Our company is in the process of implementing Microsoft Dynamics CRM 2011. We have many “customer preference fields” that require multi-select functionality. My service provider’s solutions were not to the requirement most of the time, so I thought I would try to fix this up. I had this solution in mind and contacted Microsoft, for which they provided link to Jim’s blog. That is a great idea from Jim. But our requirement was little different, mainly because some of my lists of values are long. Thanks to Karen Carvalho (our Snr Marketing Manager), Christine Suell (my supervisor) and Andrew Bourke (Project Manager) for all the support and encouragement. Also thanks to Google and bloggers from whom I have found out syntax for many of the statements.

**The solution is simple though the code looks lengthy (I wanted error checks and customisation as far as possible). Anyone can implement this without modifying code.**

Advantages:

* Picklist can be of any length
* No need to modify the code
* Even non-technical persons can implement
* Values are searchable in your reports
* There is no special characters in the text field, so you can view the list without formatting
* There is no additional formatting or fields in the form except the two you create
* Does not occupy space on your form, requires only a single column
* There is only one function you need to implement at the onload event, so in case you lose this on an update you could easily bring it back.
* Fully free (of course LIKES and donations welcome!)

Disadvantages:

* Limitation on support (the method may be supported but the script is not supported)
* You need to create one additional field
* The values are stored in text field

**Solution:**

Those who need step-by-step explanation and screen shot, please scroll down.

Steps:

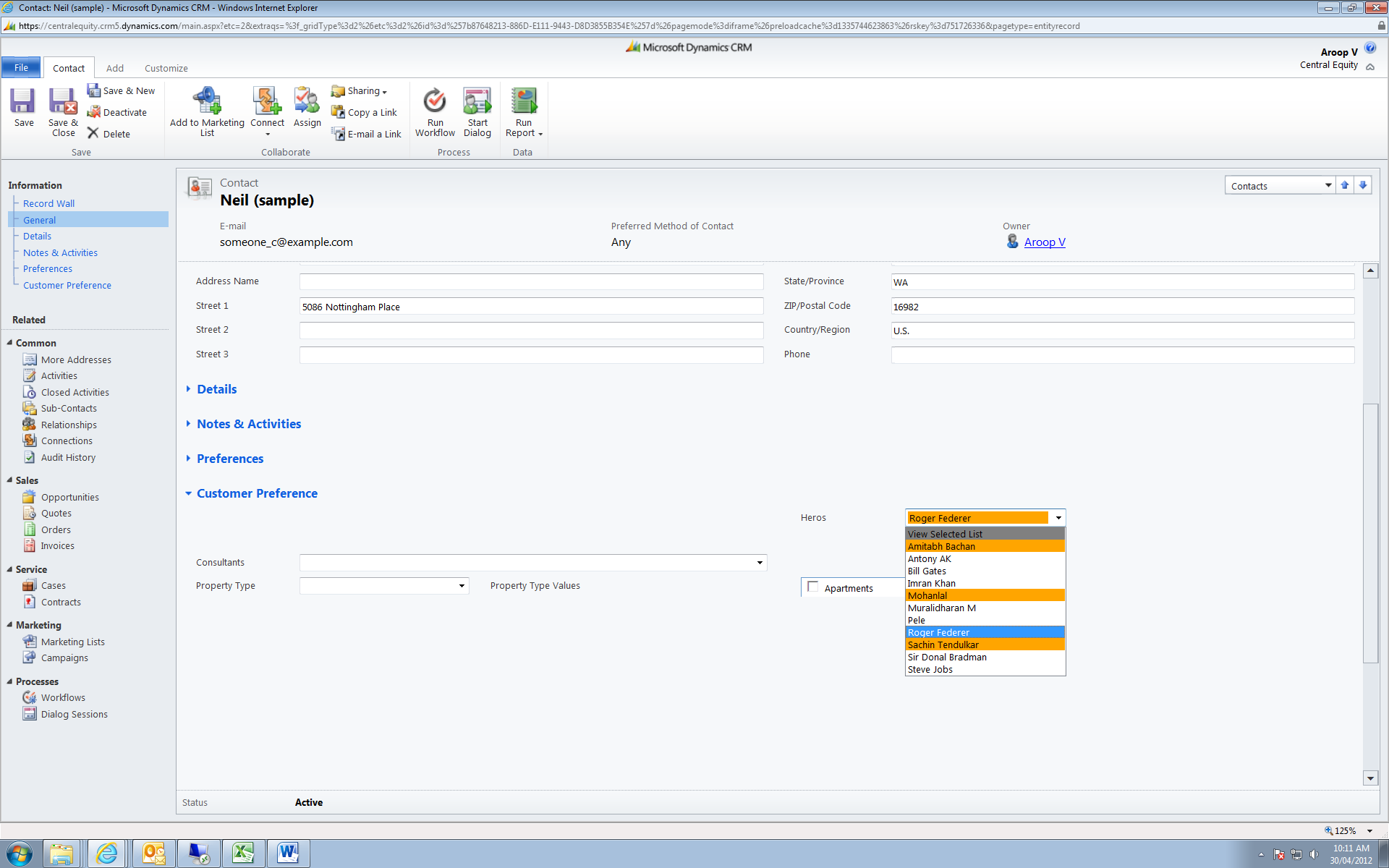
1. Create an option set and a multiline text field . You can choose your own names
2. Hide the text field (script takes care even if you forget, but suggested )
3. Add the script to Web Resource library.
4. Register a function on the onload event of the form and name the function as MultiPickList3
5. Pass the string parameters as show below:

Format: **<optionset fieldname> , <option value text field>** , [<option menu header> ], [<select color>] , [<deselect color> ]

Eg: “new\_herospl”,”new\_herostext”, ””, “green”, “white”

Only first 2 are compulsory. The 3rd parameter is a tricky one. It is assumed that the first option in your list of values (Picklist option field) is the heading, so suggest leaving it blank. However if you don’t have a header item you have to pass some junk string such as “~~no heading~~” or “xxxxxxx”.

