# 

# EXTENDED DEMOGRAPHY

AMXEDITOR – PLUGIN

EXTENDED DEMOGRAPHY 1

1. Introduction. 2

2. AD\_Reference Table. 4

3. New Messages on AD\_Message Table. 6

4. Fields Added on C\_Country Table. 7

5. Capture Sequence on Table C\_Country. 8

6. New Table C\_Community 9

7. New Tables C\_Municipality and C\_Parish 11

8. Fields Added on C\_Region Table. 14

9. Fields Added on C\_Location. 16

10.Country-Community-Region-Municipality-Parish Window. 17

11.Business Partner Location Extended Example. 19

A1. Plugin Installation. 21

Author : Luis Amesty Linares

Madrid, Spain August 2015

Revision October 2019

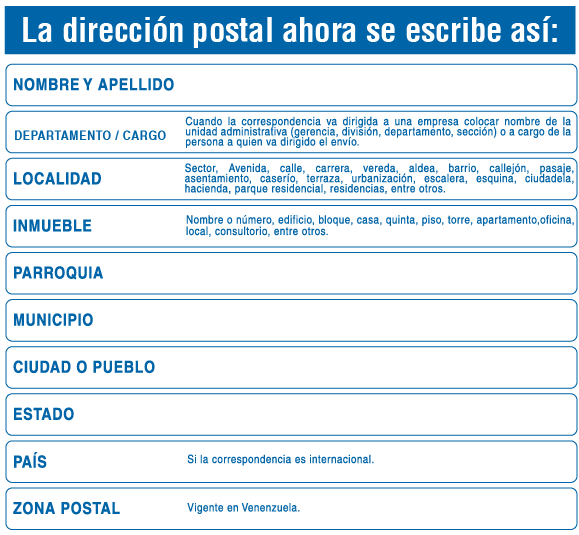
[luisamesty@gmail.com](mailto:luisamesty@gmail.com)

# 1. Introduction.

This Plugin is related to Demographic Aspects on Idempiere ERP. It is well known that on some countries there is a political division with concepts that go beyond country, region and city.

For example, in Spain we talk about Communities, they are related to Group of Provinces that keeps Local Government. They also have a political division for Provinces called Municipalities (Municipality-*Municipio*).

Another example is Venezuela, they have a political division for Provinces (*Estados*) called Municipalities (Municipality-*Municipio*) and a Municipality is also divided on Parishes (Parish-*Parroquia*). Particularity Main Mail office in Venezuela has published a new rule indicating the way postal address is to be written:



Communities, Region, Municipality and Parish, can be use for statistic and BI analysis on Sales, Cost, Personnel and other type of transaction results.

Postal Codes is another concern to be covered in future on this plugin, because a default value can be obtained from Region, Provinces, Municipality and Parish. Also Street Names provides a rule for finding zip codes.

In order to accomplish what have been explained before, some tables have to be modified, and some others tables have to be added to implement the new Window and plugin.

|  |  |
| --- | --- |
| **Tables** | |
| **Modify** | **Add** |
| *C\_Country (2)* | *C\_Municipality* |
| *AD\_Message (1)* | *C\_Parish* |
| *C\_Location (2)* | *C\_Zipcode* |
| *C\_Region (2)* | *C\_Community* |
| *AD\_Reference (1)* |  |

(1) Add New Records (2) Add New Fields

I have been working with Idempiere since version 1.0c, before I made some test on Adempiere 3.70. Right now I am migrating my old Visual FoxPro application with one of my customer using Idempiere 2.0 in VENEZUELA.

A list of Lectures and References related with this Material. Some information, procedures, code and data taken from:

Wiki Page: **Developing plug-ins without affecting the trunk**

<http://wiki.idempiere.org/en/Developing_plug-ins_without_affecting_the_trunk>

Jan Thielemann Videos in special:

Developing iDempiere 2.0 - Custom Editors (IEditorFactory and IDisplayTypeFactory)

<https://www.youtube.com/watch?v=jVhKnFzsMZo>

DCS-ERP Repository from Orlando Curieles.

<https://bitbucket.org/dcs_bitbucket/>

Initial Venezuelan Demography.

Redhuan Daniel Oon (red1) Repository.

<https://bitbucket.org/red1/>

Ghintech Location.

<https://bitbucket.org/ghintech/com.ghintech.location>

Carlos Ruiz (CarlosRuiz\_globalqss)

<https://bitbucket.org/CarlosRuiz_globalqss/>

Plugin Source code, documentation and jar files are located on Bitbucket repository:

<https://bitbucket.org/amerpsoft/amerpsoft-editors-com/src/default/>

You may clone it with mercurial:

$ hg clone https://bitbucket.org/amerpsoft/amerpsoft-editors-com/src/default/

I really appreciate all material, readings and information provided by forum, wiki pages and repositories. This was my fist little contribution to the group and I hope it will be useful to you.

# 2. AD\_Reference Table.

***AD\_Reference*** Table, it is an Application Dictionary’s Table that contains all Small List arrays associated with fields, where it is necessary to select from Short Length Selection List. It is used instead of a Table. Normally Tables are used when Selection contains Large List Counts of records.

When defining Tables on Application Dictionary, *AD\_Reference* table is used for selection of Field Type on Table’s Columns. For example, we can see records among others:

|  |  |
| --- | --- |
| **Name** | **Description** |
| …. | …. |
| Image | Binary Image Data |
| Integer | 10 Digit numeric |
| List | Reference List |
| Location (Address) | Location/Address |
| **LocationExtended** | **Location Extended /Address with Parish and Municipality** |

Location Reference Type is related with Address Fields, normally associated with *C\_Location* table records thru *C\_Location\_ID* field. This reference selects the type of editor associated with the Table’s Field on AD Model. For example:

Integer: is associated with a plain integer number editor.

Date: is associated with a Date Format (MM/DD/YYYY) editor.

Location: is associated with Address Location Field (see Figure 2.1)

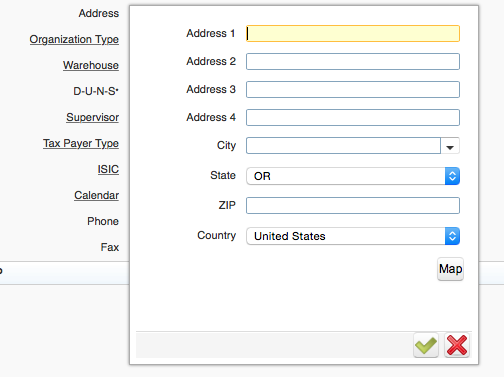


Figure 2.1

***LocationExtended*** Is the name of the new reference to be added to Application Dictionary in order to run Plugin. It must to be defined exactly with this name, because plugin code will find on *AD\_Reference* table as it.

When Plugin be completed and running, it can be associated to the *C\_Location\_ID* field on C\_*BPartnerLocation* Table to a *LocationExtended* reference, the address form will be presented as figure 2.2:

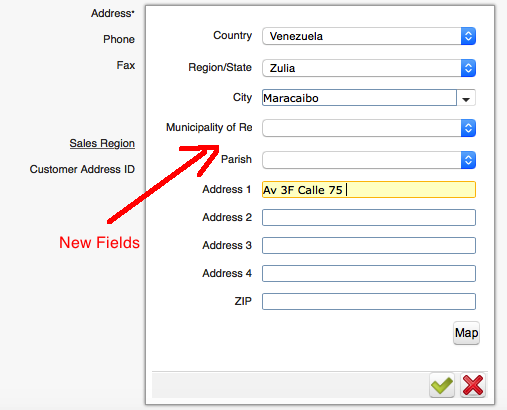


Figure 2.2

# 3. New Messages on AD\_Message Table.

New Messages translations are required to be presented on Plugin Forms. Add new messages Community, Municipality, Parish and Zipcode, and their respective translation to your Localization Language. It must be done using AD window Message: (See examples)

