



# uniCenta oPOS

---

## Administrator Guide

**Jack Gerrard**  
**September 2009**

This Administrator Guide deals with the setup and configuration of uniCenta oPOS after installation. It is a “How to” of key points for Security, Stock and Receipts to enable a quick customization to suit the users operation.

## Contents

<b>Managing Authentication and Authorisation .....</b>	<b>3</b>
Editing Roles and Permissions .....	3
Sales panel .....	3
Sales panel button .....	3
Payment method .....	3
Customer panel.....	3
Stock Management panel .....	4
Sales Management panel.....	4
Maintenance panel .....	4
Users .....	5
<b>Setting up Stock .....</b>	<b>5</b>
Taxes .....	5
Categories .....	5
Products .....	5
<b>Change the default receipt layout.....</b>	<b>6</b>
Edit the receipt.....	6
Language .....	6
Ticket.....	6
Display.....	7
Editing the image .....	7

# Managing Authentication and Authorisation

---

## Editing Roles and Permissions

Go to **Maintenance** under **Administration** and Select **Roles**.

To edit the permissions you have to create a new **Role** and edit its permissions. The format used is XML which describes the forms used by the **Role** to which it has permission to use.

By default the Administrator **Role** has access to all the application. Use this **Role** as the template for new roles.

Below is the application's list of options you can grant access to a **Role**:

### Sales panel

```
<class name="com.openbravo.pos.sales.JPanelTicketSales"/>
<class name="com.openbravo.pos.sales.JPanelTicketEdits"/>
<class name="com.openbravo.pos.customers.CustomersPayment"/>
<class name="com.openbravo.pos.panels.JPanelPayments"/>
<class name="com.openbravo.pos.panels.JPanelCloseMoney"/>
<class name="/com/openbravo/reports/closedpos.bs"/>
```

### Sales panel button

```
<class name="button.print"/>
<class name="button.opendrawer"/>
```

### Payment method

```
<class name="payment.cash"/>
<class name="payment.cheque"/>
<class name="payment.paper"/>
<class name="payment.magcard"/>
<class name="payment.free"/>
<class name="payment.debt"/>
<class name="refund.cash"/>
<class name="refund.cheque"/>
<class name="refund.paper"/>
<class name="refund.magcard"/>
```

### Customer panel

```
<class name="com.openbravo.pos.forms.MenuCustomers"/>
<class name="com.openbravo.pos.customers.CustomersPanel"/>
<class name="/com/openbravo/reports/customers.bs"/>
<class name="/com/openbravo/reports/customersb.bs"/>
```

## Stock Management panel

```
<class name="com.openbravo.pos.forms.MenuStockManagement"/>
<class name="com.openbravo.pos.inventory.ProductsPanel"/>
<class name="com.openbravo.pos.inventory.ProductsWarehousePanel"/>
<class name="com.openbravo.pos.inventory.CategoriesPanel"/>
<class name="com.openbravo.pos.panels.JPanelTax"/>
<class name="com.openbravo.pos.inventory.StockDiaryPanel"/>
<class name="com.openbravo.pos.inventory.StockManagement"/>
<class name="/com/openbravo/reports/products.bs"/>
<class name="/com/openbravo/reports/productlabels.bs"/>
<class name="/com/openbravo/reports/productscatalog.bs"/>
<class name="/com/openbravo/reports/inventory.bs"/>
<class name="/com/openbravo/reports/inventoryb.bs"/>
<class name="/com/openbravo/reports/inventorybroken.bs"/>
<class name="/com/openbravo/reports/inventorydiff.bs"/>
```

## Sales Management panel

```
<class name="com.openbravo.pos.forms.MenuSalesManagement"/>
<class name="/com/openbravo/reports/usersales.bs"/>
<class name="/com/openbravo/reports/closedproducts.bs"/>
<class name="/com/openbravo/reports/taxes.bs"/>
<class name="/com/openbravo/reports/chartsales.bs"/>
<class name="/com/openbravo/reports/productsales.bs"/>
```

## Maintenance panel

```
<class name="com.openbravo.pos.forms.MenuMaintenance"/>
<class name="com.openbravo.pos.admin.PeoplePanel"/>
<class name="com.openbravo.pos.admin.RolesPanel"/>
<class name="com.openbravo.pos.admin.ResourcesPanel"/>
<class name="com.openbravo.pos.inventory.LocationsPanel"/>
<class name="com.openbravo.pos.mant.JPanelFloors"/>
<class name="com.openbravo.pos.mant.JPanelPlaces"/>
<class name="/com/openbravo/reports/people.bs"/>
<class name="com.openbravo.possync.ProductsSyncCreate"/>
<class name="com.openbravo.possync.OrdersSyncCreate"/>
<class name="Menu.ChangePassword"/>
<class name="com.openbravo.pos.panels.JPanelPrinter"/>
<class name="com.openbravo.pos.config.JPanelConfiguration"/>
```

## Users

Choose **Administration** then **Maintenance** and select **Users**.