Your solution is ready to use. I have tested this in Microsoft Dynamics CRM 2011 and should work in CRM 4 with a change of code in one or two places. However you have to test it for your installation and your comments are always welcome. I have not made use of some java functions because of compatibility issues.



**Code without explanation:**

**function MultiPickList3**(param1, param2, param3,param4,param5)

{

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\* Solution: Multi Select Pick List function:

//\*\*\*\*\*\* Platform: Microsoft Dynamics CRM 2011

//\*\*\*\*\*\* Last updated date: 25th Apirl 2012

//\*\*\*\*\*\* Created by: Aroop Vachali, aroopv@gmail.com, Company: Central Equity, Melbourne, Australia

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\* Courtsey: Karen Carvalho (Marketing Manager), Christine Suell (Supervisor), Google and all bloggers (for many

// of the syntaxes)

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

try

{

var fn = arguments.callee.toString().match(/function\s+([^\s\(]+)/);

if (param1==null || param2==null)

{

alert("Error: Parameter missing. \n Format: <optionset fieldsname> , <option value text field> , [<option header>], [<select color>] ,   
 [<deselect color>] , \n ["+"Function="+fn[1]+"]" );

return true;

}

var tparamtype=Xrm.Page.data.entity.attributes.get(param1).getAttributeType();

if (tparamtype!="optionset")

{ alert (param1+"(first parameter) should be of type optionset \n"+"[function="+fn[1]+"]");

return true;

}

var tparamtype=Xrm.Page.data.entity.attributes.get(param2).getAttributeType();

if (tparamtype!="memo")

{ alert (param2+"(second parameter) should be of type memo (text with mutiline) \n[function="+fn[1]+"]");

return true;

}

var plOptions=param1;

var plText=param2;

var plMenuItem="View Selected List";

var SelectedList\_orig = crmForm.all[plText];

var FullList=crmForm.all[plOptions];

var SelCtr=-1;

var new\_selColor="orange";

var new\_deSelColor="white";

if (param4!=null)

new\_selColor=param4;

if (param5!=null)

new\_deSelColor=param5;

var SelectedList =SelectedList\_orig.value.split("\r\n");

crmForm.all[plText].style.display="none";

if(FullList!=null && SelectedList!=null)

{

initColor();

if (param3!=null && param3!="")

{

plMenuItem=param3;

}

else

{

plMenuItem=FullList.options[0].text;

changeColor("grey",0);

}

for (var ctr=0; ctr<SelectedList.length;ctr++)

{

selCtr=SelectedIndex(SelectedList[ctr]);

if (selCtr >-1)

{

changeColor(new\_selColor,selCtr);

}

}

}

**function SelectedIndex**(selectedText)

{

var FullListText;

for (var ctr1=0; ctr1<FullList.options.length;ctr1++)

{

FullListText=FullList.options[ctr1].text;

if ( FullListText==selectedText)

{

return ctr1;

}

}

return -2;

}

crmForm.all[plOptions].attachEvent('onchange', OnChangePL);

**function OnChangePL**()

{

var SelCtr=-1;

var sel=crmForm.all[plOptions].SelectedText;

if (sel==plMenuItem)

return;

SelCtr=SelectedIndex(sel);

var SelColor="grey";

SelColor=crmForm.all[plOptions][SelCtr].style.backgroundColor;

if (SelColor==new\_selColor)

{

changeColor(new\_deSelColor,SelCtr);

saveChanges(sel,selCtr,"del");

}

else

{

changeColor(new\_selColor, SelCtr);

saveChanges(sel,selCtr,"add");

}

}

**function saveChanges**(p\_selText,p\_SelCtr,p\_mode)

{

switch(p\_mode)

{

case "add":

SelectedList.push(p\_selText);

break;

case "del":

for (var ctr2=0;ctr2<SelectedList.length;ctr2++)

{

if (SelectedList[ctr2]==p\_selText)

{

SelectedList.splice(ctr2,1);

break;

}

}

break;

}

Xrm.Page.getAttribute(plText).setValue(SelectedList.join("\r\n"));

}

**function initColor**()

{

for (var ctr3=0; ctr3<FullList.options.length;ctr3++)

{ changeColor(new\_deSelColor, ctr3); }

}

**function changeColor**(newColor, newCtr)

{

crmForm.all[plOptions][newCtr].style.backgroundColor=newColor;

}

}

catch (e)