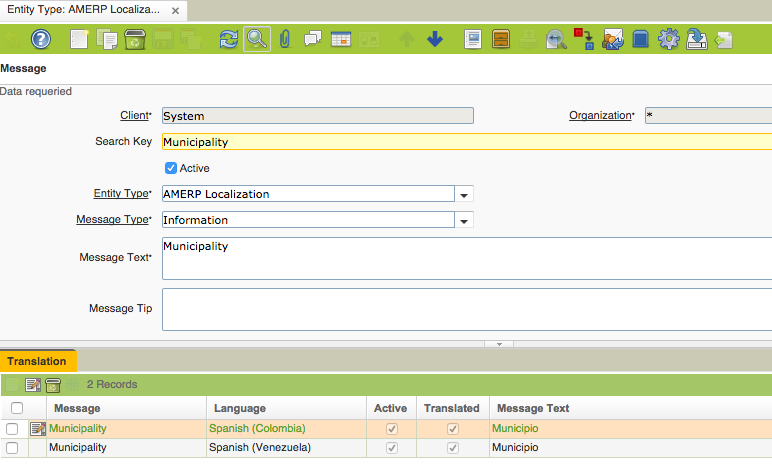


Figure 3.1

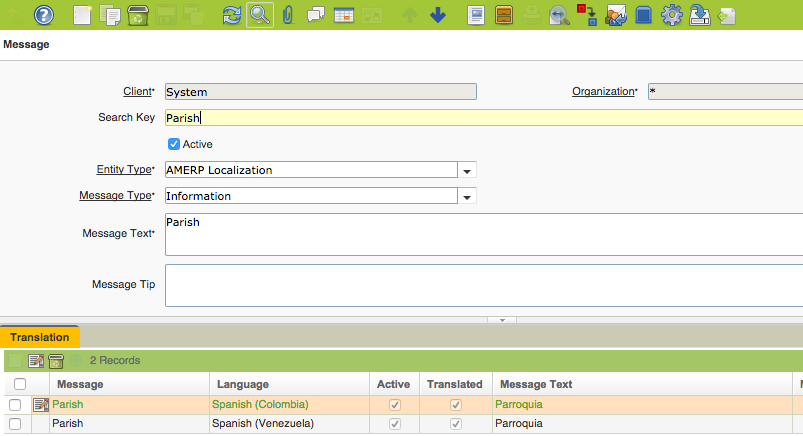


Figure 3.2

# 4. Fields Added on C\_Country Table.

C\_Country table contains Records for Countries. Some fields have been added to this table, and they have to be done using Application Dictionary. The fields added are:

|  |  |  |
| --- | --- | --- |
| **System Element** | **DB Column Name** | **Description** |
| CountryCode3 (\*) | CountryCode3 | Country Code 3 Digits Standard |
| HasCommunity | HasCommunity | Country is divided in Communities Y-N |
| HasMunicipality | HasMunicipality | Country Regions are divided in Municipalities Y-N |
| HasParish | HasParish | Country Regions are divided in Municipalities and Parish Y-N |

In some Countries the appropriate translation for Region differs, for example:

Colombia: *Departamento*.

Venezuela: *Estado*.

USA: *Estado*.

España: *Provincia*.

Take note for this, when create Field Elements and their translation:

(\*) For CountryCode3 inclusion, see Script ‘amxeditor.sql’, included on Sources.

# 5. Capture Sequence on Table C\_Country.

Capture Sequence indicates the order on which Address Fields are presented to the user when creating or modifying address fields (Location or LocationExtended).

Tokens are surrounded by @ character.

For example:

@CO@ @R@ @C@ @A1@ @A2@ @A3@ @A4@ @P@

Available Tokens:

CO: Country

R: Region

C: City

A1, A2, A3, A4: Four Address Lines.

P: Postal Code

Added Tokens to Plugin:

MU: Municipality

PA: Parish

Idempiere Administrator could change this Value in order to modify the Form’s Field Sequence on Window’s Address Fields.

Recommended sequence:

@CO@ @A1@ @A2@ @A3@ @A4@ @P@ @R@ @C@ @MU@ @PA@

Mandatory Fields can be identified by the symbol ***‘!’***.

For example:

@CO@ @A1!@ @A2!@ @A3@ @A4@ @P@ @R!@ @C!@ @MU@ @PA@

Address 1, Address 2, Region, City are Mandatory Fields.

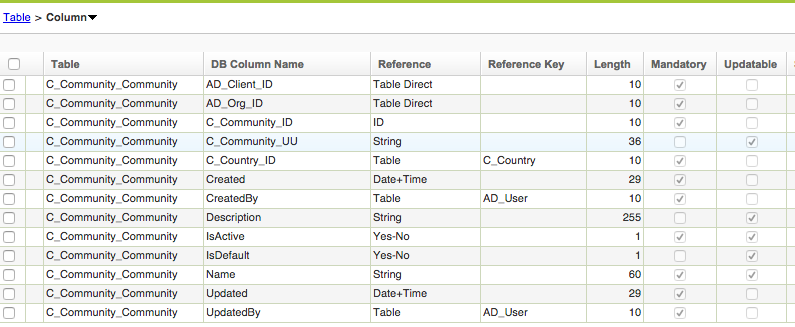
# 6. New Table C\_Community

***C\_Community*** Table contains the Community Political Territorial Division of the Country. As it will be used for Key reference on *C\_Region* table, at least one record for each Country must be created. A SQL file is provided (See amxeditor.sql).

Create Table using PostgreSQL pgAdmin3 app, with this Script:

|  |
| --- |
| /\* CREATE TABLE c\_community (COUNTRY’S COMMUNITIES ) \*/ |
| CREATE TABLE "adempiere"."c\_community" ( |
| "c\_community\_id" NUMERIC(10,0) NOT NULL, |
| "ad\_client\_id" NUMERIC(10,0) NOT NULL, |
| "ad\_org\_id" NUMERIC(10,0) NOT NULL, |
| "isactive" CHAR(1) DEFAULT 'Y'::bpchar NOT NULL, |
| "created" TIMESTAMP WITHOUT TIME ZONE DEFAULT now() NOT NULL, |
| "createdby" NUMERIC(10,0) NOT NULL, |
| "updated" TIMESTAMP WITHOUT TIME ZONE DEFAULT now() NOT NULL, |
| "updatedby" NUMERIC(10,0) NOT NULL, |
| "name" VARCHAR(60) NOT NULL, |
| "description" VARCHAR(255), |
| "c\_country\_id" NUMERIC(10,0) NOT NULL, |
| "isdefault" CHAR(1) DEFAULT 'N'::bpchar, |
| CONSTRAINT "c\_community\_pkey" PRIMARY KEY("c\_community\_id"), |
| CONSTRAINT "c\_community\_isactive\_check" CHECK (isactive = ANY (ARRAY['Y'::bpchar, 'N'::bpchar])), |
| CONSTRAINT "c\_community\_isdefault\_check" CHECK (isdefault = ANY (ARRAY['Y'::bpchar, 'N'::bpchar])), |
| CONSTRAINT "c\_communityclient" FOREIGN KEY ("ad\_client\_id") |
| REFERENCES "adempiere"."ad\_client"("ad\_client\_id") |
| ON DELETE NO ACTION |
| ON UPDATE NO ACTION |
| DEFERRABLE |
| INITIALLY DEFERRED, |
| CONSTRAINT "c\_communityorg" FOREIGN KEY ("ad\_org\_id") |
| REFERENCES "adempiere"."ad\_org"("ad\_org\_id") |
| ON DELETE NO ACTION |
| ON UPDATE NO ACTION |
| DEFERRABLE |
| INITIALLY DEFERRED, |
| CONSTRAINT "ccountry\_cregion" FOREIGN KEY ("c\_country\_id") |
| REFERENCES "adempiere"."c\_country"("c\_country\_id") |
| ON DELETE NO ACTION |
| ON UPDATE NO ACTION |
| DEFERRABLE |
| INITIALLY DEFERRED |
| ) WITHOUT OIDS; |

See Application Dictionary Window for Table C\_Community:



See Script ‘amxeditor.sql’, included on Sources.

# 7. New Tables C\_Municipality and C\_Parish

Some countries divide Region on Municipality entities.

This is the Script for create the table.

