FarCry 4 : Creating a plugin using Form Tools

In FarCry 4 plugins are more or less self-contained applications that you can during instalation choose to deploy, or not, for each FarCry 4 application.

I'm going to try to explain how to create such a plugin – also using one more feature that is new to FarCry 4 namely Form Tools.

I'm going to make a plugin that will let users create forms on their own.

Prepearing the directories

Before we continue we should prepare some directories we'll need. In your plugins directory create the following directory structure:

```
idlForm
  customadmin
    idlFormItem
  packages
    types
  webskin
    idlForm
    idlFormItem
```

Defining the objects needed

To start with we need to have two kind of objects – one object we'll call idlForm and the other we'll call idlFormItem. (In my work we preface all objects with idl – you will probably want to use something else).

The idlForm object will be the object for the forms themselves, while idlFormItem will be the objecttype for the form elements – mostly input areas.

To define these objects we will create two simple cfc's and put them in the idlform/packages/types directory:

idlForm.cfc

```
<cfcomponent displayname="Form" extends="farcry.core.packages.types.types"
output="false" buseintree="true">

      <!--- type properties --->
      <cfproperty name="title" type="string" hint="Form title" required="yes"
      default="" ftlabel="Form title" ftseq="1">

      <cfproperty name="formheader" type="longchar" hint="Description /
      introductory text for the form" required="yes" default="Please fill in
      the form below:" ftlabel="Top text" ftseq="2">

      <cfproperty name="sendt" type="longchar" hint="Text to display when form
      is submitted" required="yes" default="Thank you!" ftlabel="Thank You
      text" ftseq="3">

      <cfproperty name="submittext" type="string" hint="Text for the submit
      button" required="yes" default="Send" ftlabel="Text for the submit
      button" ftseq="4">

      <cfproperty name="receiver" type="string" hint="E-mail address of the
      receiver of the form" required="yes" default="" ftlabel="Receiver
      ftseq="5">

      <cfproperty name="displayMethod" type="string" hint="Display method to
      render this HTML object with." required="yes" default="display"
      ftLabel="Display Template" ftType="webskin" ftPrefix="displayPage"
      ftSeq="6">

      <cfproperty name="aFormItems" type="array" hint="Holds objects to be
      displayed at this particular node." required="no" default=""
      ftlabel="Form Items" ftjoin="idlFormItem" ftseq="7">

</cfcomponent>
```

idlFormItem.cfc

```
<cfcomponent displayname="Custom Form Item Object"
extends="farcry.core.packages.types.types" output="false">

      <!--- type properties --->
      <cfproperty name="title" type="string" hint="Form Item title"
      required="yes" default="">

      <cfproperty name="name" type="string" hint="Form Item name (no spaces)"
      required="yes" default="">

      <cfproperty name="type" type="string" hint="The type of Form Item"
      default="textfield" fttype="list" ftlist="textfield:Kort tekst,
      textarea:Lang tekst, checkbox:Avkryssningsboks, radiobutton:Radioknapp,
      list:Liste, filefield:Vedlegg, statictext:Statisk tekst">

      <cfproperty name="linebreak" type="string" hint="If there should be a
      linebreak after the item or not" required="yes" default="1">

</cfcomponent>
```

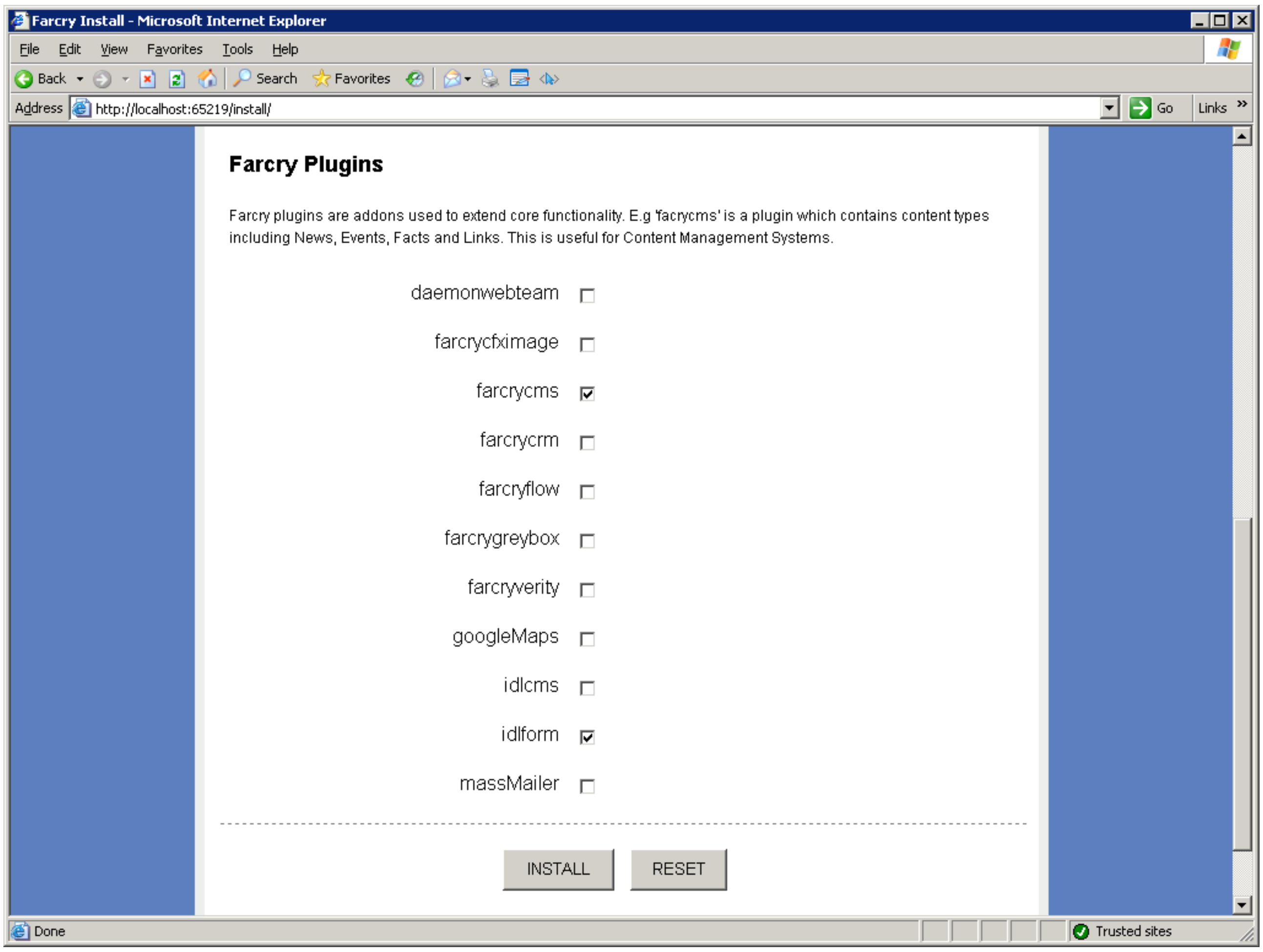
Ok – that's it!

