-up on mouse right click

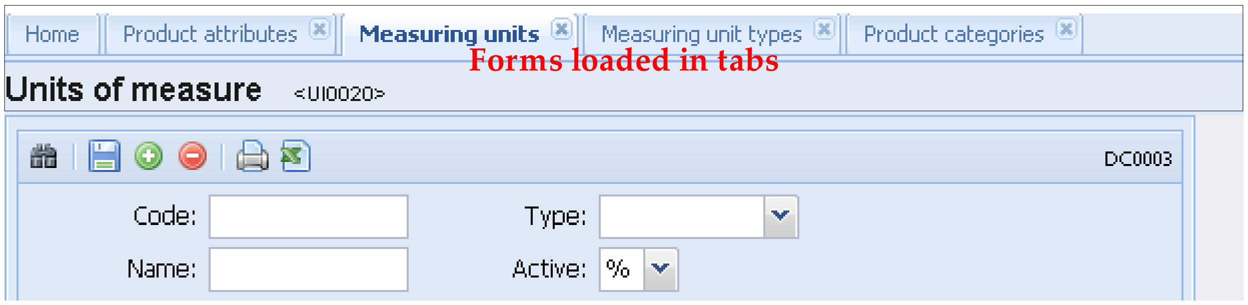
- Open:
  1. If the selected user interface is not already loaded in the work area then creates a new tab in work area and loads the form into that tab
  2. If it is already loaded but the tab is not active(the current tab is not the one where the selected form is already loaded) then activates the tab where the selected form is loaded.
  3. If it is loaded and the tab is active then reloads the form.
- Open in New Tab: A new tab is created in work area, is activated and the form loaded into.
- Add to Shortcuts: Create an entry in the Personal Shortcuts menu much like the “Add to Favorites” or “Bookmark” function provided by browsers.
- Refresh: Applies only to folders to refresh the sub-menus menus dynamically.
- Cancel: Hide the pop-up menu.

3.1.3 Footer

Contains usual footer information, company name , website and so on.

3.1.4 Work area

It is the main content area where forms are loaded. It consists of a main tab panel and every form is loaded in a separate tab. This allows a multi session work, you can easily switch from one tab to another to perform the business functions.



Work area tab panel

Once you have done with a certain form the tab panel can be closed and re-opened any time later

3.2 User interface components

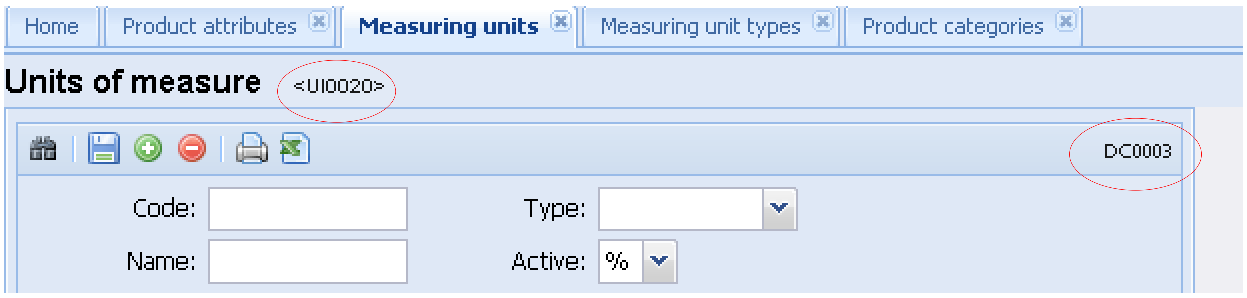
There are several base components used in user interface design:

- Forms: referred as UIxxxx
- Data-controls referred as DCxxxx
- Lists of values referred as LOVxxxx
- Reports

3.2.1 UI forms

A form (UI) is a user interface unit where users can perform certain business operations. Each form consists of one or more data-controls (DC) . There is one main data-control per form and may be other child data-controls.

For example the Invoice form has one parent data-control to edit invoice header information and one detail data-control to edit invoice lines. Each form shows the form code (for example UI0020) and the data-controls also show their code (ex: DC0003). Thus it is easy to refer to a certain user interface component based on its code although titles are translated in different languages.