To create a new **User**: Click button **New**. Enter the new **User** name in the **Name** field, and then select their **Role**. You want them to be visible, so do not change anything else.

## Setting up Stock

---

### Taxes

Go to **Stock** under **Administration** and Select **Taxes**.

Here you can set up **Tax** bands that can be associated with **Products**. This allows taxation calculations to be performed automatically.

To create a new **Tax** band click on the button **New**. Edit the name and set the rate value. After you are done; click the **Save** button.

### Categories

Go to **Stock** then **Administration** and select **Categories**.

Managing the catalog consists of setting up **Categories** and adding **Products** to those **Categories**. Note: **Products** must belong to one **Category**.

The **Category** form is very much like to the **Taxes** screen. Here you can define several **Categories** under which you can assign your **Products**.

Categories are organised hierarchically. Categories can have **sub-Categories**. To **Add** a **sub-Category**: First select a parent **Category** from the drop down **Category** list.

Optionally you can add a picture for each **Category** by editing the **Image** field. To do this: Select the **folder button**; then **Browse** your files and select the desired image.

You can also **Add** all the **Products** that belong to one **Category** by pressing the button **Add to Catalog** and also remove all of them pressing the button **Remove from Catalog**.

### Products

Select **Stock** under **Administration** and Select **Products**.

The value entered in the **Barcode** field is used to search **Products** when using the barcode reader.

You must set the **Buy Price** and the **Sell Price**. Either enter a **Sell Price**, with or without taxes included, or edit the desired percent **Margin Per Unit**.

Note: Every **Product** must belong to a **Category** and must have a **Tax** band set.

If desired, you can **Add** a picture for each **Product** editing the **Image** field. . To do this: Select the **folder button**; then **Browse** your files and select the desired image.

Select the **Stock** tab if you want to edit the **Stock Cost by Year** and **Stock Volume** values for report purposes.

In the **Properties** tab you can decide if the product will be visible in the **Catalog** panel of the **Sales** panel an whether it is an **Auxiliary Product** and if it is a **Product** which will be sold by weight.

# Change the default receipt layout

---

## Edit the receipt

Advanced configuration of uniCenta oPOS is done via the **Resources** panel rather than with a GUI. A good example of this is for **Receipts**.

To change the default ticket you must go to the **Resources** panel and edit the resource with the name **Printer.Ticket**.

uniCenta oPOS uses Apache Velocity to parse the XML templates. The [reference to Velocity](#) contains further information for the syntax used.

In the receipt there are available all of the public methods of the **net.adrianromero.tpv.ticket.TicketInfo** class in the **\$ticket** variable and the public methods of the **net.adrianromero.tpv.ticket.TicketLineInfo** class in the **\$ticketlines** variable. (Please refer to external Openbravo POS documentation)

## Language

uniCenta oPOS uses the a new language based on XML to print receipts and output to a customer display and independent of the model of printer and customer display used. This language supports one customer display and several printers.

## Ticket

**Ticket** prints a receipt. When it finishes, it cuts the paper. **Ticket** can contain a **printer** attribute which sets the printer.

```
<ticket printer=2> ... </ticket> - the printer 2 will print the receipt.
```

The following elements can be used between **Ticket Tags**:

### Line

This sets a **Text line** in a receipt. It can contain a parameter **Size Setting** to set size of the Text. Possible values are:

- 0** - Normal size (by default)
- 1** - Double height
- 2** - Double weight
- 3** - Double height and weight

```
<line size=3> ... </line>
```

Between **Line Tags** to print text **<text>** tag is required. The **Text Tag** can contain the following parameters:

- align** - defines the Alignment of the text string:
  - left** - align text to the left (Default value)
  - right** - align text to the right
  - center** - sets the text in the centre.
- length** - defines the length in characters used to fit the text strings.
- bold** - defines whether if the text is bold or not. By default false.
- underline** - defines whether if the text is underlined or not. By default false

```
<text align ="right" length="10" underline="true">Some text</text>
```

## Image

Insert an **Image** in the receipt. This is resource name that contains the **Image** file. The **Image** is converted to black and white before sending it to the printer.

```
<image>Your image</image>
```

## Barcode

A **barcode**. The number of the barcode. EAN-13 format is used.

```
<barcode>the number</barcode>
```

## Opendrawer

Opens the Cash-drawer attached to the receipt printer.

```
<opendrawer/>
```

The default ticket template can serve as a good starting point.

## Display

### Line

A **Text Line** in the display. Enter text between **Line Tags** to display text <text> tag. The **Text Tag** can contain the following parameters:

- align** - defines the alignment of the text string
  - left** - align text to the left (Default value)
  - right** - align text to the right
  - center** - sets the text in the centre
- length** - defines the length in characters used to fit the text strings

## Editing the image

The default **Image** can be changed by editing the resource **Printer.Ticket.Logo**. The image types supported vary by printer. Most thermal printers support image printing and most dot-matrix don't. The image must be 256 pixels wide at most.

To remove the **Image** from the ticket you have to edit the **Printer.Ticket** resource and remove the references to the **Printer.Ticket.Logo** resource.