{ alert (e.description);}

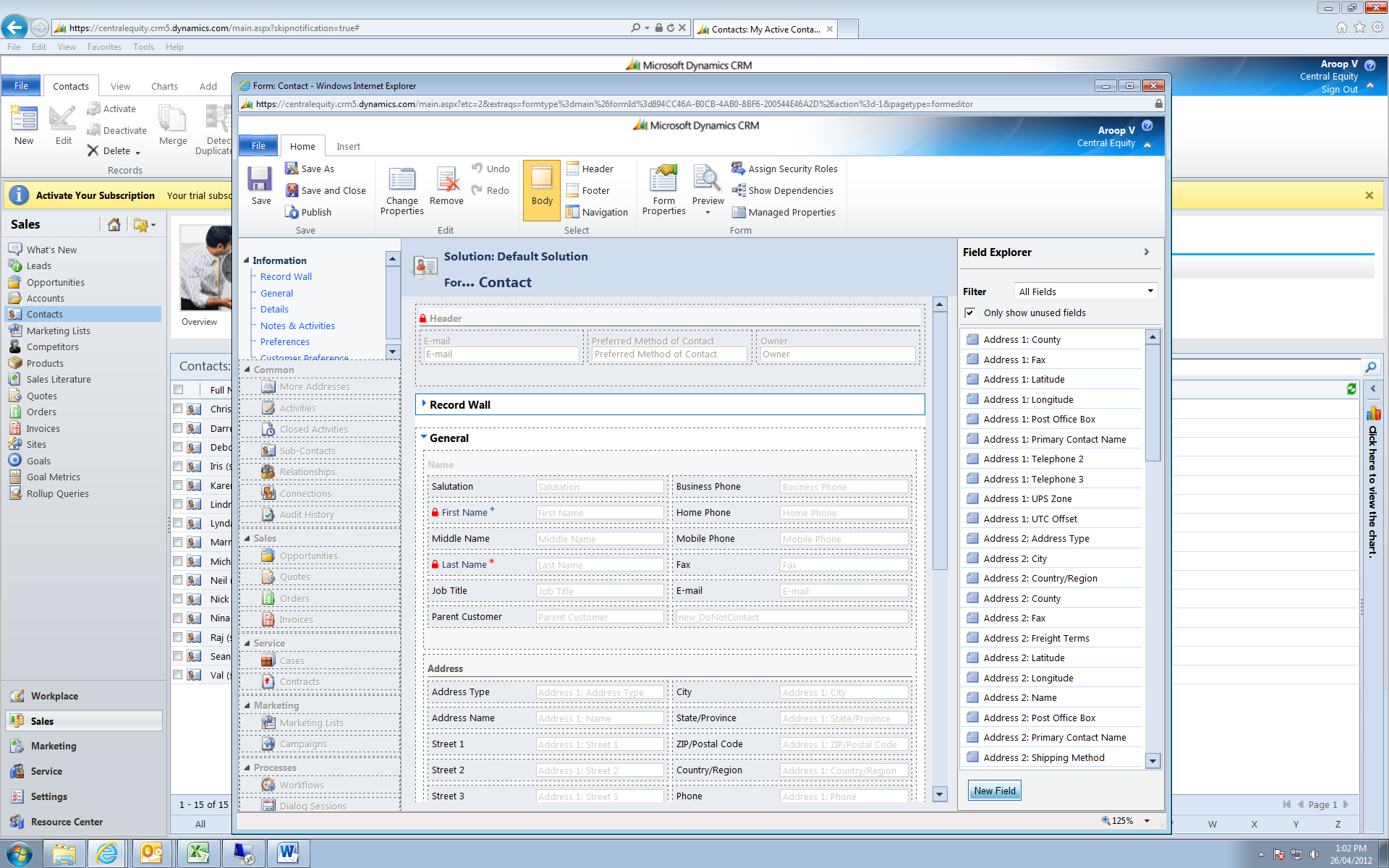
}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