--- C\_ Municipality TABLE (c\_municipality)

CREATE TABLE "adempiere"."c\_municipality" (

"c\_municipality\_id" NUMERIC(10,0) NOT NULL,

"ad\_client\_id" NUMERIC(10,0) NOT NULL,

"ad\_org\_id" NUMERIC(10,0) NOT NULL,

"isactive" CHAR(1) DEFAULT 'Y'::bpchar NOT NULL,

"created" TIMESTAMP WITHOUT TIME ZONE DEFAULT now() NOT NULL,

"createdby" NUMERIC(10,0) NOT NULL,

"updated" TIMESTAMP WITHOUT TIME ZONE DEFAULT now() NOT NULL,

"updatedby" NUMERIC(10,0) NOT NULL,

"name" VARCHAR(60) NOT NULL,

"capital" VARCHAR(60) NOT NULL,

"c\_country\_id" NUMERIC(10,0),

"c\_region\_id" NUMERIC(10,0),

"isdefault" CHAR(1) DEFAULT 'Y'::bpchar,

CONSTRAINT "c\_municipality\_pkey" PRIMARY KEY("c\_municipality\_id"),

CONSTRAINT "c\_municipality\_isactive\_check" CHECK (isactive = ANY (ARRAY['Y'::bpchar, 'N'::bpchar])),

CONSTRAINT "c\_municipalityclient" FOREIGN KEY ("ad\_client\_id")

REFERENCES "adempiere"."ad\_client"("ad\_client\_id")

ON DELETE NO ACTION

ON UPDATE NO ACTION

DEFERRABLE

INITIALLY DEFERRED,

CONSTRAINT "c\_municipalityorg" FOREIGN KEY ("ad\_org\_id")

REFERENCES "adempiere"."ad\_org"("ad\_org\_id")

ON DELETE NO ACTION

ON UPDATE NO ACTION

DEFERRABLE

INITIALLY DEFERRED,

CONSTRAINT "ccountry\_cmunicipality" FOREIGN KEY ("c\_country\_id")

REFERENCES "adempiere"."c\_country"("c\_country\_id")

ON DELETE NO ACTION

ON UPDATE NO ACTION

DEFERRABLE

INITIALLY DEFERRED,

CONSTRAINT "cregion\_cmunicipality" FOREIGN KEY ("c\_region\_id")

REFERENCES "adempiere"."c\_region"("c\_region\_id")

ON DELETE NO ACTION

ON UPDATE NO ACTION

DEFERRABLE

INITIALLY DEFERRED

) WITHOUT OIDS;

Note:

PRIMARY KEY for C\_Municipality Table is c\_municipality\_id.

Some countries also, divide Municipality on Parish entities.

This is the Script for create the table.

CREATE TABLE "adempiere"."c\_parish" (

"c\_parish\_id" NUMERIC(10,0) NOT NULL,

"ad\_client\_id" NUMERIC(10,0) NOT NULL,

"ad\_org\_id" NUMERIC(10,0) NOT NULL,

"isactive" CHAR(1) DEFAULT 'Y'::bpchar NOT NULL,

"created" TIMESTAMP WITHOUT TIME ZONE DEFAULT now() NOT NULL,

"createdby" NUMERIC(10,0) NOT NULL,

"updated" TIMESTAMP WITHOUT TIME ZONE DEFAULT now() NOT NULL,

"updatedby" NUMERIC(10,0) NOT NULL,

"name" VARCHAR(60) NOT NULL,

"c\_country\_id" NUMERIC(10,0) NOT NULL,

"c\_region\_id" NUMERIC(10,0) NOT NULL,

"c\_municipality\_id" NUMERIC(10,0) NOT NULL,

"isdefault" CHAR(1) DEFAULT 'N'::bpchar,

CONSTRAINT "c\_parish\_pkey" PRIMARY KEY("c\_parish\_id"),

CONSTRAINT "c\_parish\_isactive\_check" CHECK (isactive = ANY (ARRAY['Y'::bpchar, 'N'::bpchar])),

CONSTRAINT "c\_parishclient" FOREIGN KEY ("ad\_client\_id")

REFERENCES "adempiere"."ad\_client"("ad\_client\_id")

ON DELETE NO ACTION

ON UPDATE NO ACTION

DEFERRABLE

INITIALLY DEFERRED,

CONSTRAINT "c\_parishorg" FOREIGN KEY ("ad\_org\_id")

REFERENCES "adempiere"."ad\_org"("ad\_org\_id")

ON DELETE NO ACTION

ON UPDATE NO ACTION

DEFERRABLE

INITIALLY DEFERRED,

CONSTRAINT "ccountry\_cparish" FOREIGN KEY ("c\_country\_id")

REFERENCES "adempiere"."c\_country"("c\_country\_id")

ON DELETE NO ACTION

ON UPDATE NO ACTION

DEFERRABLE

INITIALLY DEFERRED,

CONSTRAINT "cmunicipality\_cparish" FOREIGN KEY ("c\_municipality\_id")

REFERENCES "adempiere"."c\_municipality"("c\_municipality\_id")

ON DELETE NO ACTION

ON UPDATE NO ACTION

DEFERRABLE

INITIALLY DEFERRED,

CONSTRAINT "cregion\_cparish" FOREIGN KEY ("c\_region\_id")

REFERENCES "adempiere"."c\_region"("c\_region\_id")

ON DELETE NO ACTION

ON UPDATE NO ACTION

DEFERRABLE

INITIALLY DEFERRED

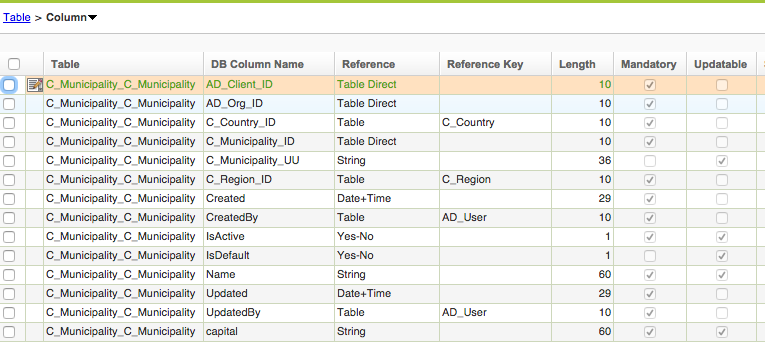
) WITHOUT OIDS;

Note:

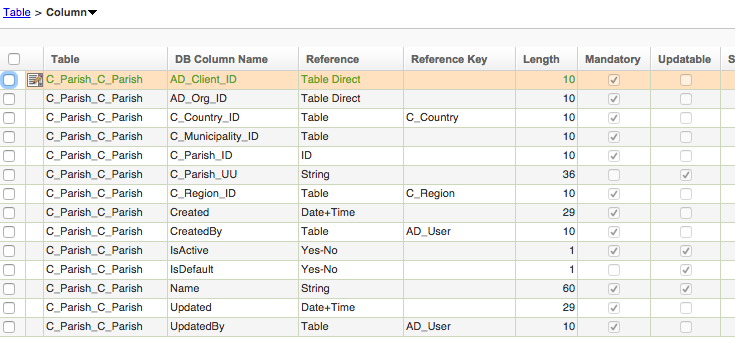
PRIMARY KEY for C\_Parish Table is c\_parish\_id.

Finally take care of Adequate Name and Description on AD using Upper and Lower Cases on AD Tables.

C\_Municipality Table looks like this on AD:



C\_Parish Table looks like this on AD:



# 8. Fields Added on C\_Region Table.

The table *C\_Region* is the second level on demographic structure. On this new proposal *C\_Community* is the New second level, nevertheless in order to be compatible with Standard Address structure on Idempiere, Community an Region are both Second Level, but on *C\_Region* table a Community ID has to be indicated if Idempiere Administrator desire to use it.

See Figure 8.1 For the New Demographic Structure of Plugin:

C\_Country

C\_Region

C\_Community

C\_City

C\_Municipality

C\_Zipcode (\*)

C\_Parish

Figure 8.1

(\*) Future Development

A new Field (*C\_Community\_ID*) is added to *C\_Region* Table. This Field Accept Null values, because some country may not use this feature. See Figure 8.2.

