



New window creation tutorial

2nd October 2006

Revision 1.1



openbravoTM
opening ERP's future!

Visit us at www.openbravo.com



Table of Contents

I. Introduction.....	3
II. Create new tables in the Oracle database.....	4
III. Register the new tables in the Openbravo ERP dictionary.....	5
IV. Creating a new Window.....	9
V. Linking a new window to the Openbravo menu.....	14



I. Introduction

Openbravo is a pure web-based extended ERP for SMEs (Small and Midsize Enterprises). The extended ERP solution includes basic CRM (Customer Relationship Management) and BI (Business Intelligence) and a broad range of standard functionalities such as purchasing, warehousing, projects, manufacturing and sales management, financial management and much more. Openbravo is in production in companies spanning several industries, including distribution, services and manufacturing.

The aim of this tutorial is to show how to create a new application window in Openbravo ERP. As an example, we use a new window that allows us to introduce information about documents.

This tutorial assumes that you already have installed and running Openbravo ERP.



II. Create new tables in the Oracle database

First you have to create the new tables in the database engine that Openbravo uses as backend (currently only Oracle is supported). You can use Oracle SQL*Plus application provided with your standard Oracle package.

Once the SQL*Plus is started, you are asked to log in. If you have used the default Openbravo configuration parameters during the installation, use TAD as “username”, TAD as a “password” and ORCL as a host name. You get now SQL prompt that allows you to introduce SQL statements directly.

For this example, we type the following SQL statement to create the documents table that we use in this tutorial:

```
CREATE TABLE DOCUMENTS
(
  DOCUMENTS_ID      NUMBER(10)          NOT NULL,
  AD_CLIENT_ID      NUMBER(10)          NOT NULL,
  AD_ORG_ID         NUMBER(10)          NOT NULL,
  ISACTIVE           CHAR(1 BYTE) DEFAULT 'Y' NOT NULL,
  CREATED            DATE DEFAULT SYSDATE NOT NULL,
  CREATEDBY          NUMBER(10)          NOT NULL,
  UPDATED             DATE DEFAULT SYSDATE NOT NULL,
  UPDATEDBY          NUMBER(10)          NOT NULL,
  NAME               NVARCHAR2(60)       NOT NULL,
  LASTEDITED         DATE               NOT NULL,
  DOCSIZE            NUMBER(15)          NOT NULL
);
```

Once executed, a new table called “Documents” is created in Oracle.

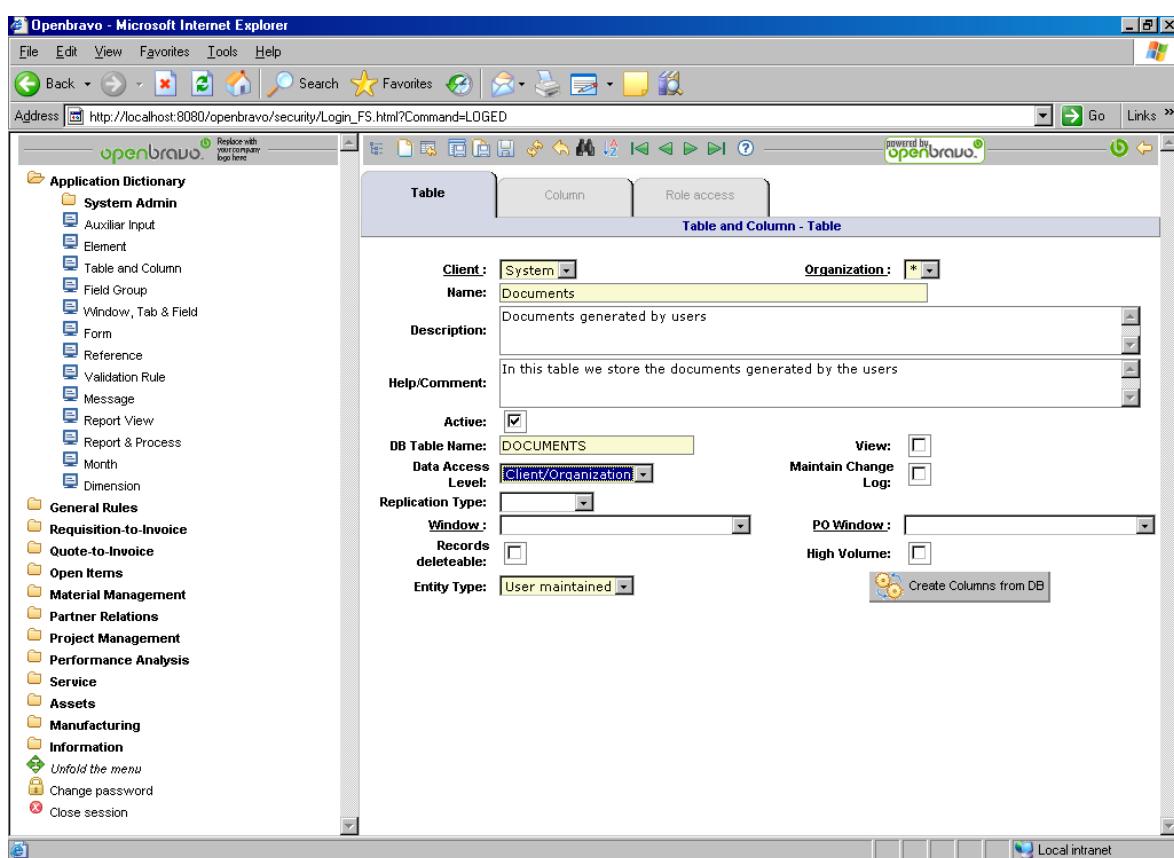
When defining your own tables, it is always necessary to consider that:

- ◆ You always need to define the AD_Client_ID, AD_Org_ID, IsActive, Created, CreatedBy, Updated and UpdatedBy fields as shown in the sample. These fields are used internally by OpenBravo ERP.
- ◆ The primary key of the table is not shown in the application at any moment, therefore although in the example “NAME” and “LASTEDITED” fields could form a major key, we use an extra field that is a unique identifier, in this case DOCUMENTS_ID. It is recommended to name the primary key field using the name of the table plus the _ID suffix.



III. Register the new tables in the Openbravo ERP dictionary

Now you have to register the newly created database tables in the Openbravo ERP dictionary. To achieve this, first log into Openbravo using an account with access to the System Administration role. Now, go to the “Table and Column” option in the “Application directory” menu. Select the “New Record” icon () from the toolbar. The following form is displayed:



Here there is a description of the main fields of this form:

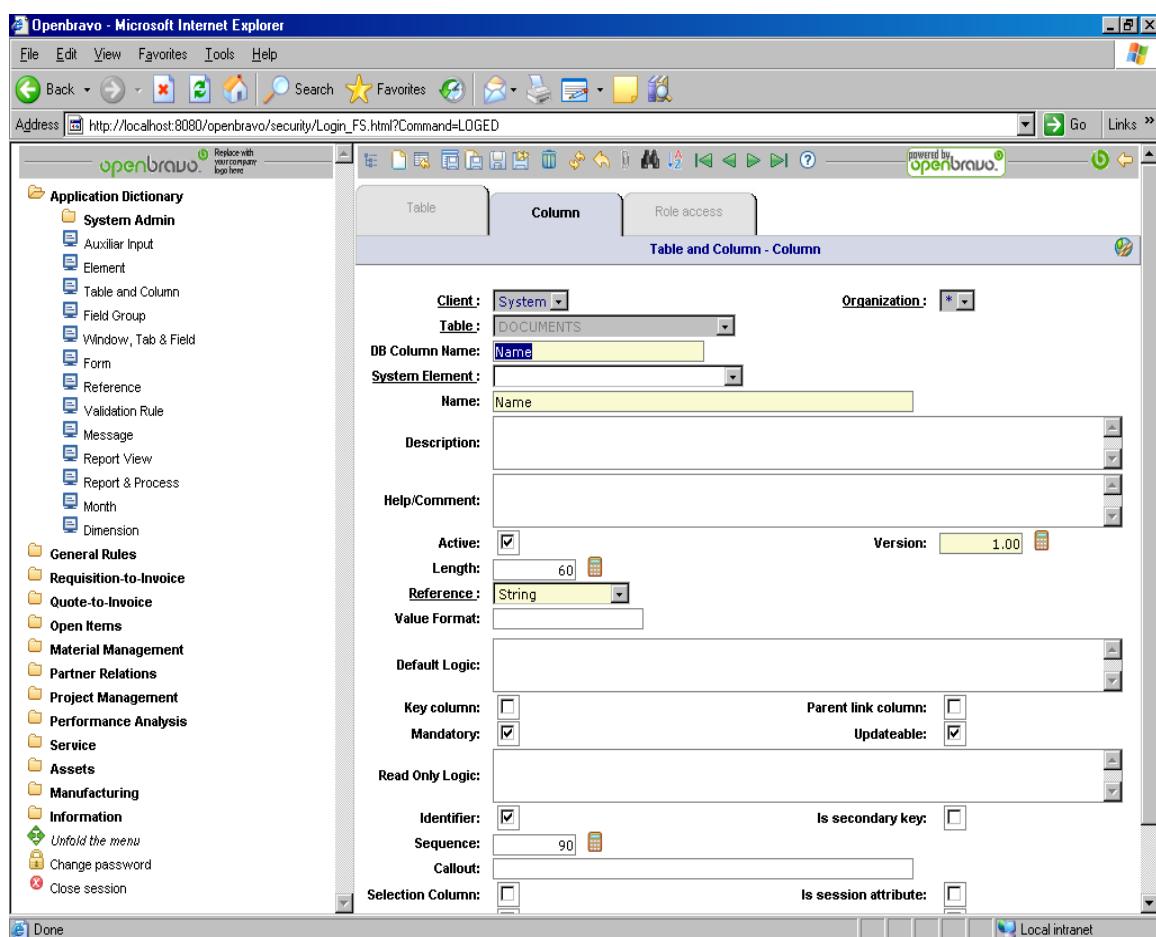
- ◆ **Name.** Defines the name which Openbravo uses to name the defined database table into the Application Dictionary.
- ◆ **Description.** Gives a small description to the table dictionary.
- ◆ **Help/Comments:** Defines the text that is displayed in the help Window.
- ◆ **DB Table name:** Defines the database table name that we have previously created in the database engine.
- ◆ **Entity type.** The type of entity. For our own tables, we should use “user maintained”.