Let's look at what we've done!

Just one small thing more before we take a look how this works out. In the idlform/webskin/idlform directory we make a displayPageStandard.cfm file with the following content:

```
<cfdump var="#stobj#">
```

Let's now install a new FarCry 4 application. We will see that idlform is on the list of available plugins. Now let's install the new application.



Farcry Plugins

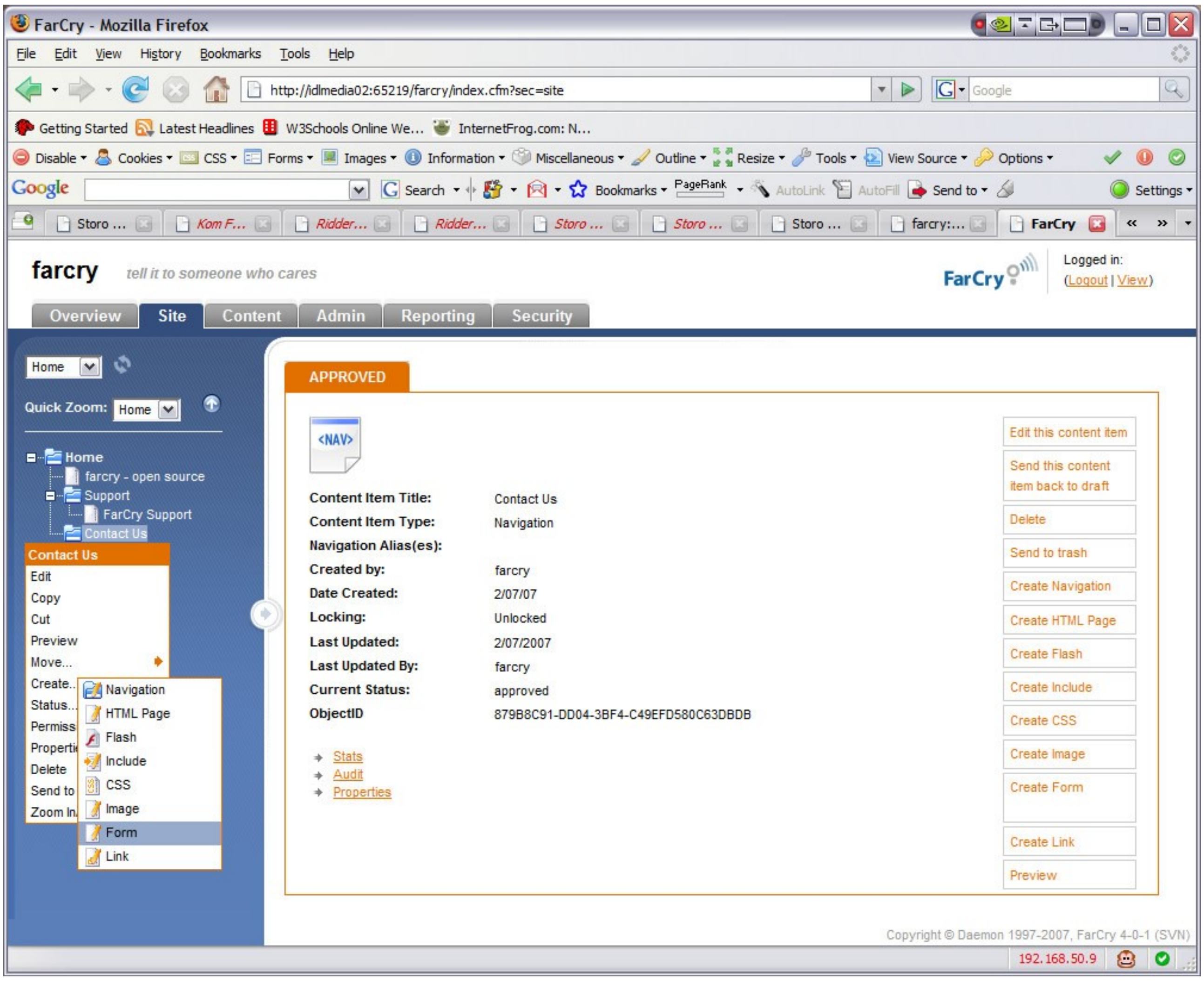
Farcry plugins are addons used to extend core functionality. E.g 'facrycms' is a plugin which contains content types including News, Events, Facts and Links. This is useful for Content Management Systems.

daemonwebteam	☐
farcrycfximage	☐
farcrycms	☑
farcrycrm	☐
farcryflow	☐
farcrygreybox	☐
farcryverity	☐
googleMaps	☐
idlcms	☐
idlform	☑
massMailer	☐

Farcry Install - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:65219/install/
Go Links
INSTALL
RESET
Done
Trusted sites

Once we've installed the new FarCry application let's create a new navigation node in the site (I'll name mine «Contact Us», and then right click on the new navigation node.

Yes – it's there. The option to create a new form.