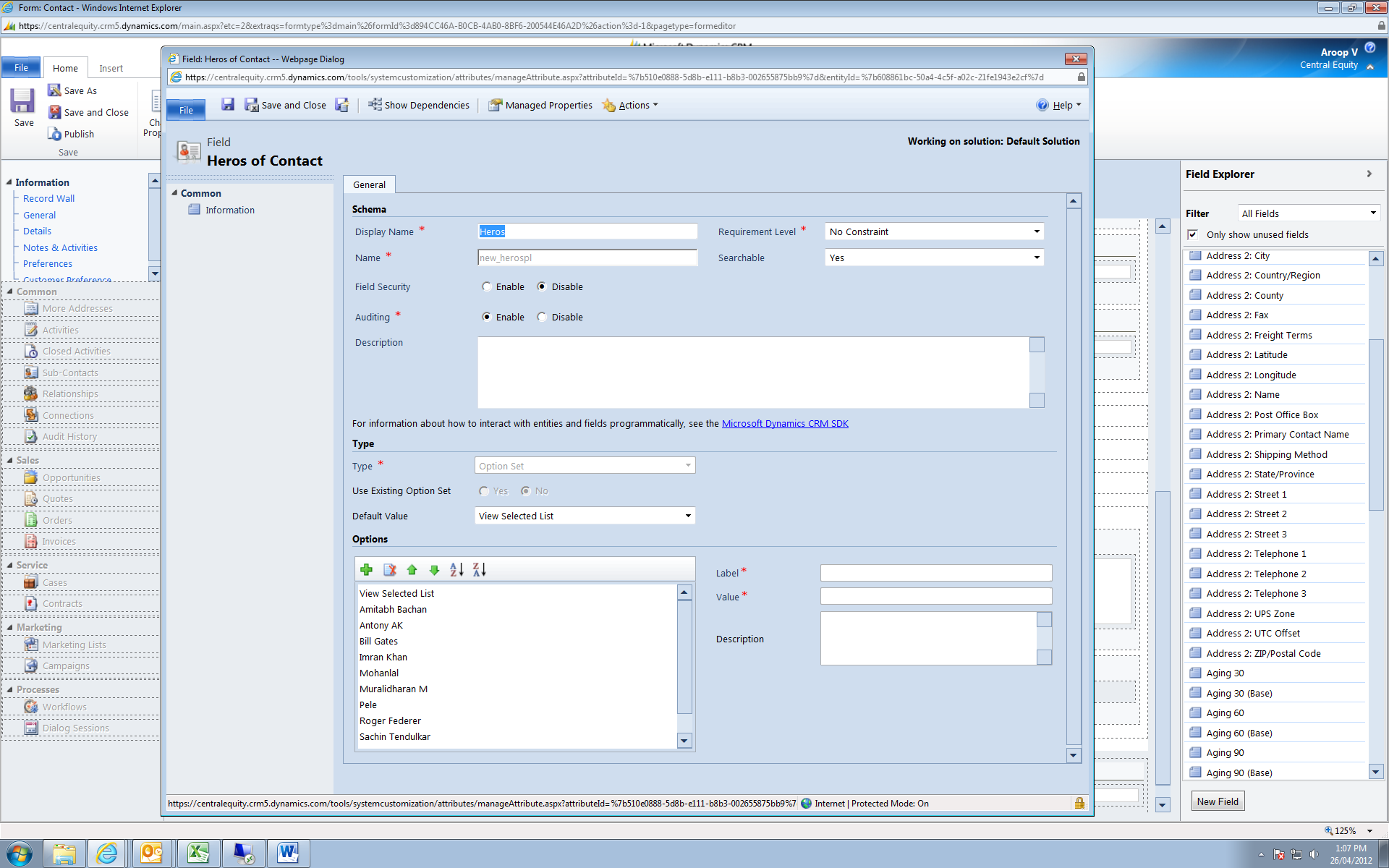
**Step-by-step explanation:**

1. Create an option set and a multiline text field . You can choose your own names

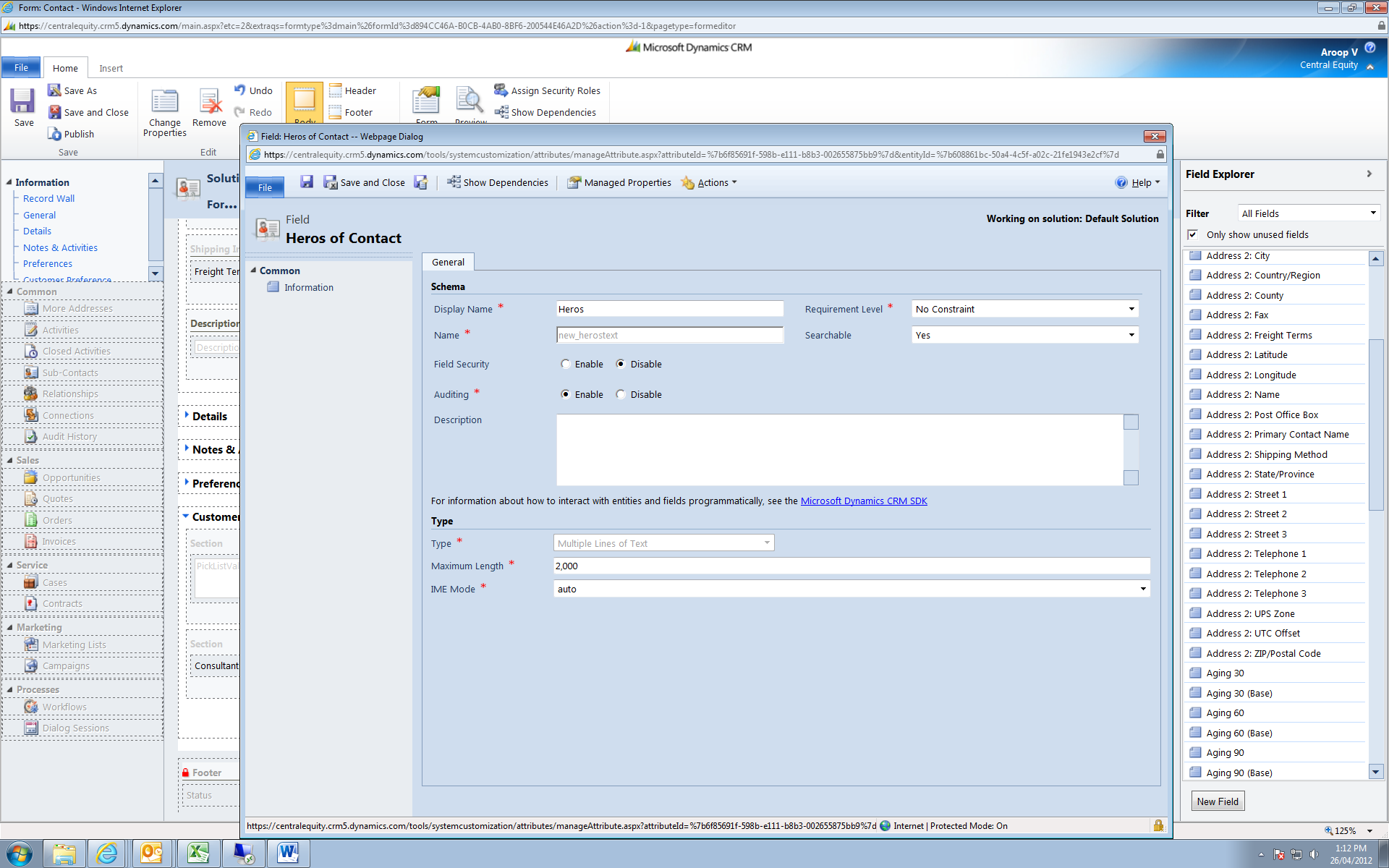
Navigation: Contacts->Customize Contacts->Forms->formType Main->New field



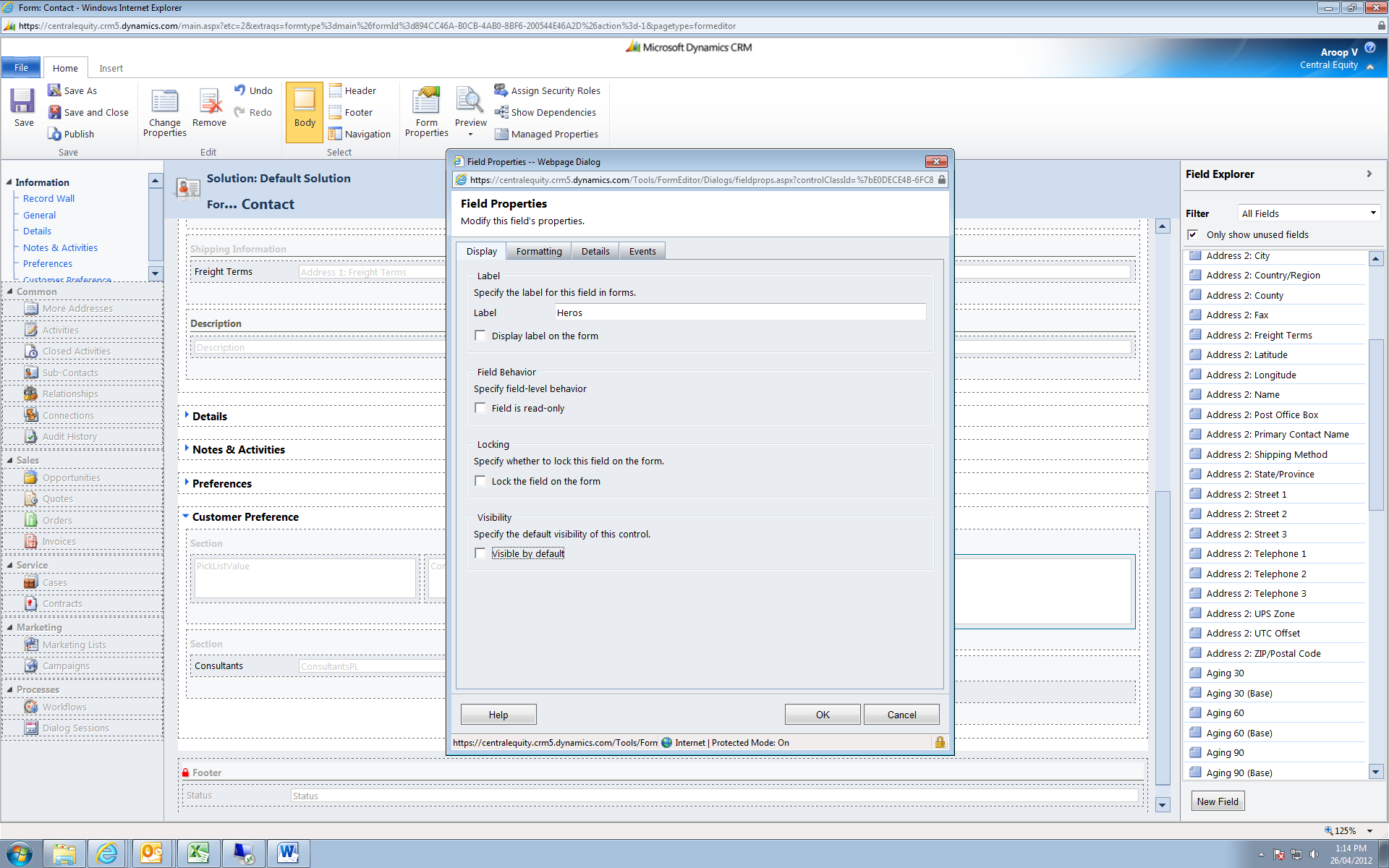
Add a field of type optionset, name the field, add header text as the first item and add list of values as shown below, save the changes