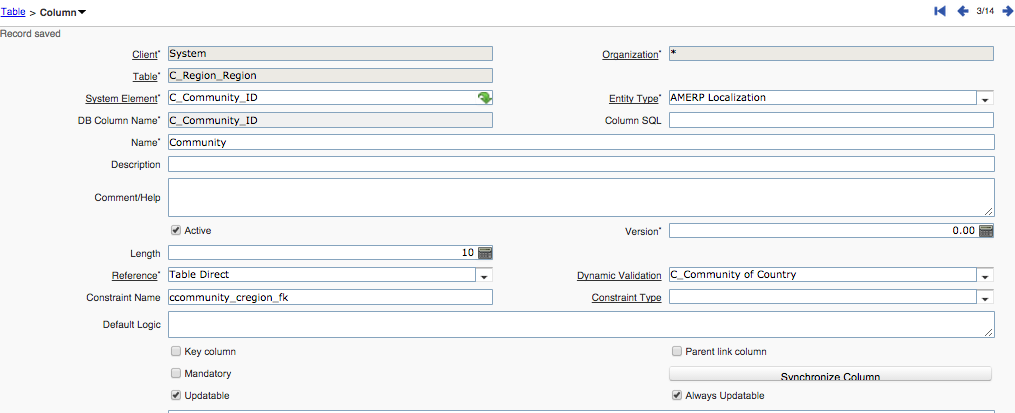


Figure 8.2

A new Validation Rule is created in order to accept only Communities from selected Country. See figure 8.3.

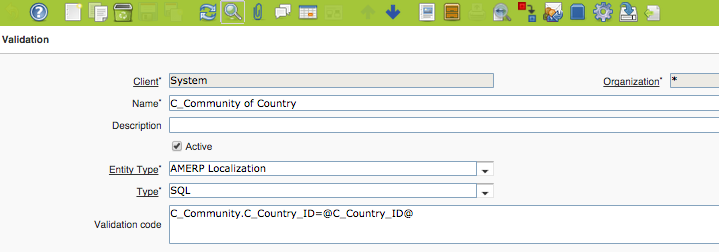


Figure 8.3

# 9. Fields Added on C\_Location.

*C\_Location* Table contains the all addresses included on Business Partners, Employees, Wharehouses, Organization Info, and any other table containing **Location** or **LocationExtended** Reference Fields.

New fields added are *C\_Municipalty\_ID* and *C\_Parish\_ID*, both have been mentioned before. See Figures 9.1 and 9.2 for AD window on Table C\_Location.

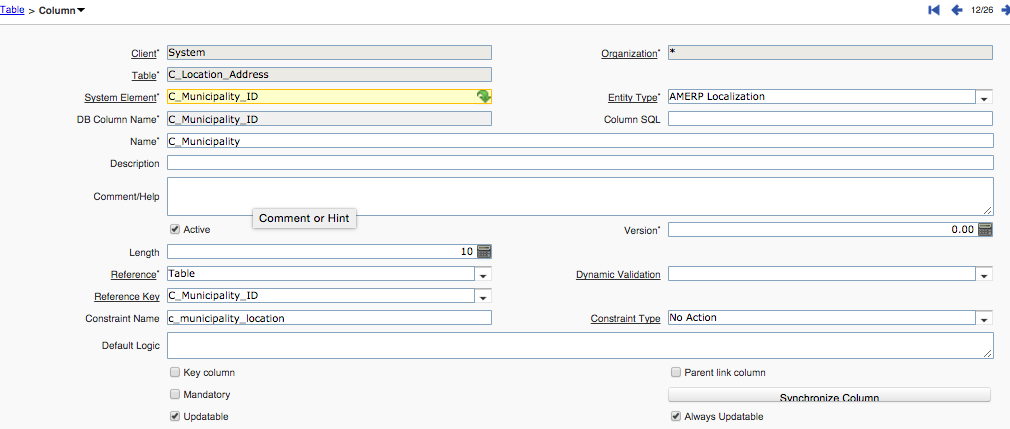


Figure 9.1.

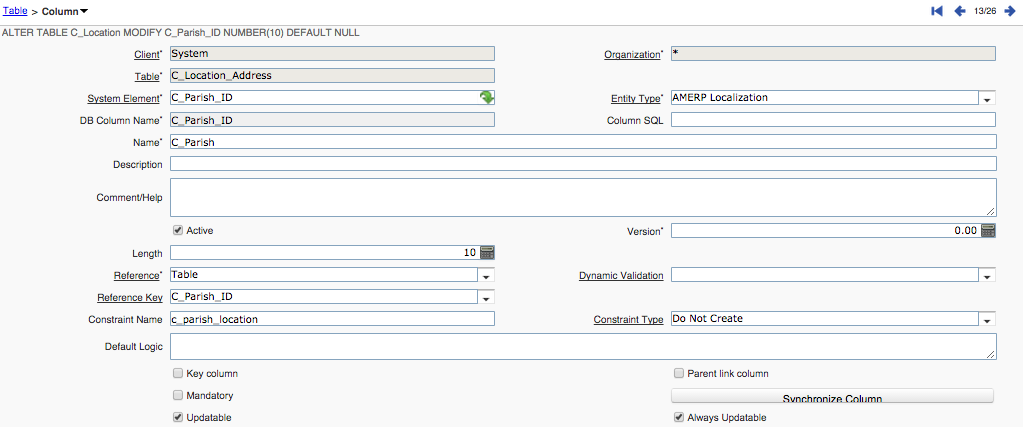


Figure 9.2

Both Fields may contain Null values in order to be compatible with Idempiere Address Standard.

Field *C\_City\_ID* may contain Null Value, because some Country can be configured with ‘*Allow Cities out of List’ feaure*. A City String Field is Provided for Storing values when this happens.

# 10.Country-Community-Region-Municipality-Parish Window.

Finally a New Window is created to perform extended address and Demographic Structure for the Countries. It has tried to develop the window so that it can be used as Standard Country-Region-City one. But also the Idempiere Administrator can perform the construction of new Communities, Municipalities and Parishes.

It is shown below the main Window, and will explain some fields that has to be set for Country in order to perform adequate update of actual Demography Structure.

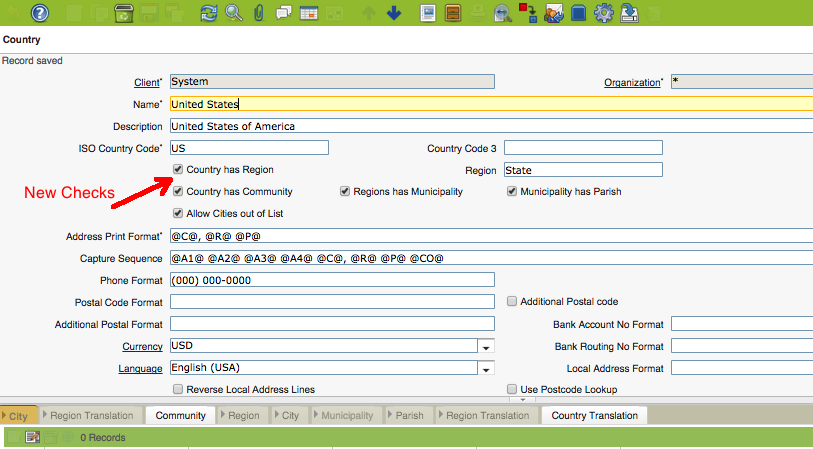


Figure 10.1

New Fields Added:

Country has Region

Country has Community

Regions has Municipality

Municipality has Parish

Idempiere Administrator may perform Community input through database, but just in case it is necessary to include the new Country’s Communities using the Window, the ‘*Country has Community’ check* has to be set in order to have Community Tab available and then include Communities.

After completing income communities, regions have to be associated with their respective communities. In order to do this job, Idempiere Administrator has to deselect ‘*Country has Community’* check in order to have Standard Region Tab available. Having this Region Tab Visible, Regions can be updated in order to set Community associated with them. See Figure 10.2 and 10.3.