Form and data-control code

3.2.2 Data-controls

Data controls are logical entities built around a certain data-source and respond to user actions.

They consist of a set of properties (fields) and certain methods (perform business functions) .

Each data-control has two parts:

– view (or presenter) which renders a user interface to deal with the data

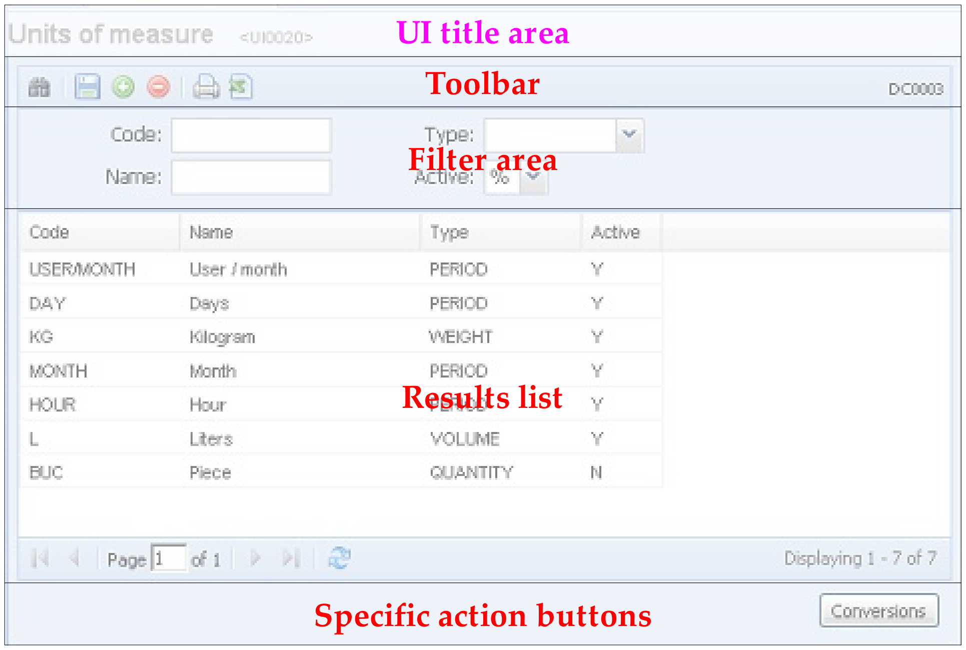
– server side controller which perform the business logic functions and return the proper results.

Data controls are the jig-saw pieces the whole application is built upon. From the data-control presenter perspective, there are three main types of data-controls: grid, editor and grid with editor.

In the following paragraphs we detail these types highlighting similarities and differences.

3.2.2.1 Grid data-control

A grid data-control presents records in a list view with or without in-line editing enabled.

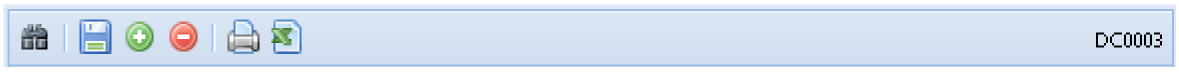


Code	Name	Type	Active
USER/MONTH	User / month	PERIOD	Y
DAY	Days	PERIOD	Y
KG	Kilogram	WEIGHT	Y
MONTH	Month	PERIOD	Y
HOUR	Hour	PERIOD	Y
L	Liters	VOLUME	Y
BUC	Piece	QUANTITY	N

Grid data-control overview

It consists of four main areas: toolbar area, filter area, results list and specific action buttons area.

3.2.2.1.1 Toolbar area



Grid data-control toolbar

Standard toolbar buttons for a grid data-control are:

Apply Filter: Shortcut key: F8. Fetches the records according to the specified filter criteria.

Save: Shortcut key: Ctrl+S. Saves the changes in grids where editing is allowed. If neither insert nor update is allowed this button is disabled.

To use keyboard shortcut the in-line editing finished.

Create new: Shortcut key: Ctrl+N. Insert a new line in grid, set new record default field values and sets the focus to the first field.

Delete: Shows a delete confirm dialog box with to options:

Yes - Deletes the selected record

No - Just close the confirm dialog box.