Repeat the same procedure to add another field, name it, select Mutiple lines of text as type, see below screen shot

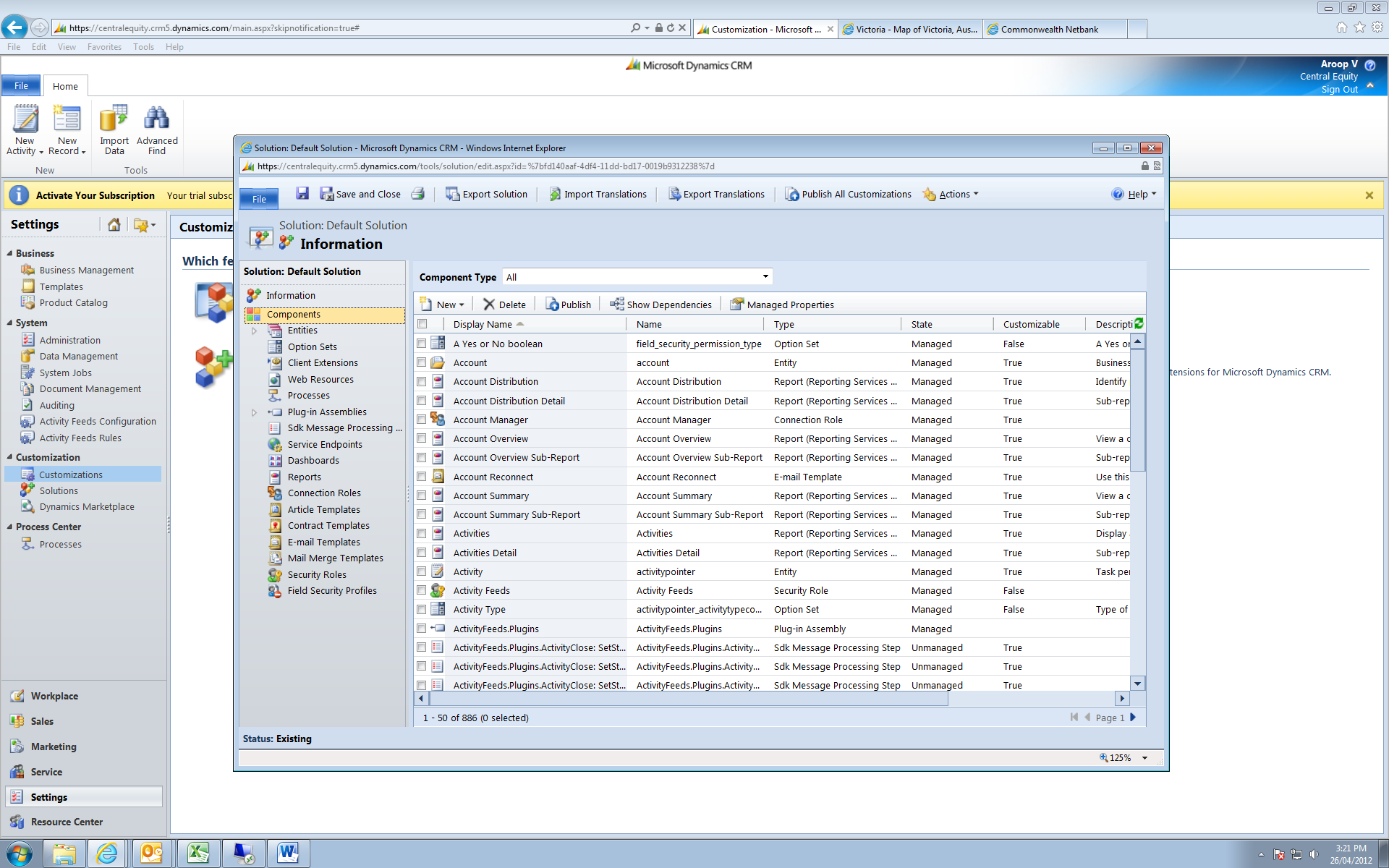


1. On the display tab of the field properties uncheck all of the options, this is to hide the field as well as label

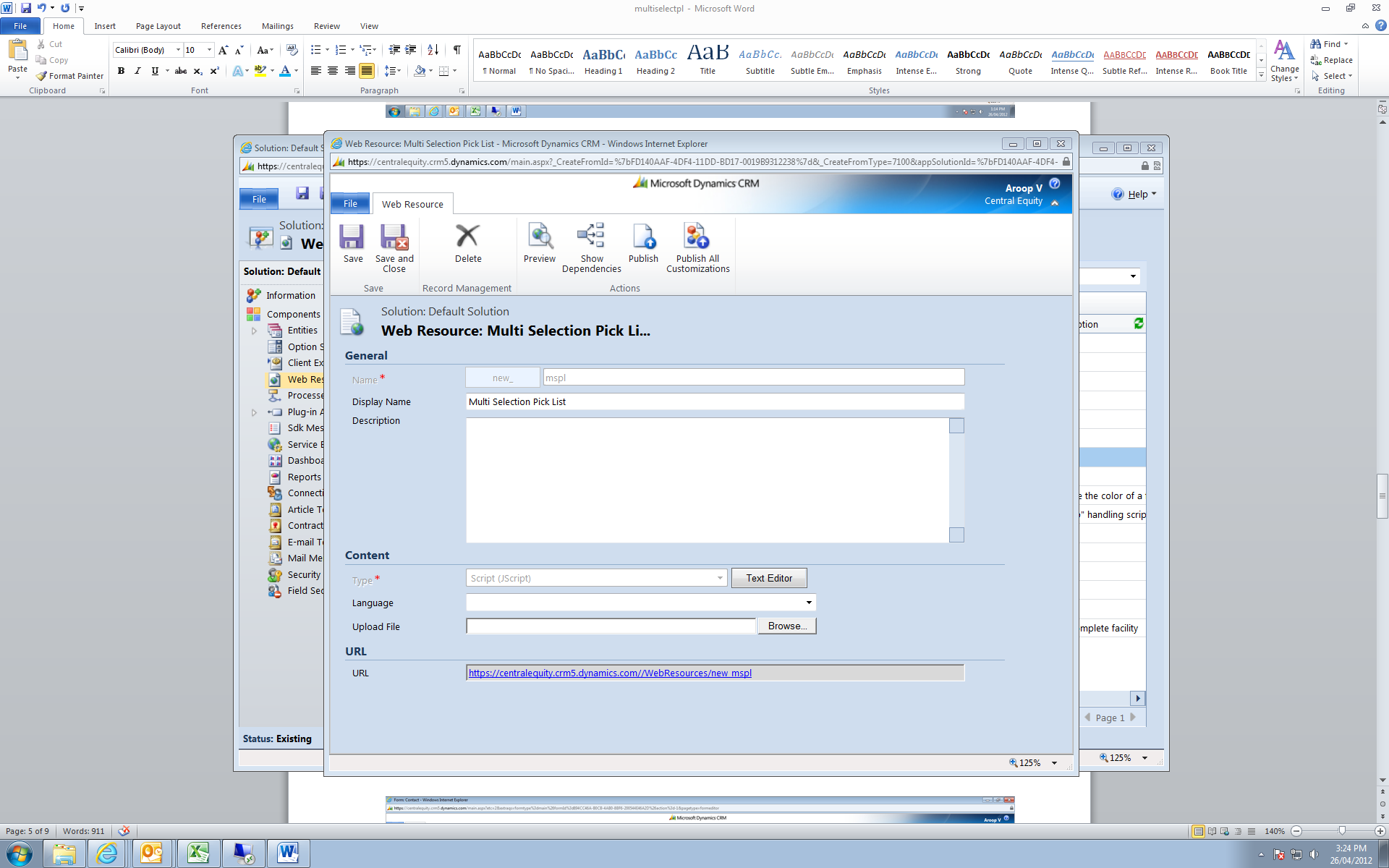