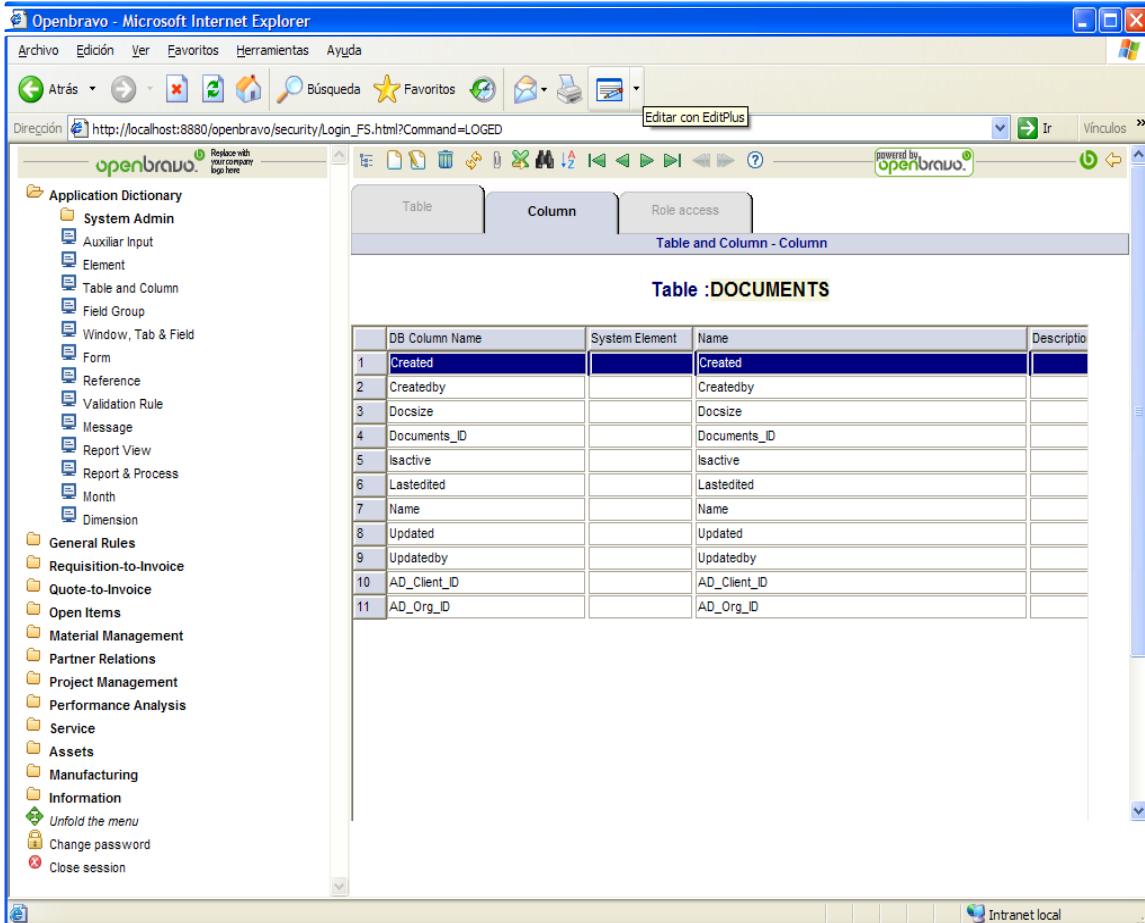
In our example, the name field is “Documents”, the DB table name is “DOCUMENTS” also, and the entity type is set to “user maintained”.

Once you have filled up all the details you should store the new table definition using the save icon () from the toolbar. Now, press the button “Create columns from DB” to create the entries needed to define the entries in the columns. When the creation process is finished, you will be informed of the number of columns that have been added into the application dictionary.

In the “column” tab of this form you can see all the columns defined for the table that we are introducing and customize the properties for each column.



If you want to see a list of all the created columns you can click the Relation icon ().



DB Column Name	System Element	Name	Description
1. Created		Created	
2. Createdby		Createdby	
3. Docsize		Docsize	
4. Documents_ID		Documents_ID	
5. IsActive		IsActive	
6. Lastedited		Lastedited	
7. Name		Name	
8. Updated		Updated	
9. Updatedby		Updatedby	
10. AD_Client_ID		AD_Client_ID	
11. AD_Org_ID		AD_Org_ID	

These columns are assigned a reference (which defines the data type) depending on its name and data type. A System Element is automatically assigned, the name of this element is the same as the column, if exists an element matching this rule is associated otherwise a new one is created.