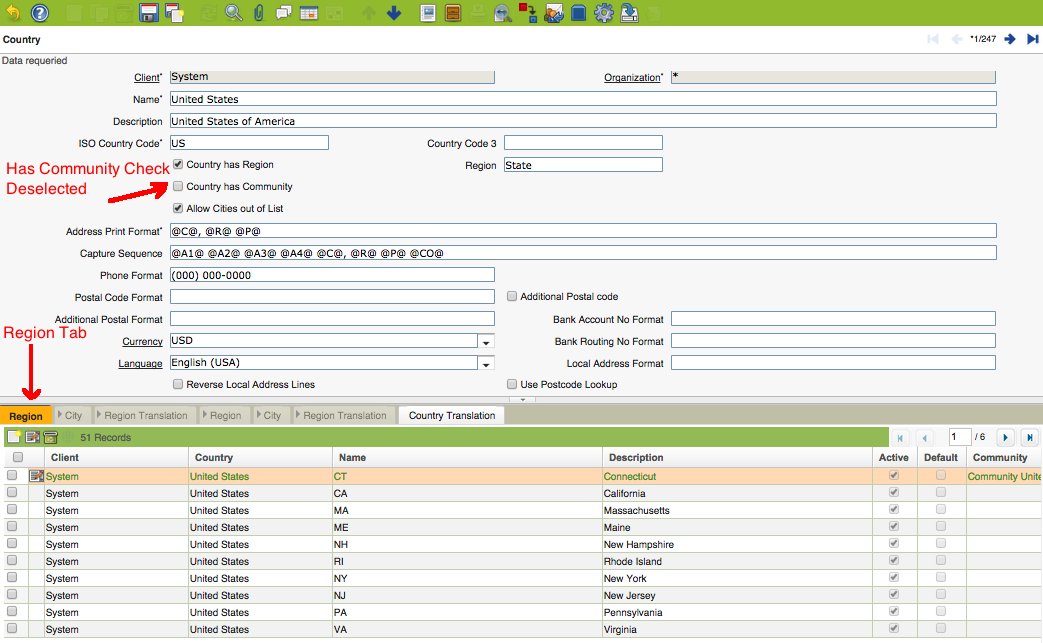


Figure 10.2

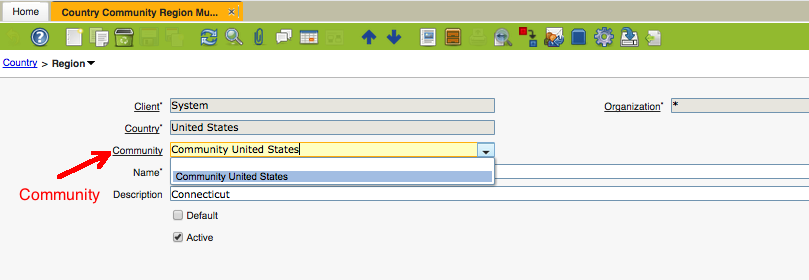


Figure 10.3

Complete Community association for all Regions and then set again ‘*Country has Community*’ Check. Community Tab will be visible again and Idempiere Administrator can perform Extended Demography Structure.

Community -> Region -> ( Municipality – City) 🡪 Parish

# 11.Business Partner Location Extended Example.

Once Plugin is installed as it is shown on A1 section, some AD modifications must be done before changes been observed. One example is Business Partner Address, on Business Partner Location Tab.

Get into Idempiere as System Administrator and locate Business Partner Location Table: ***C\_BParter\_Location***.

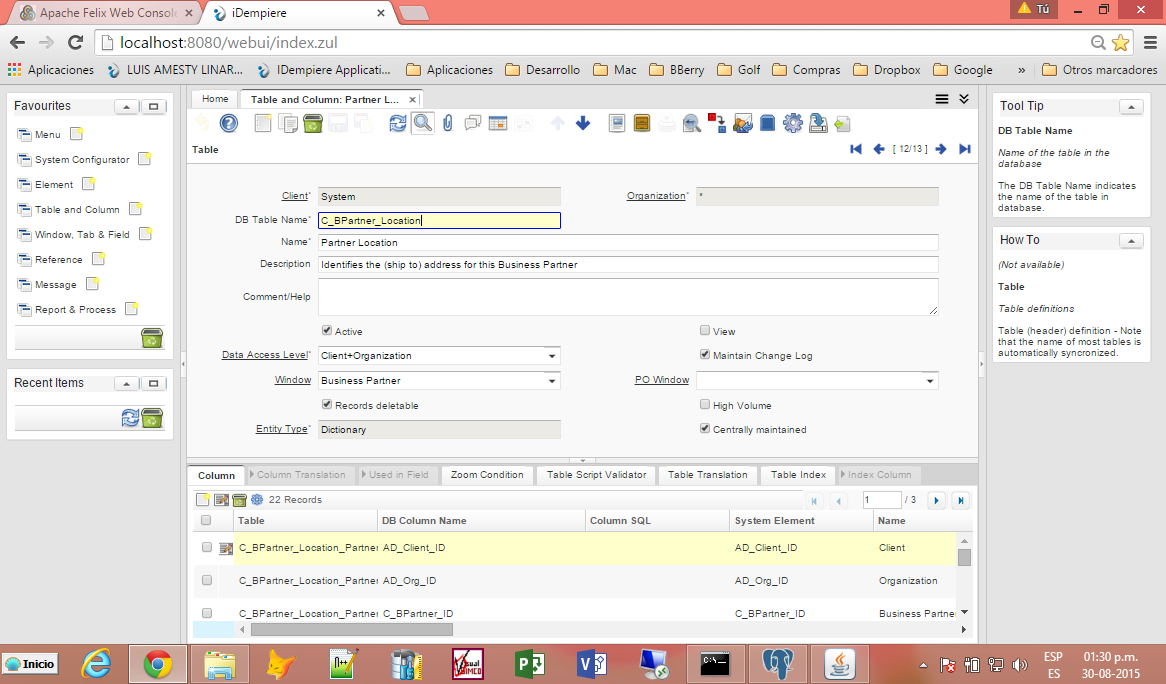


Figure 11.1

Modify ***C\_location\_ID*** Field, and set Reference as LocationExtended, as it was explained on Chapter 2.

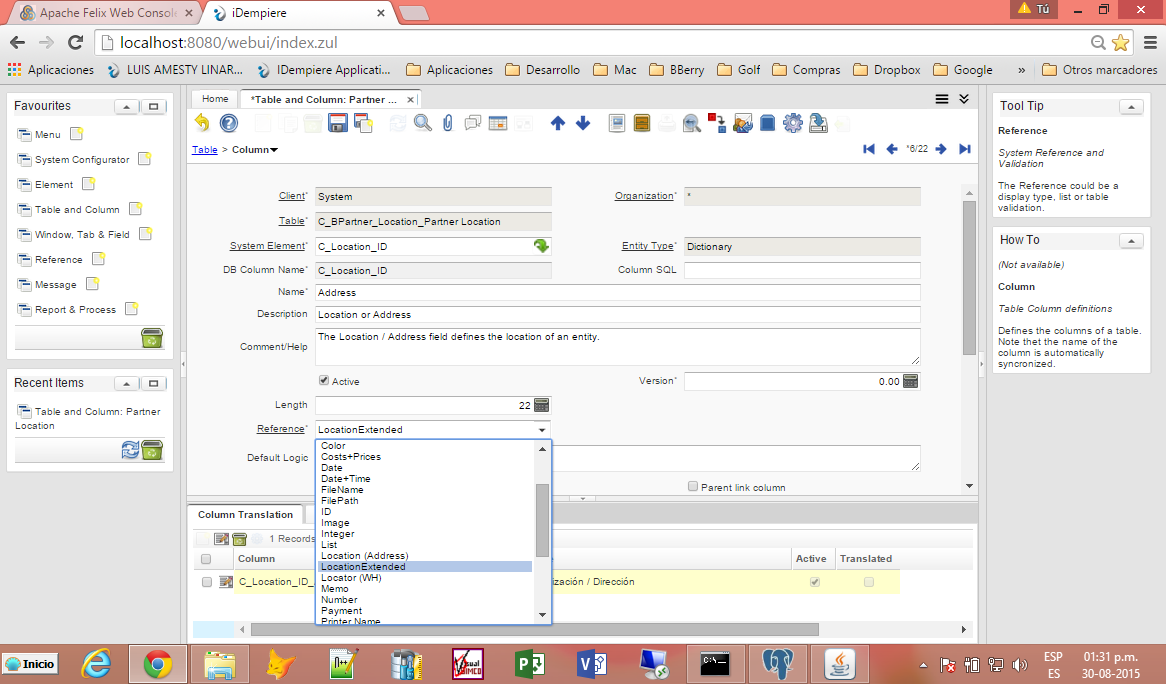


Figure 11.2

Now, you may test plugin. Get into Idempiere as SuperUser and Garden World; go to Business Partner Windows, and Location Tab. Modify Country to which was selected on Chapter 5, like Venezuela. For this example Capture Sequence was set as:

@CO@ @A1@ @A2@ @A3@ @A4@ @P@ @R@ @C@ @MU@ @PA@

Figure 11.3 shows changes observed on

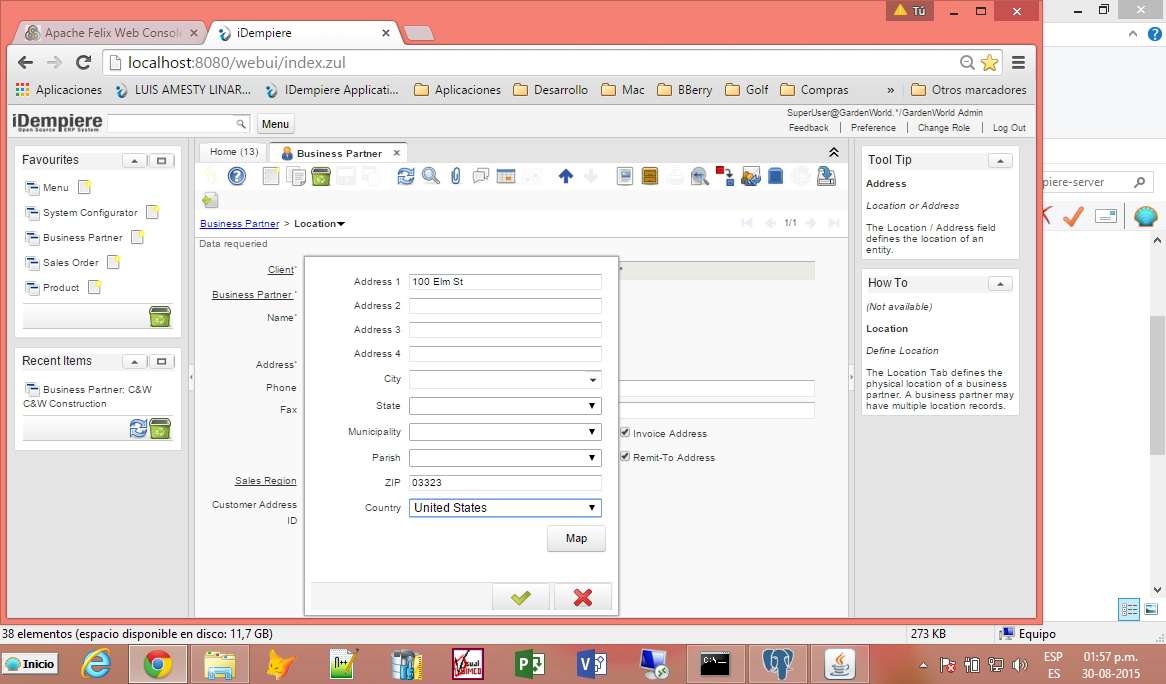


Figure 11.3

# A1. Plugin Installation.

You must follow indicated procedure as idempiere versión.

**For Version 6.2**

Follow these steps:

1. Pack-In AMERPSOFT Editor 6.2 – Step 1.zip
2. Pack-In AMERPSOFT Editor 6.2 – Step 2.zip
3. Set es\_VE as System language
4. Create Language extension for es**\_VE or xx\_**XX
5. Add country code 3 digits
6. Pack IN Venezuela Extended Demography
7. Install Plugin using Apache Felix Web Console
8. Restart Server
9. Modify **C\_BPartner\_Location** Table
10. Modify Country Record

**For Version 2.1 to 5.1:**

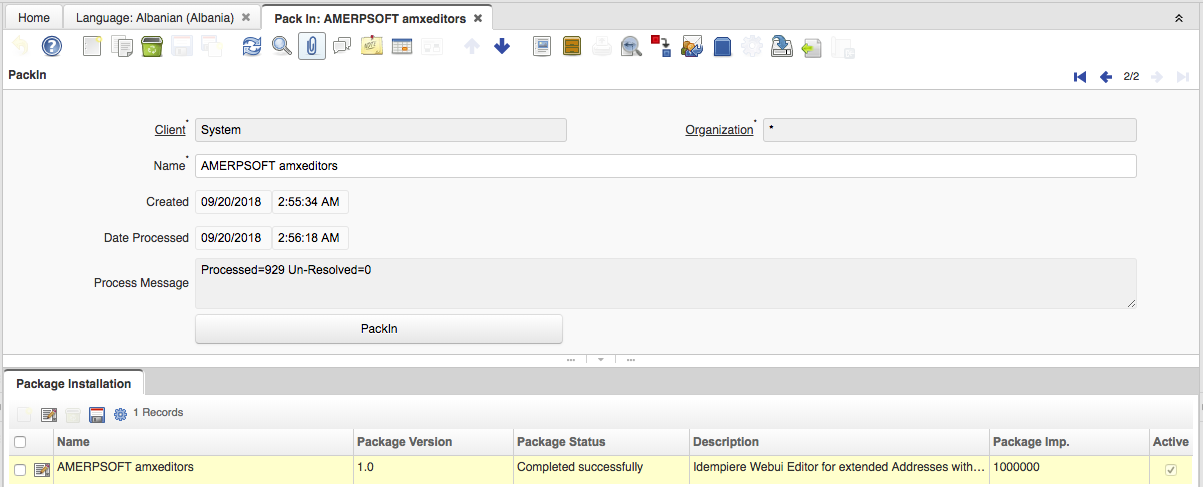
Follow these steps:

1. Pack-In AMERP amxeditor.zip
2. Set es\_VE as System language
3. Create Language extension for es**\_VE or xx\_**XX
4. Add country code 3 digits
5. Pack IN Venezuela Extended Demography
6. Install Plugin using Apache Felix Web Console
7. Restart Server
8. Modify **C\_BPartner\_Location** Table
9. Modify Country Record
10. **Pack IN AMERP amxeditor.zip**

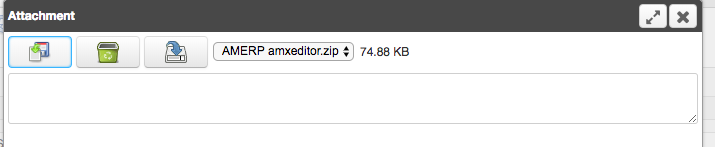
Download ‘AMERP amxeditor.zip’ from repository:

<https://bitbucket.org/amerpsoft/amerpsoft-editors-com/downloads/>

Create Pack In record and Attach: ‘AMERP amxeditors.zip’