Print: Opens the results list in a new window in a printer friendly format.

Note:
For all the keyboard shortcuts to work it is necessary that the data-control has focus.

3.2.2.1.2 Filter area

Specify filtering criteria in order to see a restricted set of results in the results list.

It is also referred to as basic filter.

Filter is case sensitive. “Foo” is not the same with “foo”.

Using placeholders:

String placeholder: “%”

Starts with: “A%” returns all results which start with “A” . Does not return those which start with “a”

Ends with: “%A” returns all results which end with “A”

Contains: “%A%” returns all results which have “A” where ever within

It is not recommended to use the “Ends with” or “Contains” criteria as they perform a full table table search and might take very long time, slowing down the system (in case of large data sets)

One character placeholder: “_”

Example: “AB_” returns any three character long strings which start with “AB”. Finds any of the following: “ABC”, “AB1”, “AB-” , but not returns “ABAA”.

These are common filtering issues applies on filtering for all types of data-controls.

3.2.2.1.3 Results list

It is a list view of the results fetched. More detailed description of the available features and behavior of the list view are available at the components description section.

If insert or update is allowed it performs in-line data editing similar to spreadsheets.

3.2.2.1.4 Specific action buttons

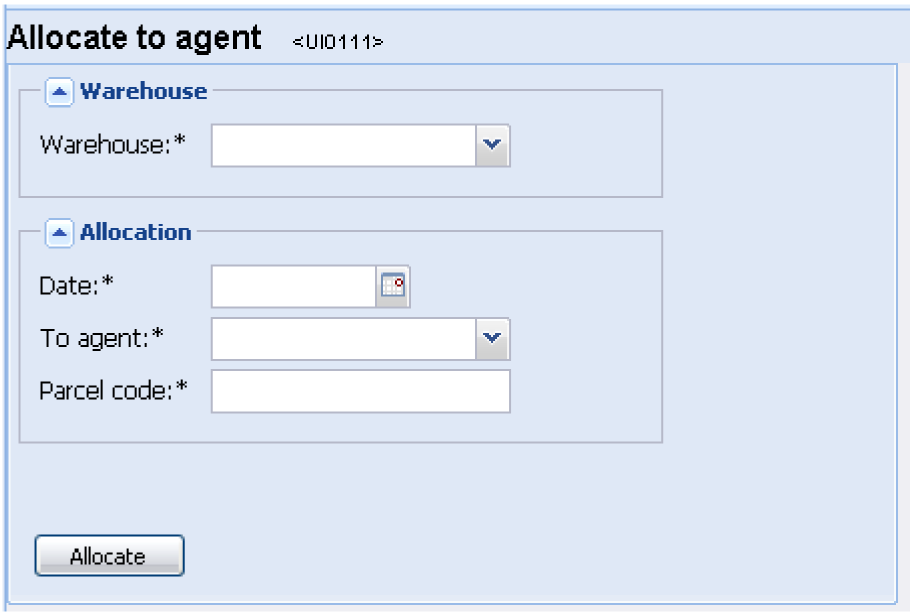
If a data-control requires some specific functions the commend buttons are placed in this area.

Specific actions might be to open child data-control windows, call server-side procedures, run specific reports, etc

3.2.2.2 Edit-Form data-control

It is a less used data-control, usually used in situations where based on some input parameters a certain server side function has to be executed.

For example DC0113 linked to UI0111 Allocate parcel to delivery agent.