Content Item Title:	Contact Us
Content Item Type:	Navigation
Navigation Alias(es):	
Created by:	farcry
Date Created:	2/07/07
Locking:	Unlocked
Last Updated:	2/07/2007
Last Updated By:	farcry
Current Status:	approved
ObjectID	879B8C91-DD04-3BF4-C49EFD580C63DBDB

Copyright © Daemon 1997-2007, FarCry 4-0-1 (SVN)

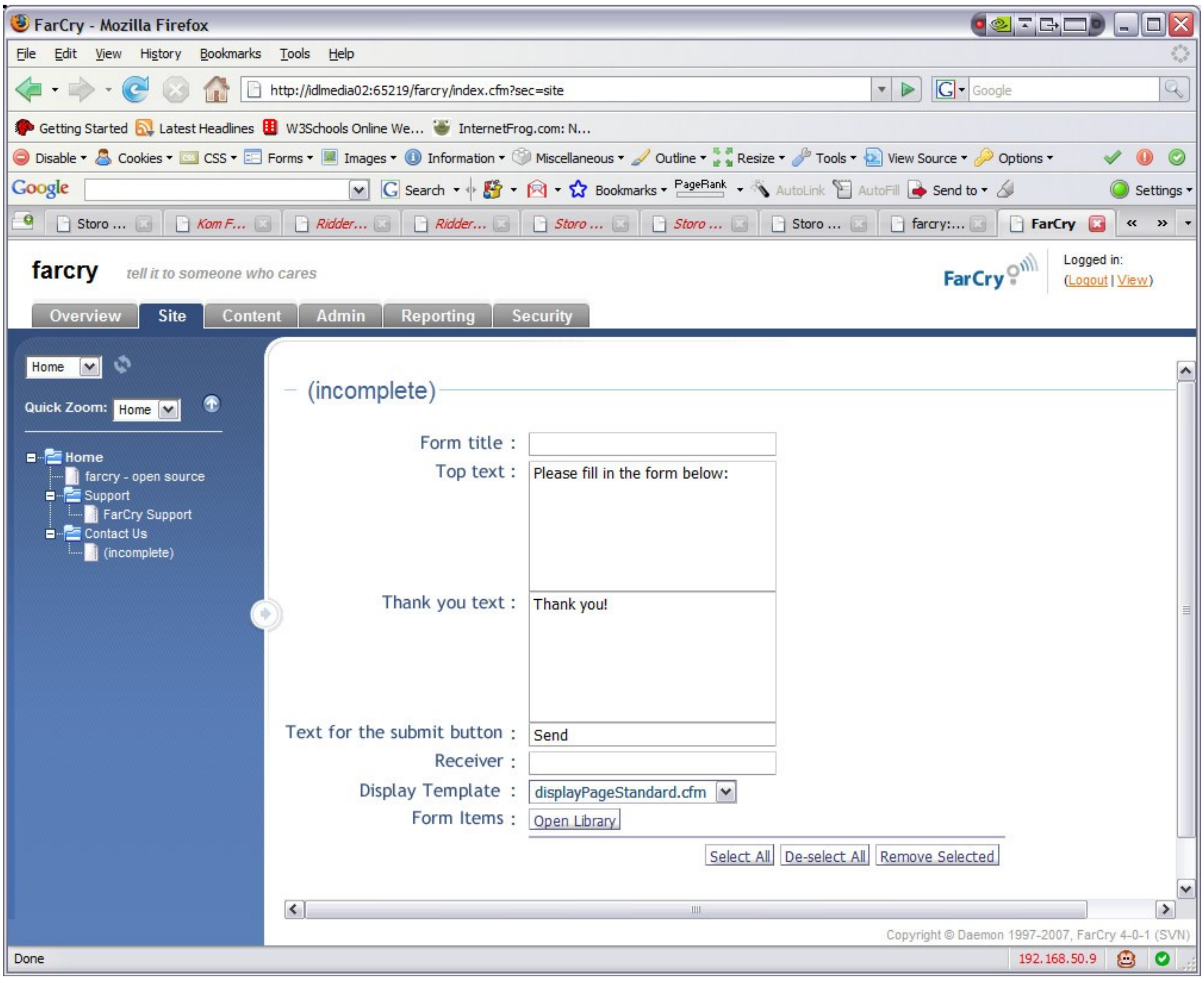
Is that smooth or what? Well it's nothing compared to what happens when you click that link. Then you'll see that you have everything you need for you (or the client you are making the site for) to create forms on the website. (I'm saying website instead of application here, as I boldly assume that the form creator will be used in the context of a FarCry CMS site).

FarCry has in fact created the database tables, the form for adding/editing new instances of the custom objects, and handles all interaction between the form and the database.

There's a lot of stuff about ORM, Scaffolding, Application- and code generation etc. in the blogsphere these days. Well ladies and gentlemen, here we have it all – at the speed of light I might say. Just from writing a couple of very simple cfc's. And we have barely scratched the surface of the posibilities in FarCry.

But I'm trailing of – let's get back to business.

Here's what you see when you click the Create Form link:



(incomplete)