Pack In ‘***AMERP amxeditor.zip’***



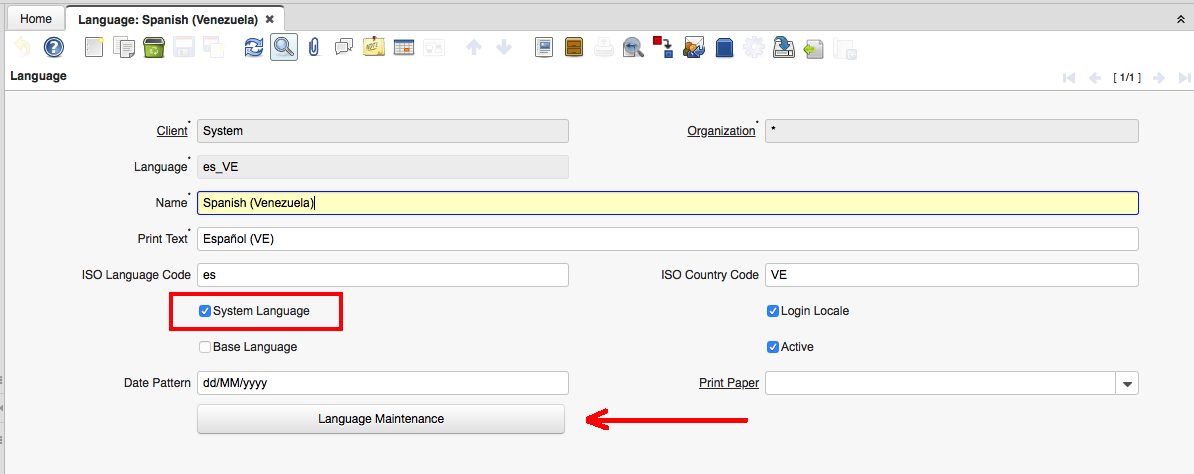
Attach ‘***AMERP amxeditor.zip’*** file

1. **Set es\_VE as System language**

Login idempiere as System in English

Go to Menu: /System Admin/General Rules/System Rules/Language

On Window Language: Locate es*\_VE or your preferred Language xx\_*XX



Language es*\_VE*

On Window Language: Check System Language

On Window Language: Execute 'Language Maintenance' Process

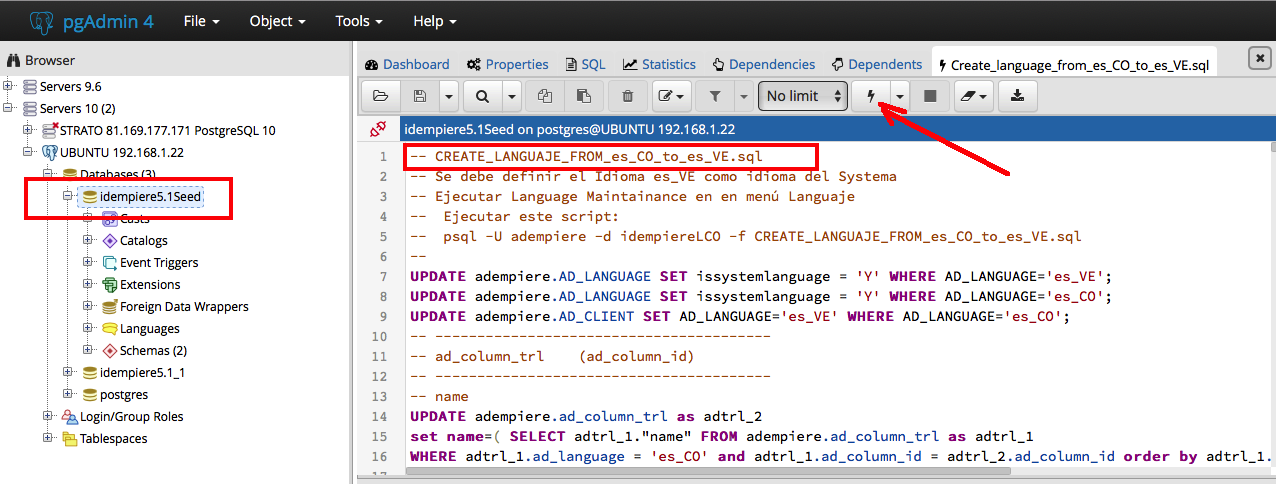
1. **Create Language extension for es\_VE or xx\_XX**

Download ‘***Create\_language\_from\_es\_CO\_to\_es\_VE.sql’*** Script from Repository

<https://bitbucket.org/amerpsoft/amerpsoft-editors-com/downloads/>

Execute Query from PostgreSQL environment

You may edit this Query for your favourite Language xx\_XX

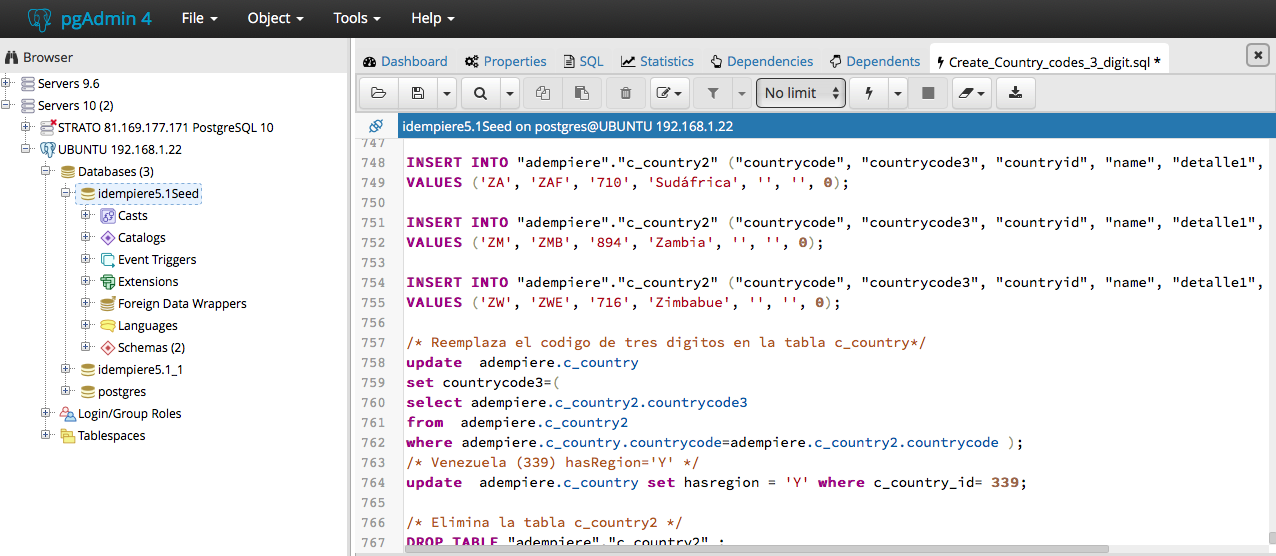


1. **Add country code 3 digits**

Download Script ***‘Create\_Country\_codes\_3\_digit.sql’*** from Repository

Execute Query from PostgreSQL environment

This Fill CountryCode3 Fields on ***C\_Country*** Table

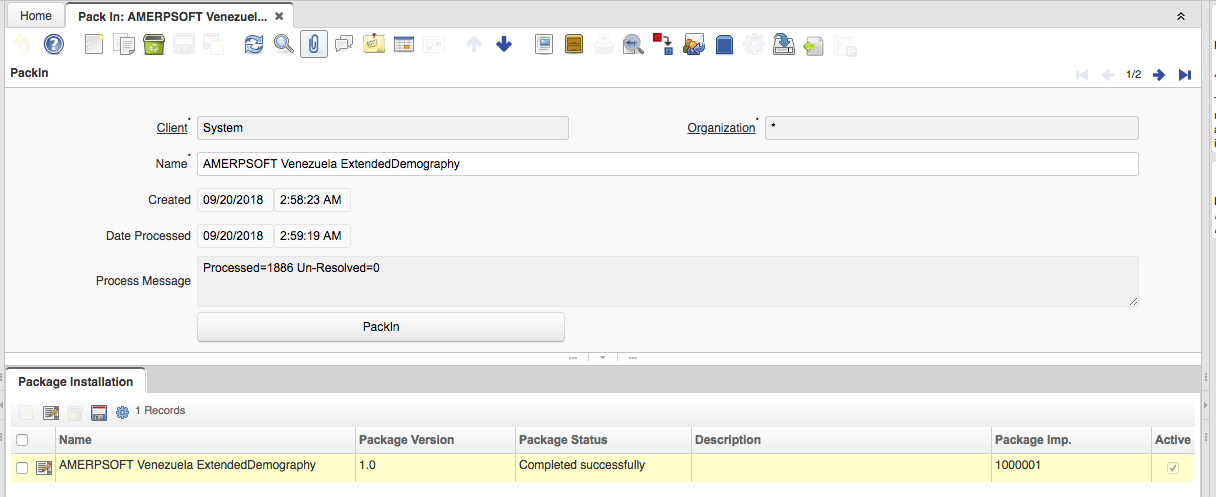


1. Pack IN Venezuela Extended Demography

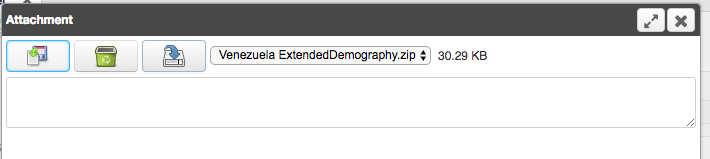
Download ***‘Venezuela ExtendedDemography.zip’*** from repository:

<https://bitbucket.org/amerpsoft/amerpsoft-editors-com/downloads/>

Create Pack In record and Attach: ‘***Venezuela ExtendedDemography.zip’***



Pack In ‘***Venezuela ExtendedDemography.zip’***



Attach ‘***Venezuela ExtendedDemography.zip’’*** file

You may look at **'GeografiaVenezolanaCompleta.sql'**

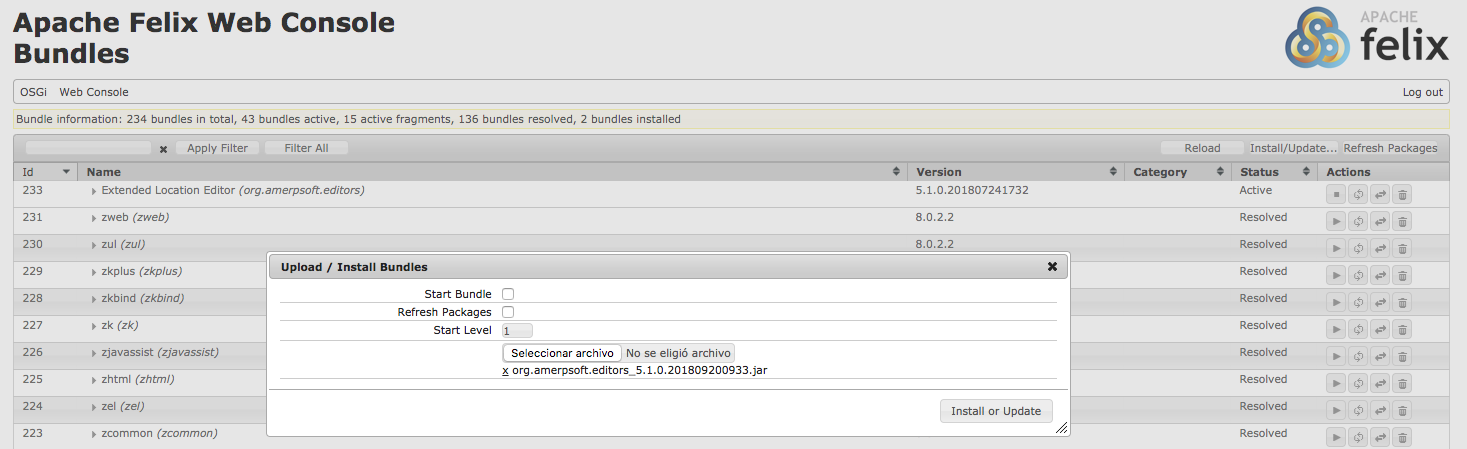
And make your Country Extended Demography Pack In.

1. **Install Plugin using Apache Felix Web Console**

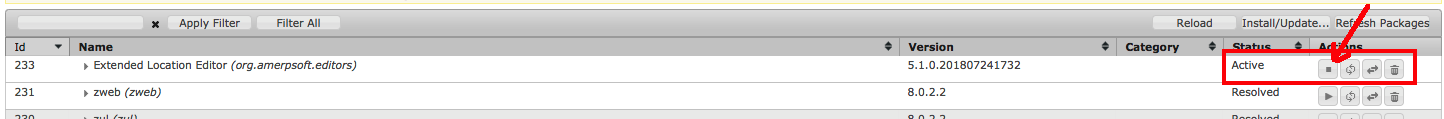
Download ***‘org.amerpsoft.editors\_5.1.0.201809200933.jar’*** from repository:

<https://bitbucket.org/amerpsoft/amerpsoft-editors-com/downloads/>

May be another version, check last one.



Verify plugin is updated and running.



1. **Restart Server**

You must restart Idempiere Server in order to accomplish the changes.

1. **Modify C\_BPartner\_Location Table**

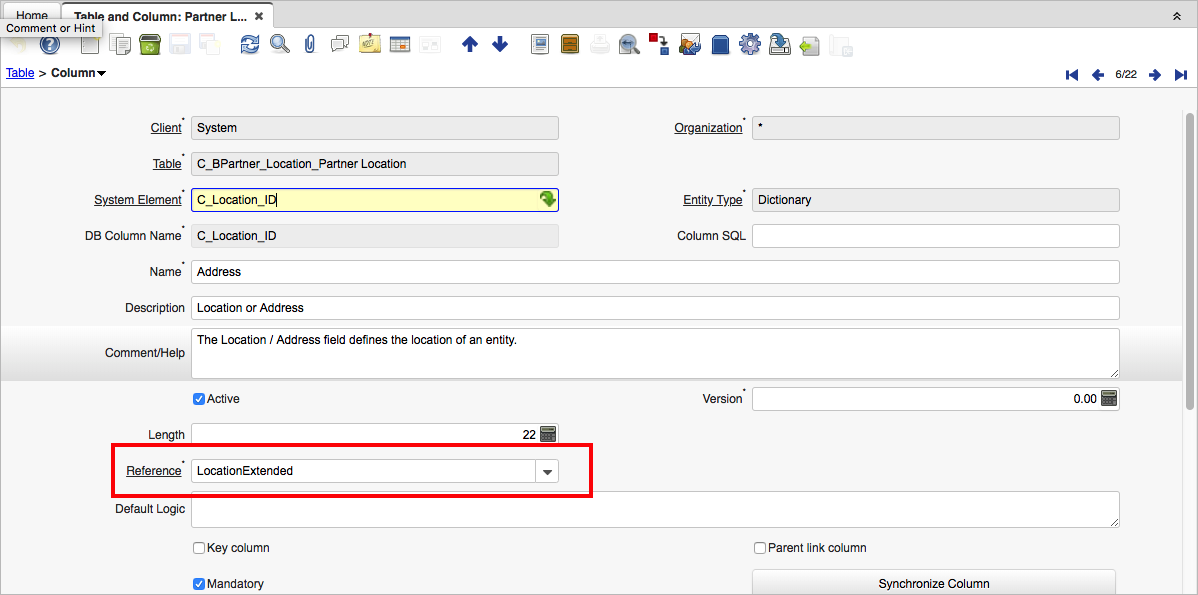
Login idempiere as System in English

Go to Menu: /Application Dictionary/ Table and Column

On Window Table and Column: Locate C\_BPartner\_Location Table

On Tab Column: Locate C\_Location\_ID Field

Change Reference to ‘***LocationExtended’***



1. **Modify Country Record**

Login idempiere as System in English

Go to Menu: /System Admin/General Rules/System Rules/Country Community …

This is the new window for Country Maintenance added from plugin.

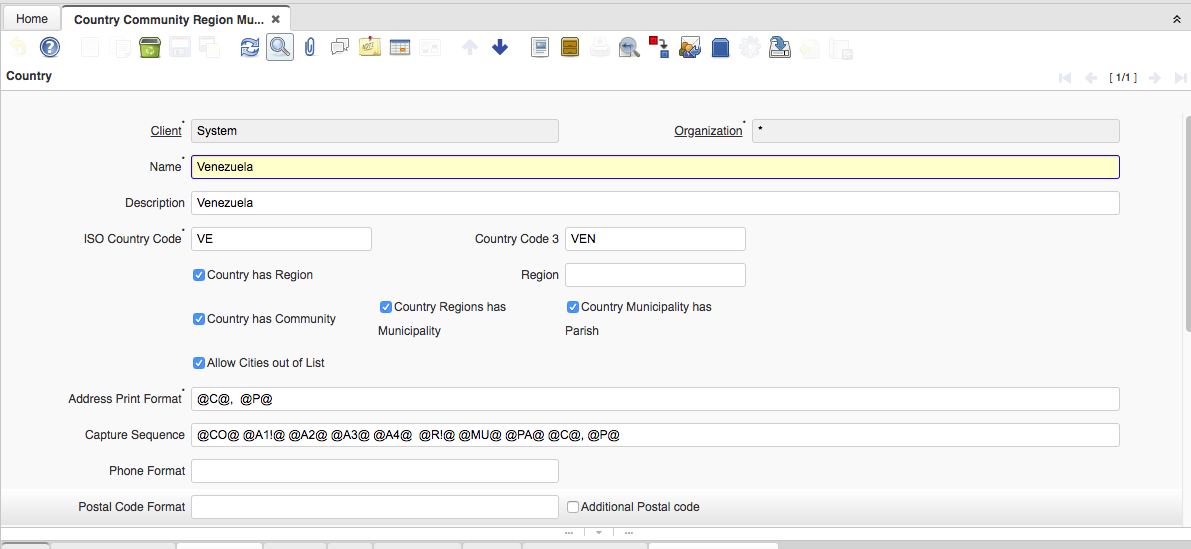
Check:

* Country has Community
* Country Regions has Municipality
* Country Municipality has Parish

Modify Capture Sequence as explained on Chapter 5:

**@CO@ @A1!@ @A2@ @A3@ @A4@ @R!@ @MU@ @PA@ @C@, @P@**

This sequence indicates plugin order on Field capture.



Capture Sequence