Edit-form data-control example

3.2.2.3 Grid with edit-form data-control

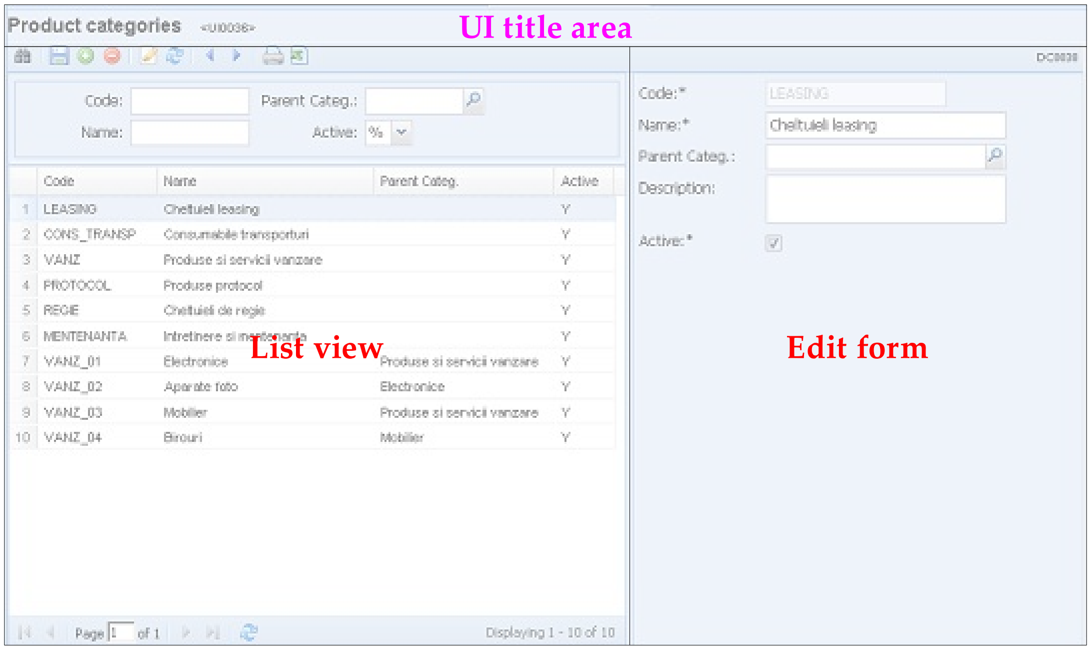
This data-control is a composite control which has a list view and a record editor as an edit-form.

It comes in three flavors based on how the list view and editor view are rendered:

1. row layout: both list and edit form are visible rendered in a row
2. column layout: both list and edit form are visible rendered in a column
3. card layout: only one view is visible at a time, either the list view or the edit-form view

The row and column layout is used for data controls with few fields whereas the card layout is used for those with many fields or complex layout in editor form.

All three layout performs the same, however the card layout due to the fact that only view is visible at a moment has some small particularities.



Product categories «UI0036»

Code: Parent Categ.:
Name: Active: %

	Code	Name	Parent Categ.	Active
1	LEASING	Cheltuieli leasing		Y
2	CONS_TRANSP	Consumabile transporturi		Y
3	VANZ	Produse si servicii vanzare		Y
4	PROTOCOL	Produse protocol		Y
5	REGIE	Cheltuieli de regie		Y
6	MENTENANTA	Intretinere si mentenanta		Y
7	VANZ_01	Electronice	Produse si servicii vanzare	Y
8	VANZ_02	Aparate foto	Electronice	Y
9	VANZ_03	Mobilier	Produse si servicii vanzare	Y
10	VANZ_04	Birouri	Mobilier	Y

Page 1 of 1 Displaying 1 - 10 of 10

DC0036

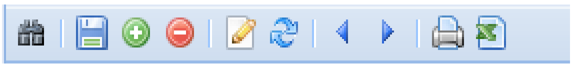
Code:* LEASING
Name:* Cheltuieli leasing
Parent Categ.:
Description:
Active:* ✓

Example of row layout grid with editor form

As you can see the list view of this control follows very much the same pattern as the grid data-control. We can find the same areas (toolbar, filter area, results list). The difference is that the grid does not allow editing. On selecting a record in list, the details are loaded into the editor form where users can modify the data.

Navigate through the results list with up-down arrow keys or with mouse click. The details of the current record are loaded into the editor form. Double-click a record or hit enter while it is selected and focused and the edit mode is triggered. The first editable field is focused and now you can start editing and navigate through the fields with the TAB key.

In case of card layout, as only one view is visible at a moment, the edit/list mode trigger described above also activates the proper view. In list view double click a record or hit Enter when the record is selected and focused thus activates the editor form with the current record loaded for edit. The same effect has the Editor/ List toolbar toggle button. When finished editing, return to list by releasing the pressed toolbar button or with Ctrl+Q keyboard buttons having the focus within the data-control editor form.