Note: The columns that are named 'line' or 'seqNo' are used to contain the position in a relation (i.e. the number of a line in an invoice). They take a *Default Value* like

```
@SQL=SELECT COALESCE(MAX(ColumnName),0)+10 AS DefaultValue  
FROM TableName WHERE xxParentColumn=@xxParentColumn@
```

It is necessary to replace the *WHERE* clause of this value (the red piece of code). The code that should appear here is the name of the column which links with the id of the parent one, i.e. for the previously mentioned invoice there is a table named C_Invoice containing the header of the invoices which is the parent table for the lines saved in C_InvoiceLine, this second table has a column named *line* and the default value that it takes is:

```
@SQL=SELECT COALESCE(MAX(LINE),0)+10 AS DefaultValue FROM  
C_INVOICELINE WHERE xxParentColumn=@xxParentColumn@
```



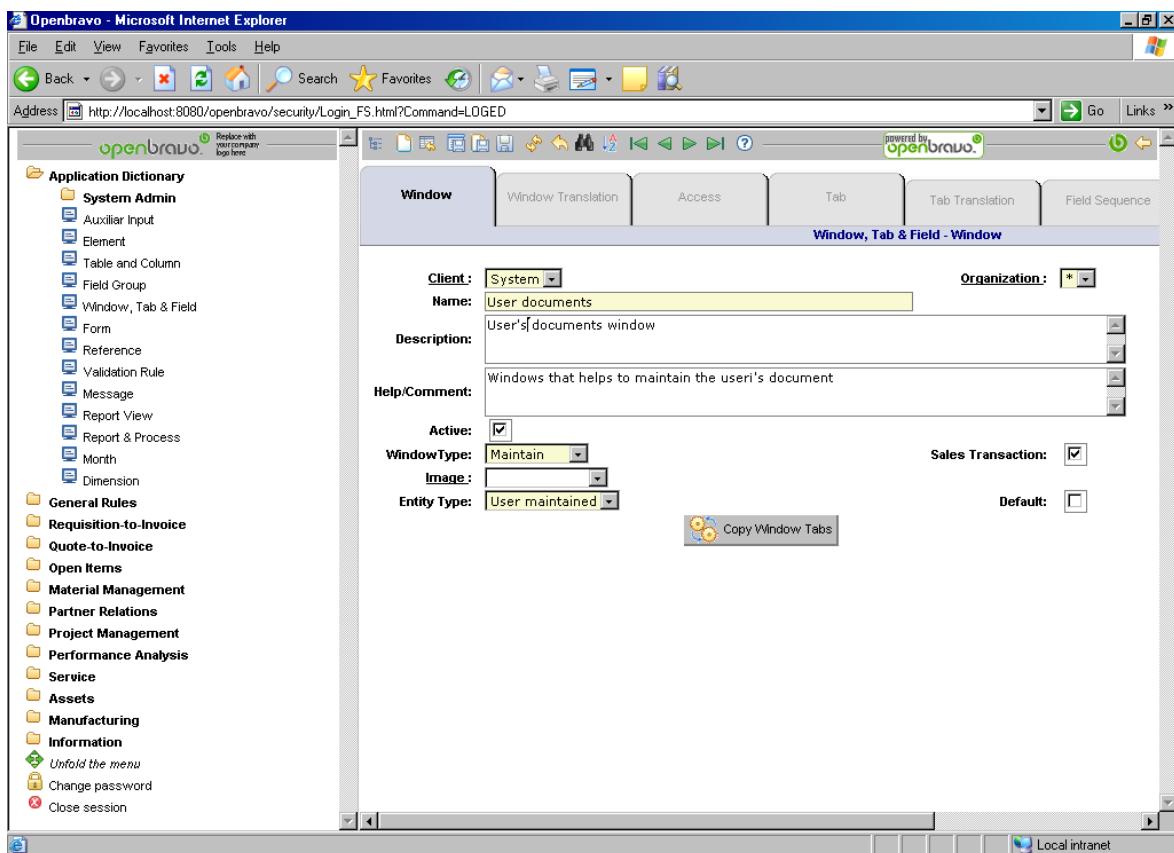
This should be modified to:

```
@SQL=SELECT COALESCE(MAX(LINE),0)+10 AS DefaultValue FROM  
C_INVOICELINE WHERE C_INVOICE_ID=@C_INVOICE_ID@
```



IV. Creating a new Window

In order to create a new window you first should log into Openbravo using an account with access to the System Administration role. Go to the “Window, Tab Field” option under the “System Admin” menu and select the “New Record” icon (New Record icon) from the toolbar. The following form is displayed:



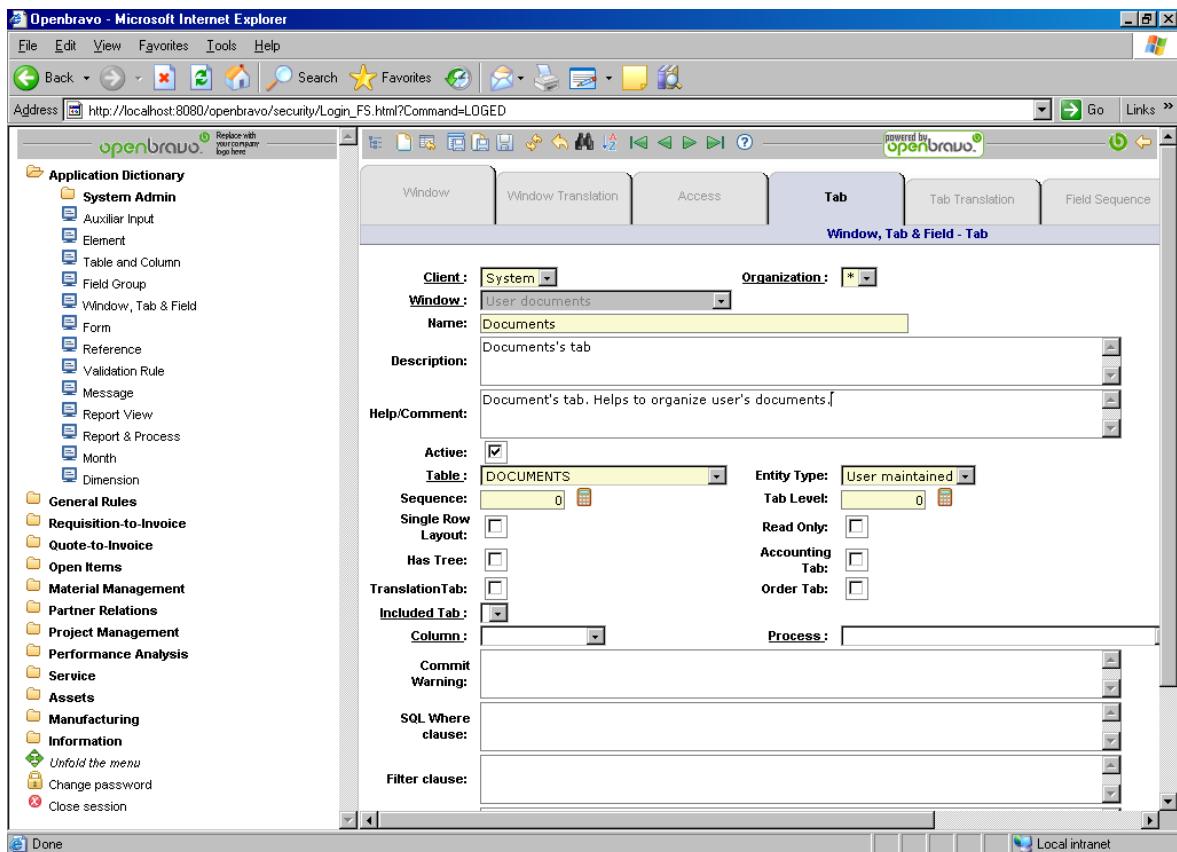
Here there is a description of the main fields of the window form definition:

- ◆ **Name.** Defines the name which Openbravo uses to name this window.
- ◆ **Description.** Gives a small description to the window.
- ◆ **Help/Comments:** Defines the text that is displayed in the help Window.

In our example, the name field is “User documents”.

Once you have filled up all the details you should store the new window definition using the save icon (Save icon) from the toolbar.

Now, you can select the “Tab” option, and you get the following form:



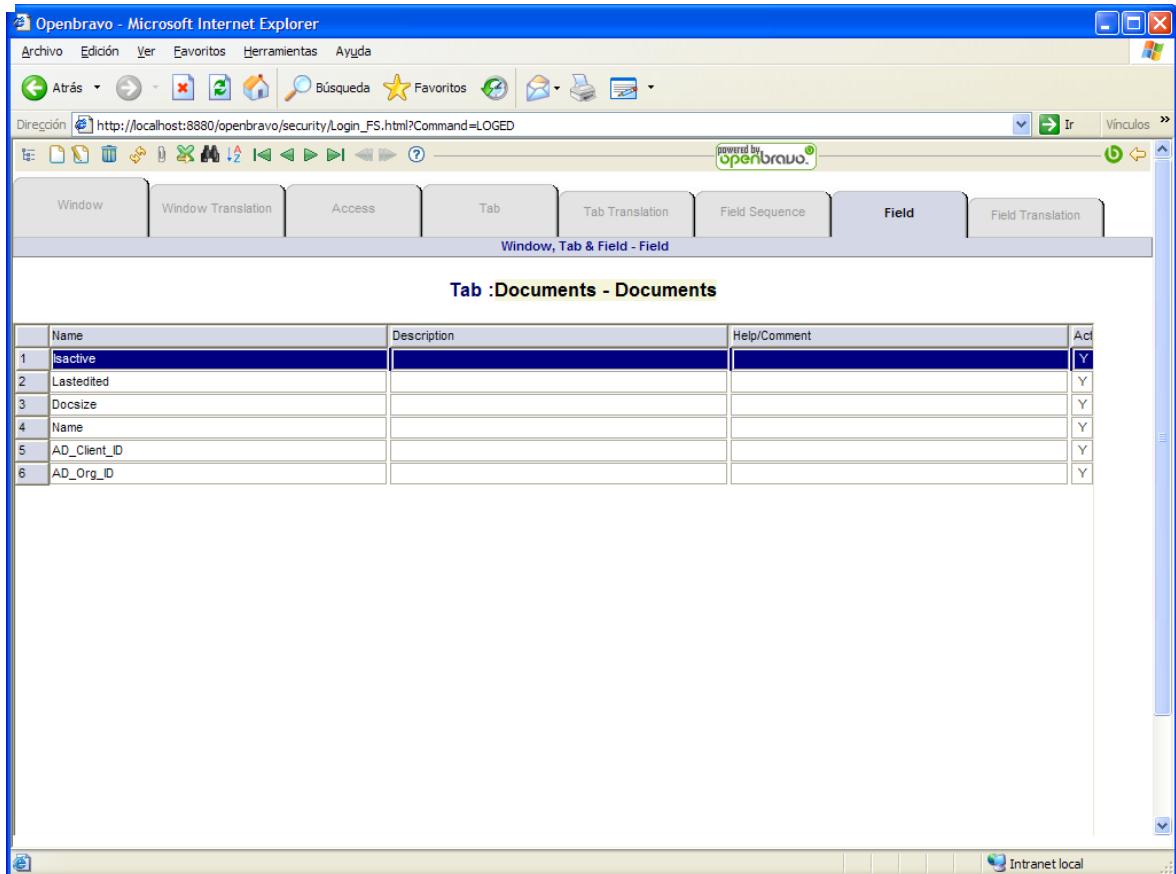
Here there is a description of the main fields of the tab form definition:

- ◆ **Name.** Defines the name which Openbravo uses to name this tab.
- ◆ **Description.** Gives a small description to the window.
- ◆ **Help/Comments:** Defines the text that is displayed in the help Window.
- ◆ **Table.** Establishes a relation between this tab and the database table.

In our example, the name field is “Documents” and we select the table name “Documents”.

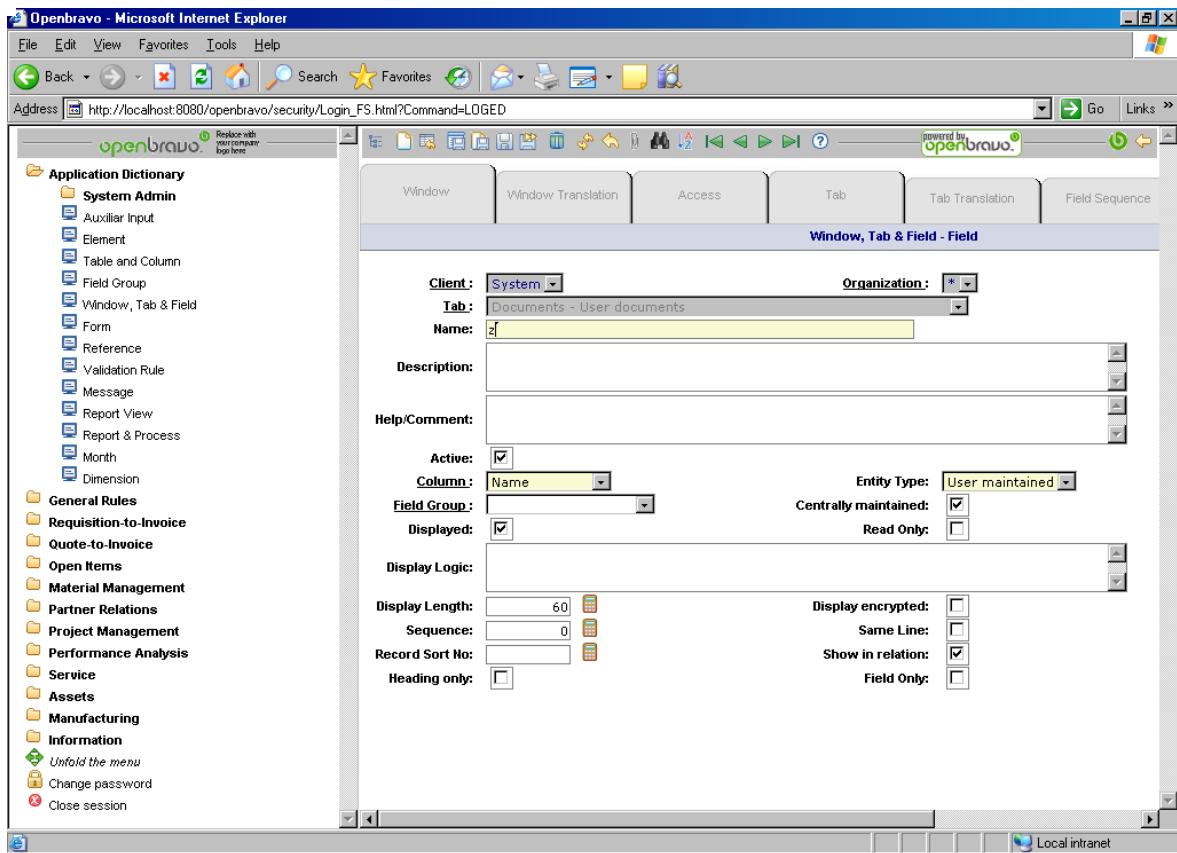
Once you have filled up all the details you should store the new tab definition using the save icon (H) from the toolbar. Now, press the button “Create fields”. When the creation process is finished, you will be informed of the number of fields that have been added into the dictionary.