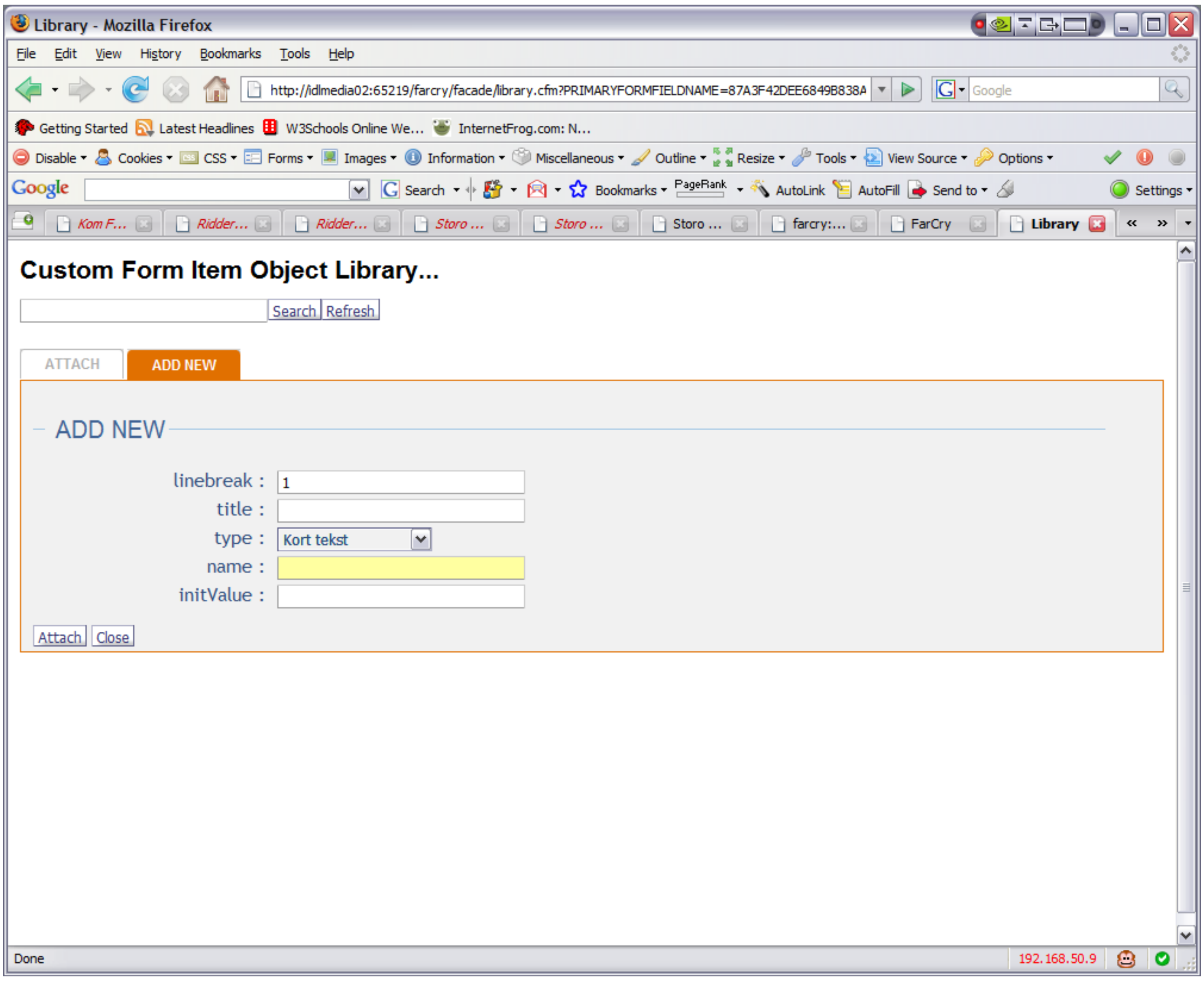
Form title :	
Top text :	Please fill in the form below:
Thank you text :	Thank you!
Text for the submit button :	Send
Receiver :	
Display Template :	displayPageStandard.cfm
Form Items :	Open Library

We see that the fields corespondence to the properties we made for the idlForm.cfc

The titles are ftlabel values for each cfproperty, and the order in which the form field apear is deceided by the ftseq numbered values. Also the kind of input field is deceided by the type value of the cfproperty.

Take a good look at the Display Template field. It found our displayPageStandard.cfm file. This is because we used the `ftType="webskin"` and `ftPrefix="displayPage"`. It's the ftType which does the magic of getting us the available templates, while ftPrefix tells what prefix the filenames should have, so if we put a file called viewTemplateForm.cfm it will not show up (unless we change it to `ftPrefix="viewTemplate"` in which case we would not get the displayPageStandard.cfm in the list).

Now let's fill in some values and then click the Open Library button for the Form Items.
This opens a new window where we can select form items to be used in our form. Since we do not have any form items from before we need to «Add New».


Library - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://idlmedia02:65219/farcry/facade/library.cfm?PRIMARYFORMFIELDNAME=87A3F42DEE6849B838A
Library
Done
192.168.50.9

Custom Form Item Object Library...

Search Refresh

ATTACH ADD NEW

ADD NEW

linebreak :	1
title :	
type :	Kort tekst
name :	
initValue :	