Grid with edit form toolbar

First four and last two toolbar items are the same as for the grid data-control.

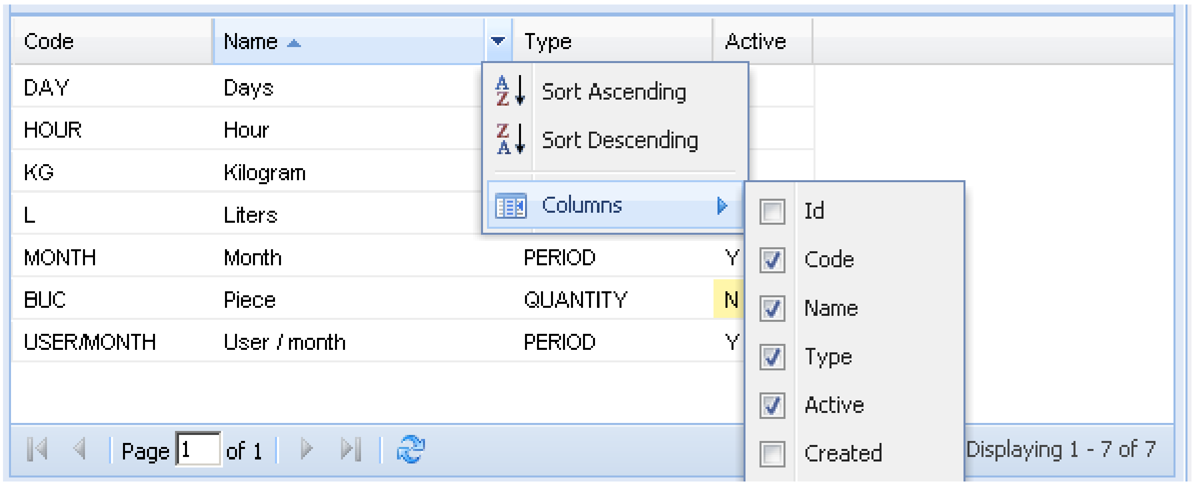
Editor/List: Shortcut key: Enter /Ctrl+Q toogle editor/list view as described above.

Refresh record: Refreshes the current (selected) record from database.

3.3 Basic controls

3.3.1 List view

Displays a collection of records. The list view behaves the same regardless of what data-control is in (grid, grid with edit-form, list of values opened by a lookup field), thus providing an easy to use uniform user interface.



Code	Name	Type	Active
DAY	Days		
HOUR	Hour		
KG	Kilogram		
L	Liters		
MONTH	Month	PERIOD	Y
BUC	Piece	QUANTITY	N
USER/MONTH	User / month	PERIOD	Y

Example of a list view

Features:

Sort: sort the results by clicking on the column header or on the arrow

Show/Hide columns: Check/Un-check the columns in the drop-down list.

Resize columns

Change columns position by drag and drop.

Results page navigator at the bottom to fetch previous/next page of records in larger datasets.

Note:
List view personalization is transferred to the export function. List print or export delivers all the data according to filter criteria in the same packaging (columns included in export, columns position, sort) as the list is configured.

3.3.2 Fields

There are several types of fields used in N21EBS most of them are familiar to users. However there are some small particularities for some of them.

Regular fields (form fields, filter panel fields):

1. Textfield: Allows regular alphanumeric and special characters content
2. Datefield: Allows only dates to be entered using the selected date format. Has a pop-up date selector panel attached.

3. Numberfield: Allows only valid numeric content (0-9,.-)
4. Checkbox
5. Combobox: Drop-down selection list.
6. Lookup: Textfield with a filterable list of values attached in a pop-up window.

Although the first four fields are the very usual in every aspect the last two fields have some specific behaviors and here is a brief description:

Combobox: When the drop-down list of options is opened for the first time (first time since the UI is loaded) it fetches the options list from the server and for performance reasons caches the values locally.