1. Add the script to Web Resource library.

Navigate to: Settings->Customization->Cystomize the system->Components->Web Resource->New



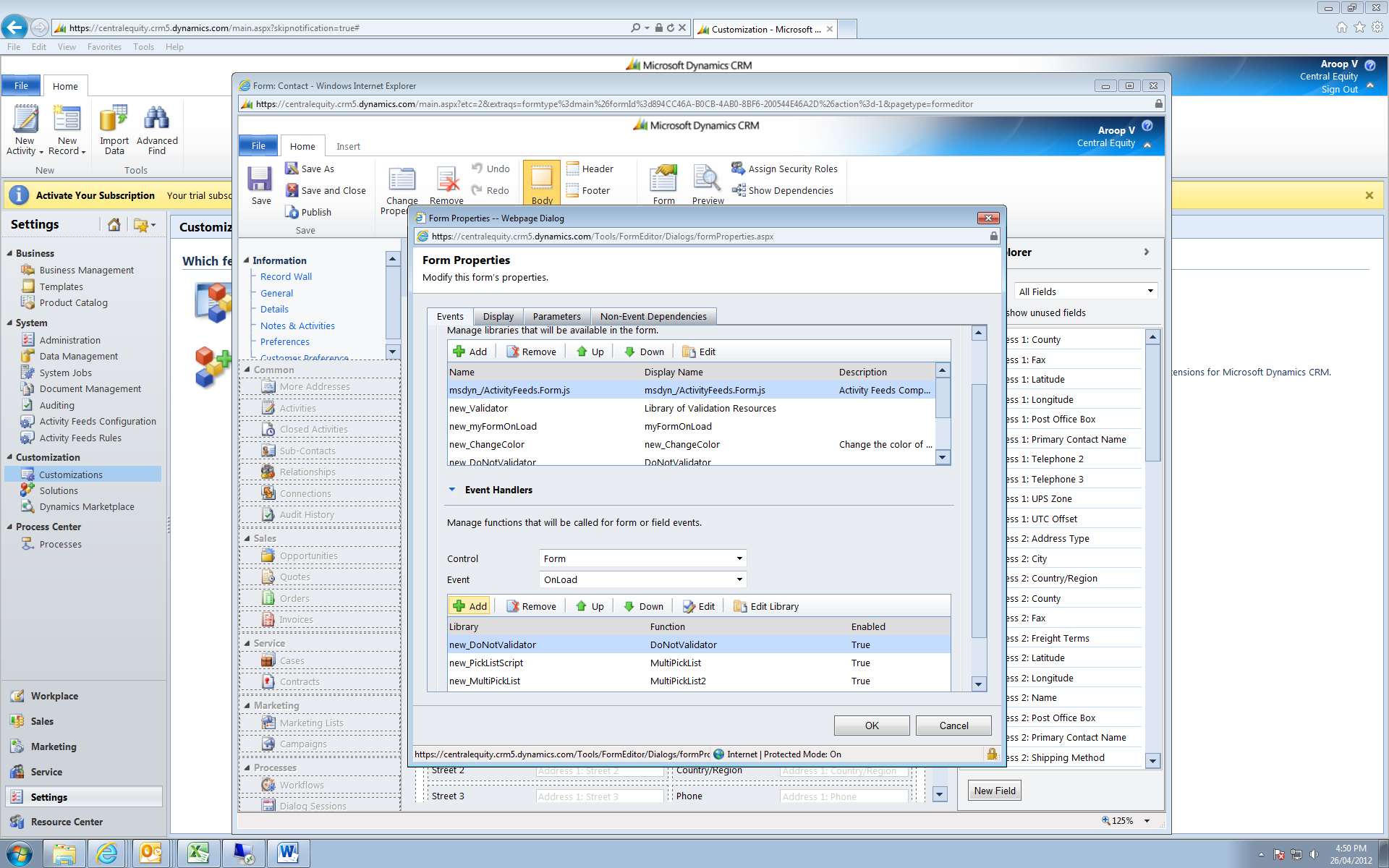
Type a name for the javascript file, type a name for the Display Name, under type select Script (Jscipt), click on the Text Editor, paste our script, save.