You can look the fields that have been created clicking the “Field” tab.



	Name	Description	Help/Comment	Act
1	Isactive			Y
2	Lastedited			Y
3	Docsizes			Y
4	Name			Y
5	AD_Client_ID			Y
6	AD_Org_ID			Y

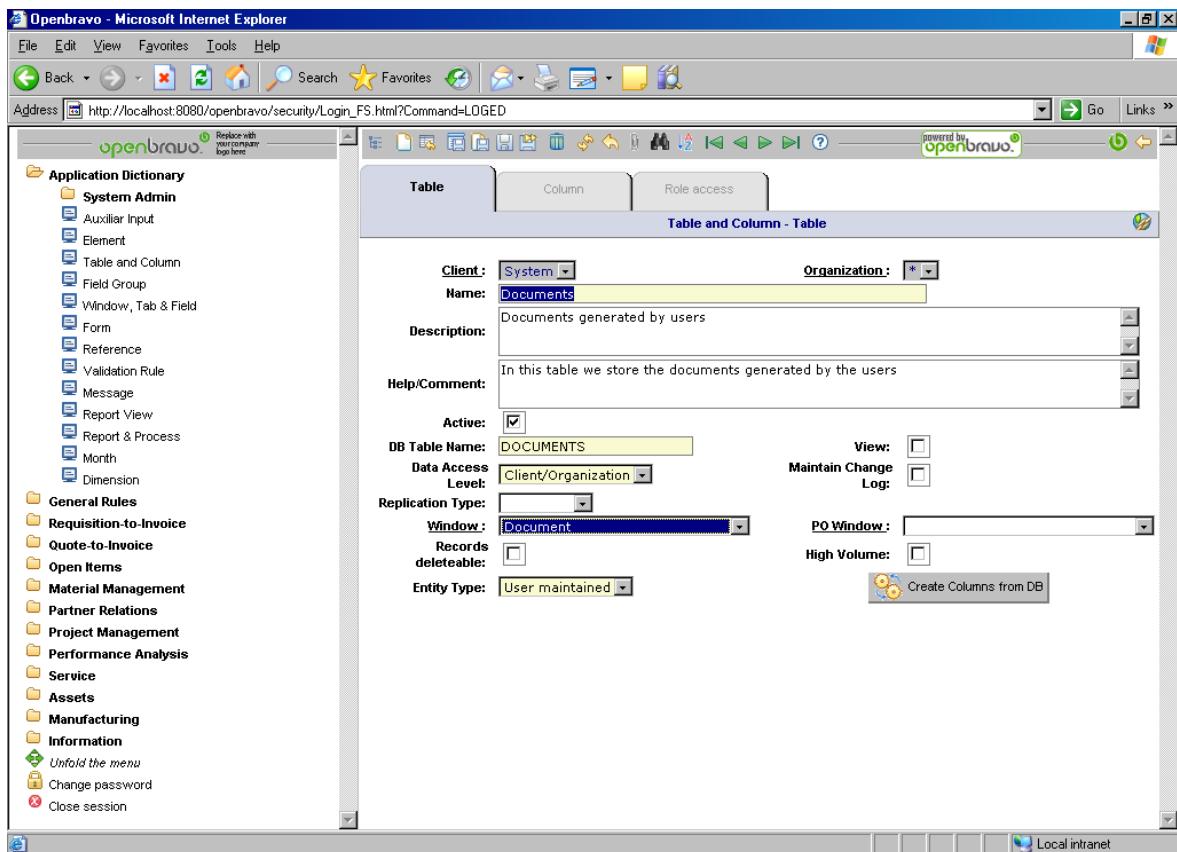
You can customize the created fields using the field tab. Once, you select the field to customize the following form will be displayed:



Here there is a description of the main fields of the columns form definition:

- ◆ **Name.** Defines the name which Openbravo uses to name this field.
- ◆ **Description.** Gives a small description to the field.
- ◆ **Help/Comments:** Defines the text that is displayed in the help Window.
- ◆ **Table.** Establishes a relation between this tab and the database table.