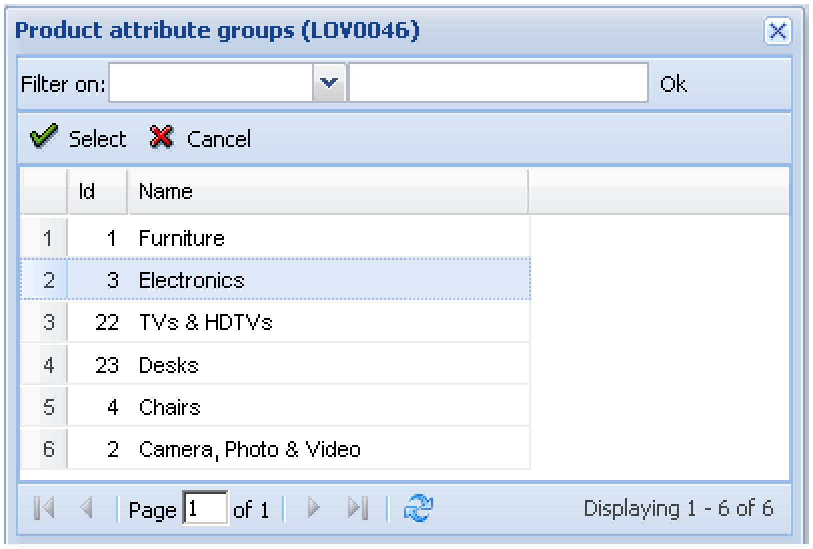
If the list of options is changed (you add or remove values) you have to close the UI tab and open it again to fetch again the data from server.

Example: The units of measure combo-box. When you open the "Product master-data" <UI0037> and open the drop-down list of the "UoM" field, the list of available units of measure are fetched from the server.

At a certain moment you realize that there is a unit which is no yet defined in your system, you will open "Units of measure" <UI0020> from the menu tree, define the new unit and save it. The "UoM" combo-box will not show the newly created record until you close the "Product master-data" tab and open it again from the menu tree.

Exception: Combo-boxes which are linked to one or more other fields; these are refreshed whenever one of the values in the linked fields is changed. For example the "Region" combo-box refreshes the values if you change the country

Lookup: Opens a window with a list view of the available options. It is a regular list view as described in the previous section. The filter however is a specific one:


Product attribute groups (LOV0046)

Filter on: Ok

Select Cancel

	Id	Name
1	1	Furniture
2	3	Electronics
3	22	TVs & HDTVs
4	23	Desks
5	4	Chairs
6	2	Camera, Photo & Video

Page 1 of 1 Displaying 1 - 6 of 6

At the top of the window select the column you want to filter on and enter the filter criteria in textfield then click "Ok"

4 Programmers guide

4.1 Architecture insides

4.2 What is a data-control

The most important thing to be understood is the data-control. These are the basic building blocks of N21 e-Business Suite.

4.3 Calling middle tier resources

Parameters are common resource and operation identifiers and data context properties. They can be sent either as GET or POST request.

1. Common action identifiers:

   _p_dc: Identifies the data-control. Ex: DC0004. (Mandatory) .(Used to be _p_form)

   _p_dc_name: (Not available yet) Identifies the data-control by name. Ex: InvoiceHeader. Is intended to be introduced to be easier to integrate calls to N21EBS. When implemented will be mandatory to specify either the data-control code by _p_dc parameter or the data-control name by this parameter.

   _p_action: What action is performed. Allowed values are: fetch, fetchRecord, initRecord, insert, update, delete, export, custom. (Mandatory)

   _p_custom_action: If parameter _p_action = "custom" it is expected the name of the custom action to be sent in this parameter. (Mandatory if _p_action = "custom")

   _p_fetch_format : Specifies the format of the data returned for all the actions except export.

   Allowed values are json and xml. (Used to be _p_data_format)

   _p_export_format : Specifies the format of the data returned for all the actions except export.

   Allowed values are "html", "csv", "pdf", "xml" and "json" (Used to be _p_exp_format)

   _p_transport: Simple transport of values, serialized comma separated parameter=value pairs to be sent back as they are. Used in asynchronous calls of insert/update in grids to identify the record ID in the collection. (Optional)
2. Data context properties. These are data-control specific parameters mainly field names, sort, pagination, etc. Check the "Data controls reference" document for description of existing data-controls..