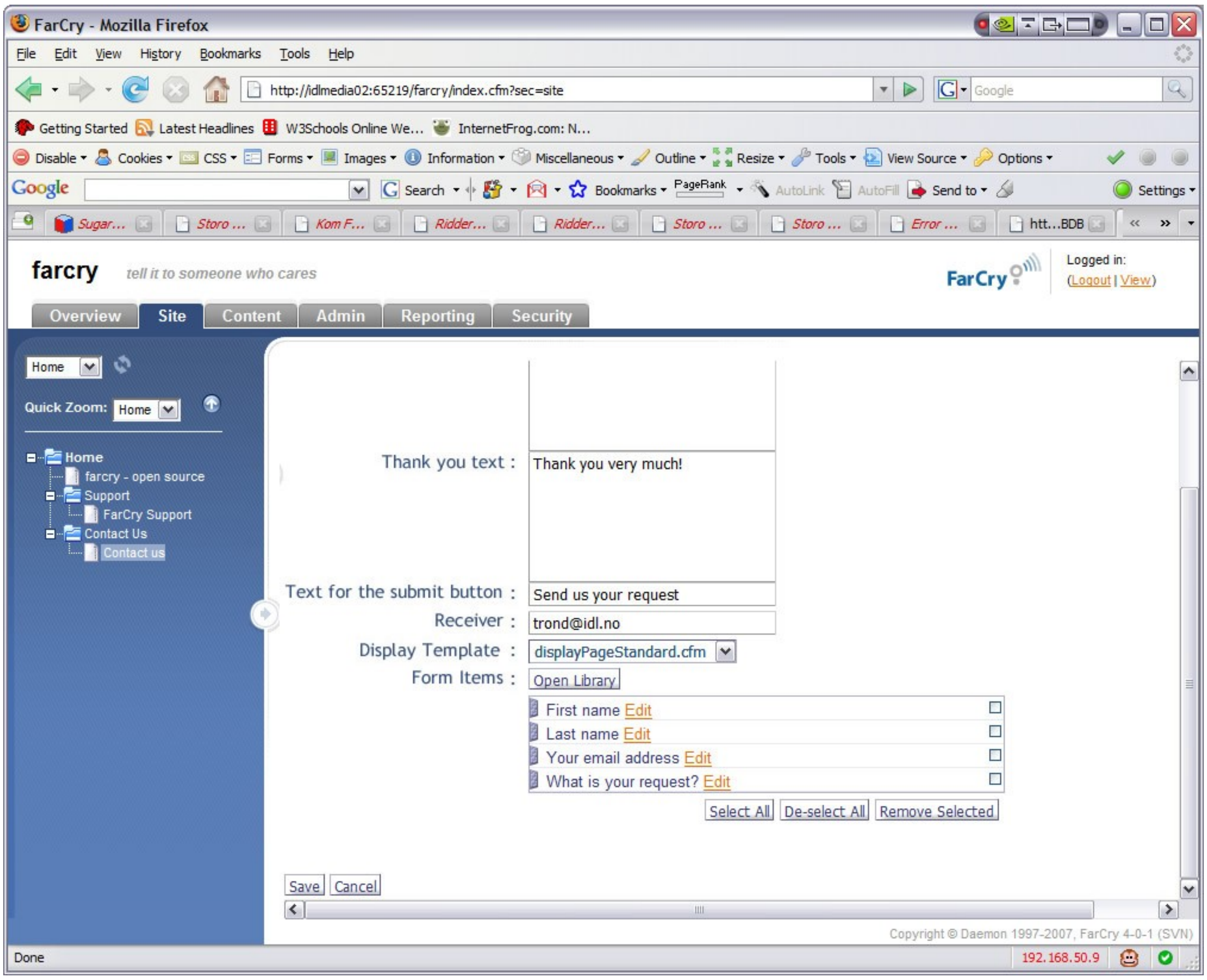
Attach Close

Todo: remove name – make english type choises

For the title I type in «First name», I leave type as it is. For initValue I write in «Write your first name here». Then I click «Attach».

While I'm at it I'll also create fields for«Last name», «Your email address» and «What is your request?».

Now – closing the «Attach form item» window and return to the main window, we see this:


FarCry - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://idlmedia02:65219/farcry/index.cfm?sec=site
farcry
tell it to someone who cares
Logged in: (Logout | View)
Overview Site Content Admin Reporting Security
Home
Quick Zoom: Home
Home
farcry - open source
Support
FarCry Support
Contact Us
Contact us
Copyright © Daemon 1997-2007, FarCry 4-0-1 (SVN)
Done
192.168.50.9

Thank you text : Thank you very much!

Text for the submit button : Send us your request

Receiver : trond@idl.no

Display Template : displayPageStandard.cfm

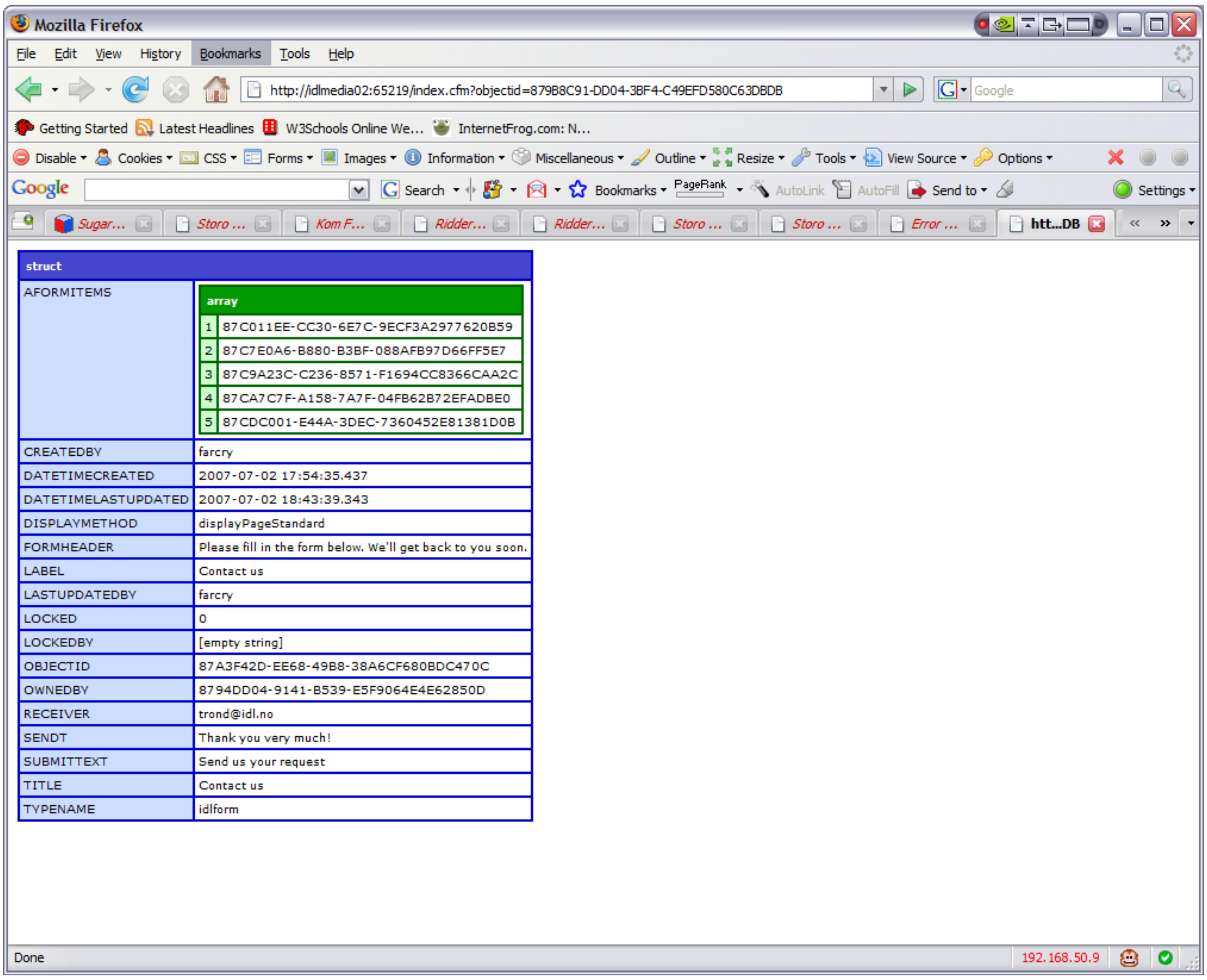
Form Items : Open Library

First name Edit ☐
Last name Edit ☐
Your email address Edit ☐
What is your request? Edit ☐