Finally, to end the process we have to associate the window that we have created to the table definition that we previously did. We have to go to the “Table and column” option under the “System Admin” menu and select the Table tab. Edit the “Documents” table and associate the window “document” to this table definition, as seen in the following screen capture.



To finish correctly the process of window creation it is necessary to compile the created window. To do that we have to write a command from the application root directory (AppsOpenbravo by default).

```
ant compile -Dtab=WindowName
```

(WindowName is the name of a created window).

- ◆ Once finished the compilation we have to restart Tomcat.



V. Linking a new window to the Openbravo menu

In order to link a new window to the Openbravo menu you first should log into Openbravo using an account with access to the System Administration role. Go to the “System Rules” option under the “General Rules” menu and select the option “Menu”. Now, select the “New Record” icon (New Record icon) from the toolbar. You get the following form:

The screenshot shows the Openbravo System Rules - Menu form. The left sidebar lists various system rule categories. The main form has the following fields:

- Name:** Document
- Description:** Document menu
- Active:**
- Entity Type:** User maintained
- Action:** Window
- Window:** Document

Here there is a description of the main fields of the Menu form definition:

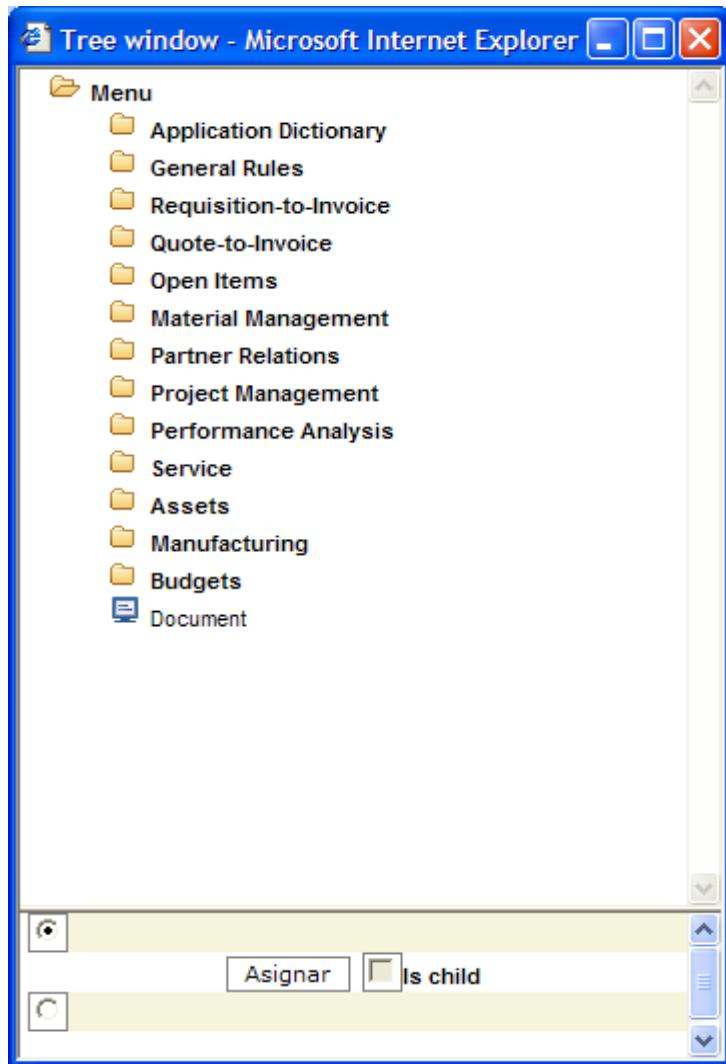
- ◆ **Name.** Defines the name which Openbravo uses to name this menu.
- ◆ **Description.** Gives a small description to the menu.
- ◆ **Action.** Indicates the action that this menu should execute when selected.
- ◆ **Window/Form/Process/Task/Report/WorkBench/Workflow.** Indicates the name of the resource associated to the action.

In our example, the name field is “Documents”, the action is “Window” and the “Window” field is “User documents”.

Once you have filled up all the details you should store the new menu definition



using the save icon () from the toolbar. Now, you have to assign the new menu entry to the current menu hierarchy. Use the tree icon () from the toolbar. The following form is displayed.



Now, you have to click on the menu entry to which you want to associate the newly created menu entry, while the button on the top is selected. Check the lower radio button and click on the recently created window icon. Finally check the "Is child" check box and click on the "Assign" button.



© Openbravo S.L. 2006

This work is licensed under the Creative Commons Attribution-ShareAlike 2.5 Spain License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/2.5/es/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

As attribution to the original author, any redistribution of this work or any derivative work must maintain this copyright notice and, visibly on all its pages, the Openbravo logo.

The most updated copy of this work may be obtained at <http://www.openbravo.com/docs/>