_p_fetch_sort: Specifies query order by clause(Used to be sort) _p_fetch_sense: Specifies the order sense (ascending, descending)(Used to be sense)

_p_fetch_start: Limit of returned results, start from record (Used to be start)

_p_fetch_size: Limit of returned results, number of returned records.(Used to be limit)

4.3.1 Fetch

A usual fetch request is like this:

frmMain?_p_action=fetch&_p_dc=<DCcode>&_p_fetch_format=<format>&<query_fields>&<options>

which says fetch data for data control <DCcode> and return results in <format>

<query_fields> query-field value parameters

<fetch-options> sort, sense, results limit. As described above in “Data context properties”

Examples:

Get all the currencies in JSON format

```
frmMain?_p_action=fetch&_p_fetch_format=json&_p_dc=DC0002
```

returns:

```
{ success : true , totalCount:4, records : [  {NAME:"US+Dollar",ID:"54",ACTIVE:"Y",CODE:"USD"},{NAME
:"Romanian+leu",ID:"1",ACTIVE:"Y",CODE:"RON"},{NAME:"Euro",ID:"2",ACTIVE:"Y",CODE:"EUR"},
{NAME:"British+Pound",ID:"45",ACTIVE:"Y",CODE:"GBP"}] }
```

Get all the currencies in XML format:

frmMain?_p_action=fetch&_p_fetch_format=xml&_p_dc=DC0002

returns:

```
<?xml version='1.0' encoding='UTF-8'?>
<response>
 <totalCount>4</totalCount>
 <records>
  <record>
       <NAME>US Dollar</NAME>
       <ID>54</ID>
       <ACTIVE>Y</ACTIVE>
       <CODE>USD</CODE>
 </record>
 <record>
       <NAME>Romanian leu</NAME>
       <ID>1</ID>
       <ACTIVE>Y</ACTIVE>
       <CODE>RON</CODE>
 </record>
 <record>
       <NAME>Euro</NAME>
       <ID>2</ID>
```

```
        <ACTIVE>Y</ACTIVE>
        <CODE>EUR</CODE>
 </record>
 <record>
        <NAME>British Pound</NAME>
        <ID>45</ID>
        <ACTIVE>Y</ACTIVE>
        <CODE>GBP</CODE>
 </record>
 </records>
</response>
```

Get all the currencies where ACTIVE=Y in XML format:

frmMain?_p_action=fetch&_p_fetch_format=xml&_p_dc=DC0002&QRY_ACTIVE=Y

4.3.2 Insert

4.3.3 Update

A usual update request is like this:

    frmMain?_p_action=update&_p_dc=<DCcode>&_p_fetch_format=<format>&<fields>

<fields> is the list of fields and values

Example:

frmMain?_p_action=update&_p_dc=DC0002
&_p_fetch_format=json&ID=2&CODE=EUR&NAME=Euro&ACTIVE=Y returns:

```
{ success : true, record : {NAME:"Euro",ID:"2",ACTIVE:"Y",CODE:"EUR"}}
```

frmMain?_p_action=update&_p_dc=DC0002

&_p_fetch_format=xml&ID=2&CODE=EUR&NAME=Euro&ACTIVE=Y returns:

```
<response>
 <success>true</success>
 <record>
        <NAME>Euro</NAME>
        <ID>2</ID>
        <ACTIVE>Y</ACTIVE>
        <CODE>EUR</CODE>
 </record>
</response>
```

frmMain?_p_action=update&_p_dc=DC0002

&_p_fetch_format=json&ID=2&CODE=EUR&NAME=Euro&ACTIVE=Y&_p_transport=storeRecordId=100 returns:

```
{ success:true, record:{NAME:"Euro",ID:"2",ACTIVE:"Y",CODE:"EUR"}, transport:{storeRecordId:"100"} }
```

4.3.4 Delete

4.3.5 Export

4.3.6 Custom action

5 External links and resources

Oracle SqlPlus documentation:
http://www.oracle.com/technology/docs/tech/sql_plus/index.html

Installing PHP and the Oracle Instant Client for Linux and Windows
http://www.oracle.com/technology/pub/notes/technote_php_instant.html

Oracle Documentation
http://www.oracle.com/technology/documentation/index.html