Select All | De-select All | Remove Selected

Save | Cancel

Ok let's save this form and go see what it looks like on the site:



```
struct
AFORMITEMS              array
                        1  87C011EE-CC30-6E7C-9ECF3A2977620B59
                        2  87C7E0A6-B880-B3BF-088AFB97D66FF5E7
                        3  87C9A23C-C236-8571-F1694CC8366CAA2C
                        4  87CA7C7F-A158-7A7F-04FB62B72EFADBE0
                        5  87CDC001-E44A-3DEC-7360452E81381D0B
CREATEDBY               farcry
DATETIMECREATED         2007-07-02 17:54:35.437
DATETIMELASTUPDATED     2007-07-02 18:43:39.343
DISPLAYMETHOD           displayPageStandard
FORMHEADER              Please fill in the form below. We'll get back to you soon.
LABEL                   Contact us
LASTUPDATEDBY           farcry
LOCKED                  0
LOCKEDBY                [empty string]
OBJECTID                87A3F42D-EE68-49B8-38A6CF680BDC470C
OWNEDBY                 8794DD04-9141-B539-E5F9064E4E62850D
RECEIVER                trond@idl.no
SENDT                   Thank you very much!
SUBMITTEXT              Send us your request
TITLE                   Contact us
TYPENAME                idlform
```

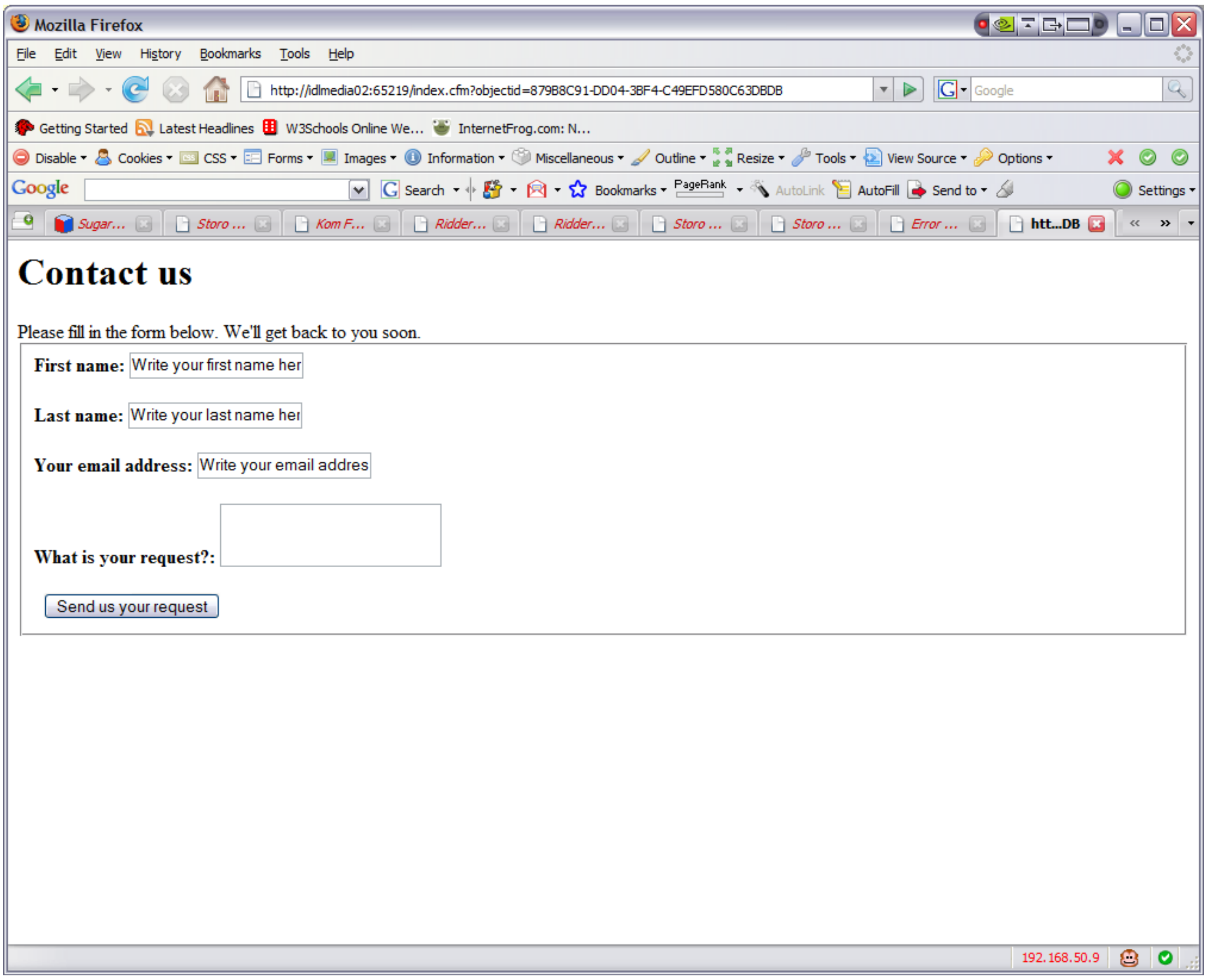
This is the `<cfdump var="#stobj#">` code we placed in the displayPageStandard.cfm file at play.

Now we can go to work on making it display something more usefull. Here's the code I use to display the form – it should not be to hard to understand what is going on here:

displayPageStandard.cfm

```
<cfoutput>
<h1>#stObj.title#</h1>
      <!--- set comma delimeted list with input types which should not have a
      label  --->
      <cfset noLabel = "statictext">
      <div class="description">#stObj.formheader#</div>
            <form action="" method="post" enctype="multipart/form-data"
            name="editform" class="idlform">
            <fieldset>
            <!--- loop over items --->
            <cfloop from="1" to="#arrayLen(stObj.aFormItems)#" index="i">
                  <cfset oFormItemService =
createObject("component","farcry.plugins.idlForm.packages.types.idlFormItem")>
                  <cfset oFormItem =
                        oFormItemService.getData(objectID=stObj.aFormItems[i])>

                  <div class="formline">
                        <!--- display (or not) label --->
                        <cfif not ListFind(noLabel,oFormItem.type)>
                              <label for="#oFormItem.objectid#">
                              <cfif Len(trim(oFormItem.title))>
                              <strong>#oFormItem.title#:</strong>
                              </cfif>
                              </label>
                        </cfif>

                        <cfswitch expression="#oFormItem.type#">
                              <cfcase value="textfield">
                                    <input name="#oFormItem.objectid#"
                                    type="text"
                                    value="#oFormItem.initValue#">
                              </cfcase>
                              <cfcase value="textarea">
                                    <textarea name="#oFormItem.objectid#"
                                    wrap="virtual">
                                    #oFormItem.initValue#
                                    </textarea>
                              </cfcase>
                              <cfcase value="checkbox">
                                    <input name="#oFormItem.objectid#"
                                    type="checkbox"
                                    value="#oFormItem.initValue#"
                                    <cfif oFormItem.initValue is 1>
                                    checked
                                    </cfif> />
                              </cfcase>
                              <cfcase value="radiobutton">
                                    <input name="#oFormItem.name#" type="radio"
                                    value="#oFormItem.initValue#"
                                    <cfif oFormItem.initValue is 1>
                                    checked
                                    </cfif> />
                                    </cfcase>
                                    <cfcase value="list">
                                          <select name="#oFormItem.objectid#">
                                          <cfloop list="#oFormItem.initValue#"
                                          index="i">
                                          <option value="#i#">#i#</option>
                                          </cfloop>
                                          </select>
                                    </cfcase>
                                    <cfcase value="filefield">
                                          <input name="#oFormItem.objectid#"
                                          type="file" />
                                    </cfcase>
                                    <cfcase value="statictext">
                                    #oFormItem.initValue#
                                    </cfcase>
                              </cfswitch>
                        </div>

                        <cfif oFormItem.linebreak is 1><br /></cfif>

                  </cfloop>
                  <div class="formline">
                  <label for="submitidlform">&nbsp;</label>
                  <input type="submit" name="submitidlform"
                  value="#stObj.submittext#" />
                  </div>
            </fieldset>
            </form>
      </div>
</cfoutput>
```

If we now make a refresh of our browser window we'll see something like this:


Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://idlmedia02:65219/index.cfm?objectid=879B8C91-DD04-3BF4-C49EFD580C63DBDB

Contact us

Please fill in the form below. We'll get back to you soon.

First name: Write your first name her

Last name: Write your last name her

Your email address: Write your email addres

What is your request?:

Send us your request

There we have it – the posibility to let the users for any of your FarCry 4 sites create their own forms. Was that easy or what?

Making it pretty

The form above is not very pretty, but how things should look can be very diferent from site to site, so naturaly we will controll the appearance of the form on a site level.

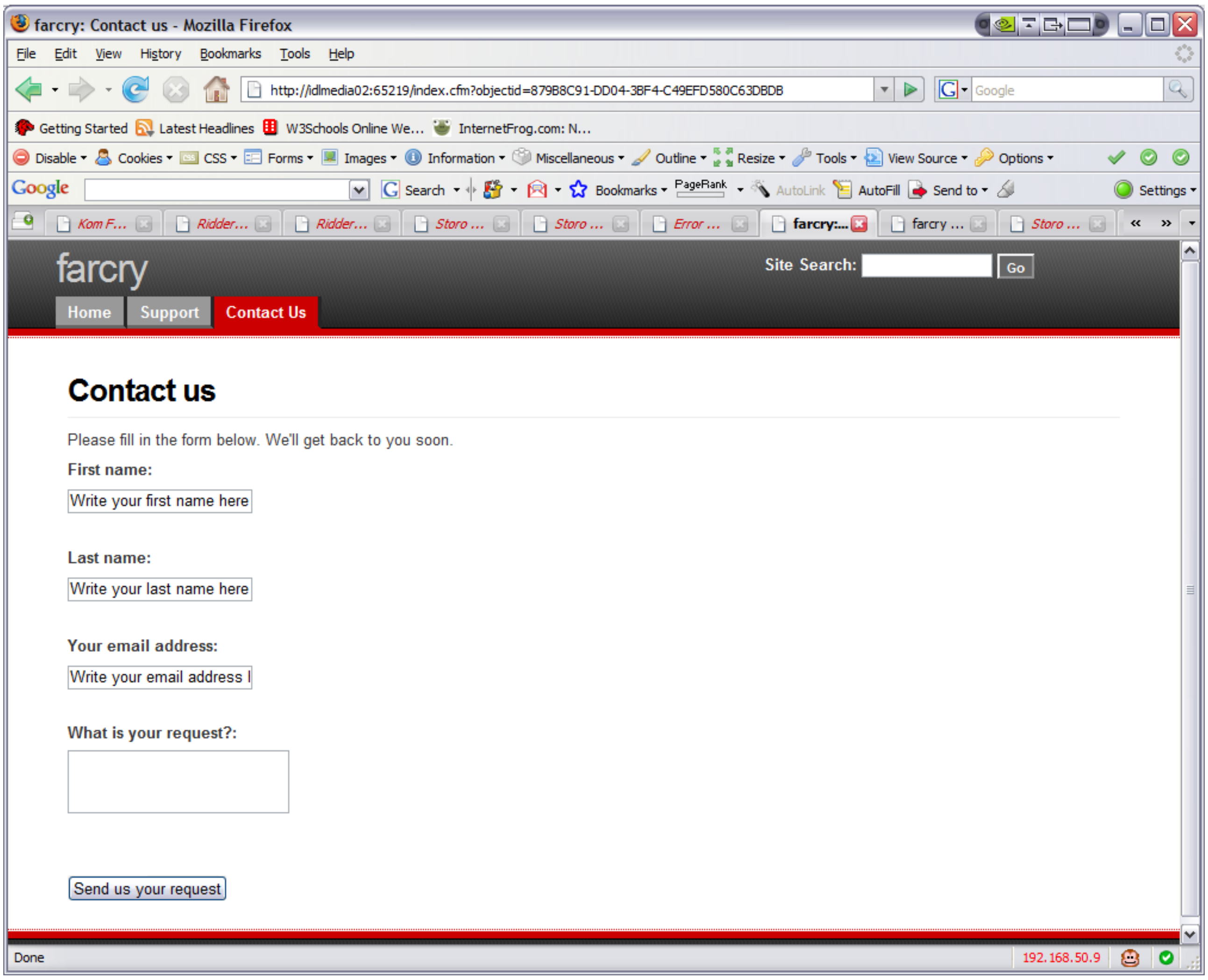
In the site I created for the example above I will make a idlform directory in the webskin directory. In this directory I will put a displayPageStandard.cfm file.