1. Register a function on the onload event of the form and name the function as MultiPickList3

Navigation: Contacts->Customize Contacts->Forms->formType= Main->Form Properties

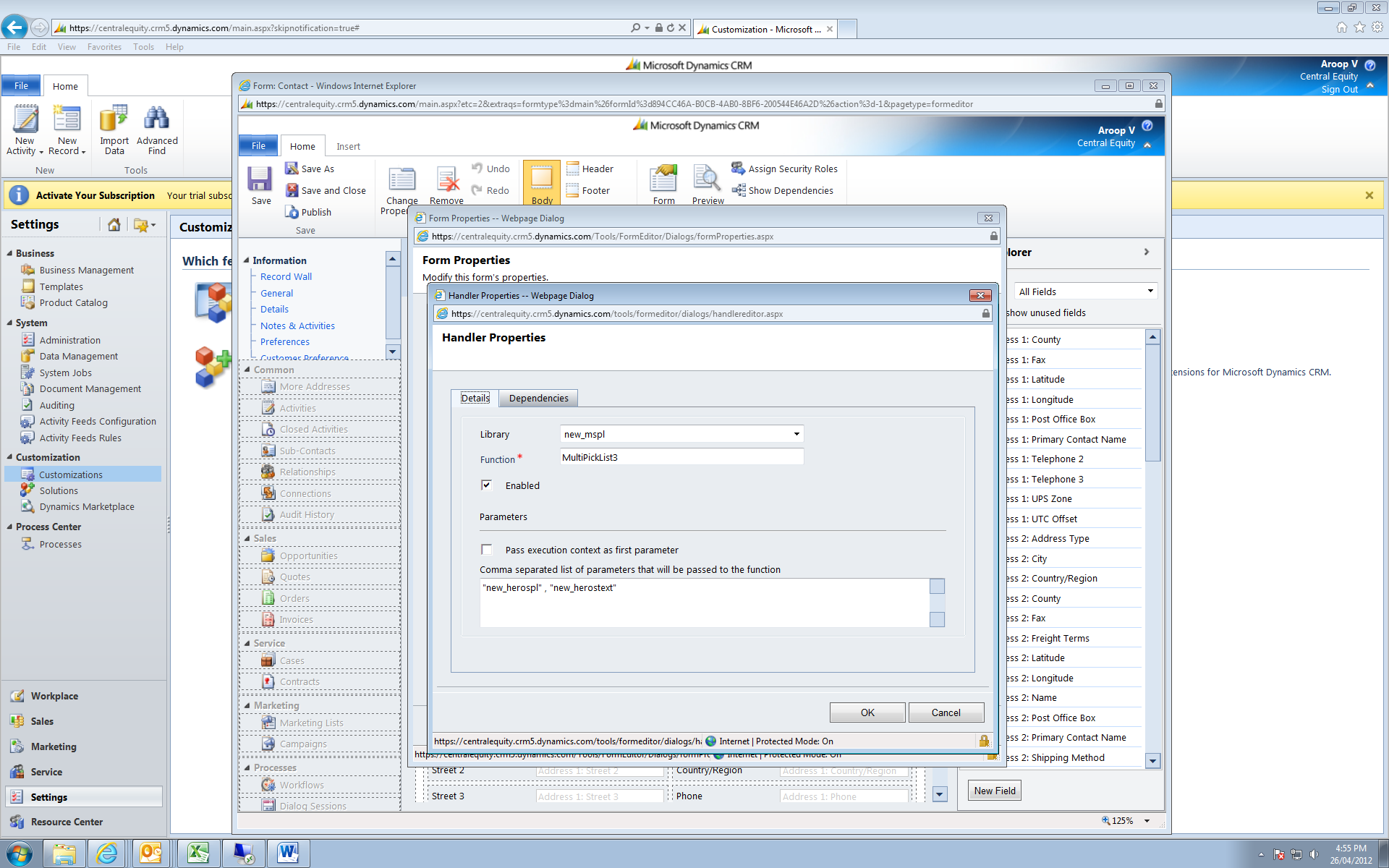
Under Event Handler select Form on Control, onLoad for Event->Add



Under Library select the javascript u registered earliar, under Function type MultiPickList3, under parameters type

your field names as given on top of the page

press OK save and Publish



**Code with explanation:**

**function MultiPickList3**(param1, param2, param3,param4,param5)

{

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\* Solution: Multi Select Pick List function:

//\*\*\*\*\*\* Platform: Microsoft Dynamics CRM 2011

//\*\*\*\*\*\* Last updated date: 25th Apirl 2012

//\*\*\*\*\*\* Created by: Aroop Vachali, aroopv@gmail.com, Company: Central Equity, Melbourne, Australia

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\* Courtsey: Karen Carvalho (Snr Marketing Manager), Christine Suell (Supervisor), Andrew Bourke, Google, Microsoft and all bloggers (for many

// of the syntaxes)

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

try

{

var fn = arguments.callee.toString().match(/function\s+([^\s\(]+)/);

// fn will hold the name of the function (here MultiPickList3), by this you can have your own function name and all error messages will send the //correct function name courtesy to the blogger, don’t remember exactly the site, sorry

if (param1==null || param2==null)

{

alert("Error: Parameter missing. \n Format: <optionset fieldsname> , <option value text field> , [<option header>], [<select color>] ,   
 [<deselect color>] , \n ["+"Function="+fn[1]+"]" );

return true;

}

var tparamtype=Xrm.Page.data.entity.attributes.get(param1).getAttributeType();

if (tparamtype!="optionset")

{ alert (param1+"(first parameter) should be of type optionset \n"+"[function="+fn[1]+"]");

return true;

}

var tparamtype=Xrm.Page.data.entity.attributes.get(param2).getAttributeType();

if (tparamtype!="memo")

{ alert (param2+"(second parameter) should be of type memo (text with mutiline) \n[function="+fn[1]+"]");

return true;

}

// the above statements will first make sure the parameter 1 and 2 are not empty, then it would check the type of fields

var plOptions=param1; //var to hold the first field ie the optionset

var plText=param2; // var to hold the first field ie the optionset

var plMenuItem="View Selected List"; // var to hold the header in the optionset , to make sure header is not selected

var SelectedList\_orig = crmForm.all[plText]; // holds the list in the text box so that that values in optionset can be marked selected on load

var FullList=crmForm.all[plOptions]; // holds the full options in array

var SelCtr=-1;

var new\_selColor="orange"; // default select color will be replaced by 4th parameter if u pass one

var new\_deSelColor="white"; // default deselect color will be replaced by 4th parameter if u pass one

if (param4!=null)

new\_selColor=param4;

if (param5!=null)

new\_deSelColor=param5;

var SelectedList =SelectedList\_orig.value.split("\r\n"); // converts the the text field values into an array, each line an element

crmForm.all[plText].style.display="none"; // hides the text field

if(FullList!=null && SelectedList!=null)

{

initColor(); // initially marks everything as de-selected, just to make sure that even external   formatting will not affect our functionality

if (param3!=null && param3!="") // if no header text for the optionlist plMenuItem will be passed with the junk text from param3

{

plMenuItem=param3;

}

else

{

plMenuItem=FullList.options[0].text; // else 1st item as header text and color it grey

changeColor("grey",0);

}

for (var ctr=0; ctr<SelectedList.length;ctr++) // onload form scan thru the text field value which we stored in SelectedList and mark them selected

{

selCtr=SelectedIndex(SelectedList[ctr]);

if (selCtr >-1)

{

changeColor(new\_selColor,selCtr); // calls the changeColor function to change the color of the selected item

}

}

}

**function SelectedIndex**(selectedText) // searches optionlist array and returns the index on the selected text passed. Can be replaced with

{ // JScript built in function array.indexOf()

var FullListText;

for (var ctr1=0; ctr1<FullList.options.length;ctr1++)

{

FullListText=FullList.options[ctr1].text;

if ( FullListText==selectedText)

{

return ctr1;

}

}

return -2; // any negative value would be ok

}

crmForm.all[plOptions].attachEvent('onchange', OnChangePL); // attach an onchange event function to the option field

**function OnChangePL**() // control comes here when an option is clicked on the option list

{

var SelCtr=-1;

var sel=crmForm.all[plOptions].SelectedText; //clicked item in the option list

if (sel==plMenuItem) // if the clicked item is List header nothing to be done

return;

SelCtr=SelectedIndex(sel);

var SelColor="grey"; // ignore

SelColor=crmForm.all[plOptions][SelCtr].style.backgroundColor; //gets the color of the selected item, could also achive thru a diff method

if (SelColor==new\_selColor) // toggle color, if it is a selected item deselect else select. This is the key to the whole logic

{

changeColor(new\_deSelColor,SelCtr);

saveChanges(sel,selCtr,"del");

}

else

{

changeColor(new\_selColor, SelCtr);

saveChanges(sel,selCtr,"add");

}

}

**function saveChanges**(p\_selText,p\_SelCtr,p\_mode) // saving the changes, saving can be done onsave event as well, didn’t want to fire another event

{

switch(p\_mode)

{

case "add":

SelectedList.push(p\_selText); //if it’s a new selection add that to the selectedlist array , a beauty from java

break;

case "del":

for (var ctr2=0;ctr2<SelectedList.length;ctr2++)

{

if (SelectedList[ctr2]==p\_selText)

{

SelectedList.splice(ctr2,1); //if the click is on a selected item then remove from the selectedlist array , another one from java

break;

}

}

break;

}

Xrm.Page.getAttribute(plText).setValue(SelectedList.join("\r\n")); // copy the selected array to textfield

}

**function initColor**()

{

for (var ctr3=0; ctr3<FullList.options.length;ctr3++)

{ changeColor(new\_deSelColor, ctr3); }

}

**function changeColor**(newColor, newCtr)

{

crmForm.all[plOptions][newCtr].style.backgroundColor=newColor;

}

}

catch (e)

{ alert (e.description);}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// If it is a useful post, please don’t forget to LIKE &|| DONATE

// Thanking you Aroop Vachali, aroopv@gmail.com

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*