If you put a directory in webskin on the site like this – FarCry will look here instead of in the directory belonging to the plugin. However since the code for displaying the user generated form is already there we can reuse it – take a look at this code:

displayPageStandard.cfm *(in the webroot directory of the site)*

```
<cfsetting enablecfoutputonly="yes">

<!--- header --->
<cfmodule template=
"/farcry/projects/#application.applicationname#/webskin/includes/dmHeader.cfm"
      layoutClass="type-a"
      pageTitle="#stObj.title#">

<cfoutput>
      <div id="content">
</cfoutput>

<cfinclude
template="/farcry/plugins/idlform/webskin/idlform/displayPageStandard.cfm">

<cfoutput>
      </div>
</cfoutput>
<!--- footer --->
<cfmodule template=
"/farcry/projects/#application.applicationname#/webskin/includes/dmFooter.cfm">
<cfsetting enablecfoutputonly="no">
```

Here we use the default FarCry cms way of including the header and footer – and the we cfinclude the displayPageStandard.cfm that we have for the plugin. Refresh app scope and you'll see this:


farcry: Contact us - Mozilla Firefox
http://idlmedia02:65219/index.cfm?objectid=879B8C91-DD04-3BF4-C49EFD580C63DBDB
farcry
Site Search:
Go
Home
Support
Contact Us
Send us your request
Done

Contact us

Please fill in the form below. We'll get back to you soon.

First name:

Write your first name here

Last name:

Write your last name here

Your email address:

Write your email address l

What is your request?:

Making it work

We now have a pretty form generated by the user, and they can generate as many as they want. But so far the form does not work – it does not do anything when it is submitted.

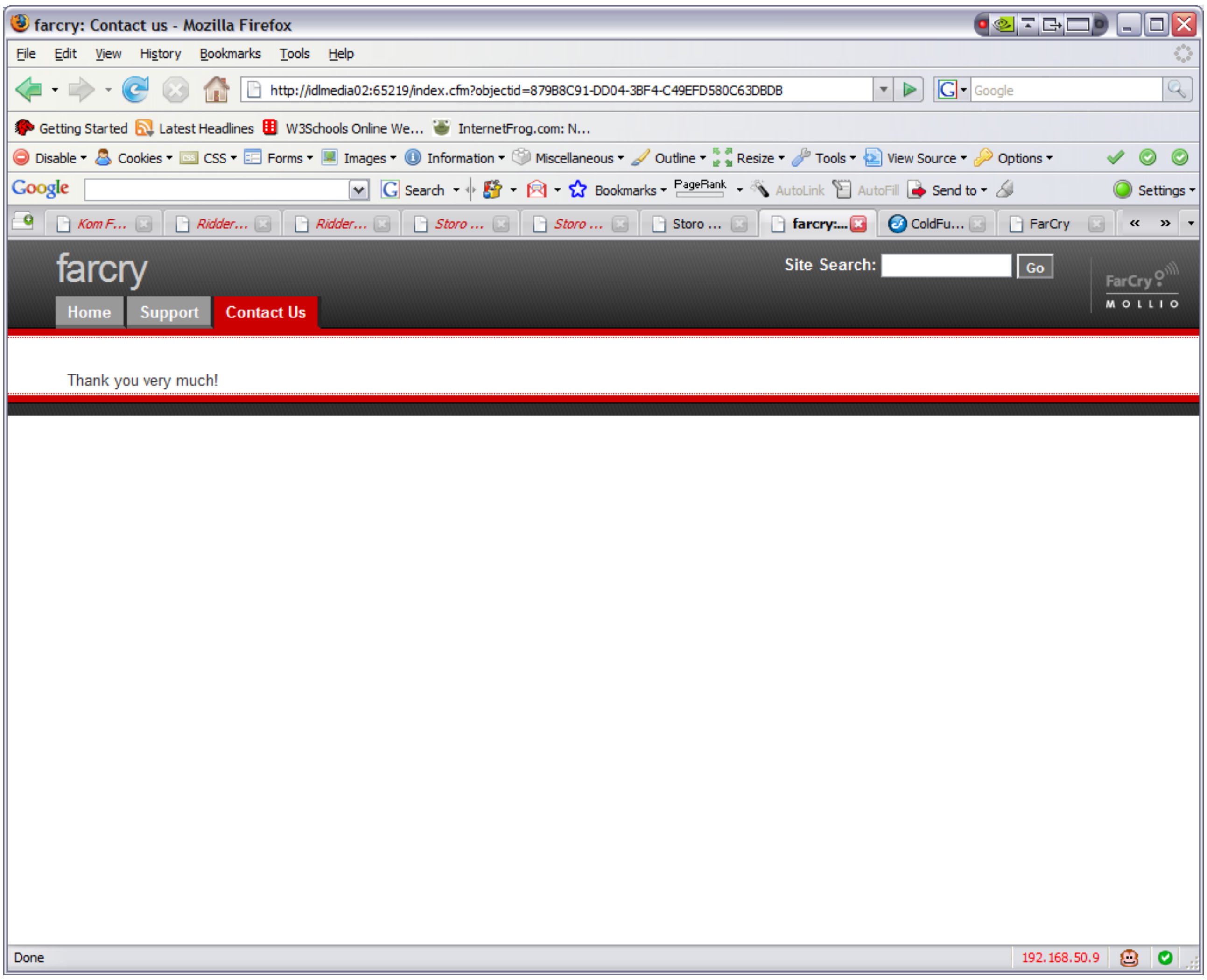
Let's rectify that. First we open up the displayPageStandard.cfm file we have under the plugin directory. Here we will add a cfif statement like so:

```
<cfoutput>
<cfif StructKeyExists(form,"submitidlform")>
            <!--- send the content of the submited form by e-mail --->
            <cfinvoke component="farcry.plugins.idlForm.packages.types.idlForm"
            method="submit">
                  <cfinvokeargument name="objectId" value="#stobj.objectId#"/>
                  <cfinvokeargument name="formData" value="#form#"/>
            </cfinvoke>
      #stObj.sendt#
<cfelse>

      THE CODE WE HAD FROM BEFORE HERE

</cfif>
</cfoutput>
```

Ok – we see that we call a function called method on the idlForm.cfc – so we need to add that one. We've been spoiled so far in not having to write much code to handle busines logic, as FarCry have beed doing all the heavy lifting for us. However now is the time to do some coding ourselves. In idlForm.cfc we insert the following:

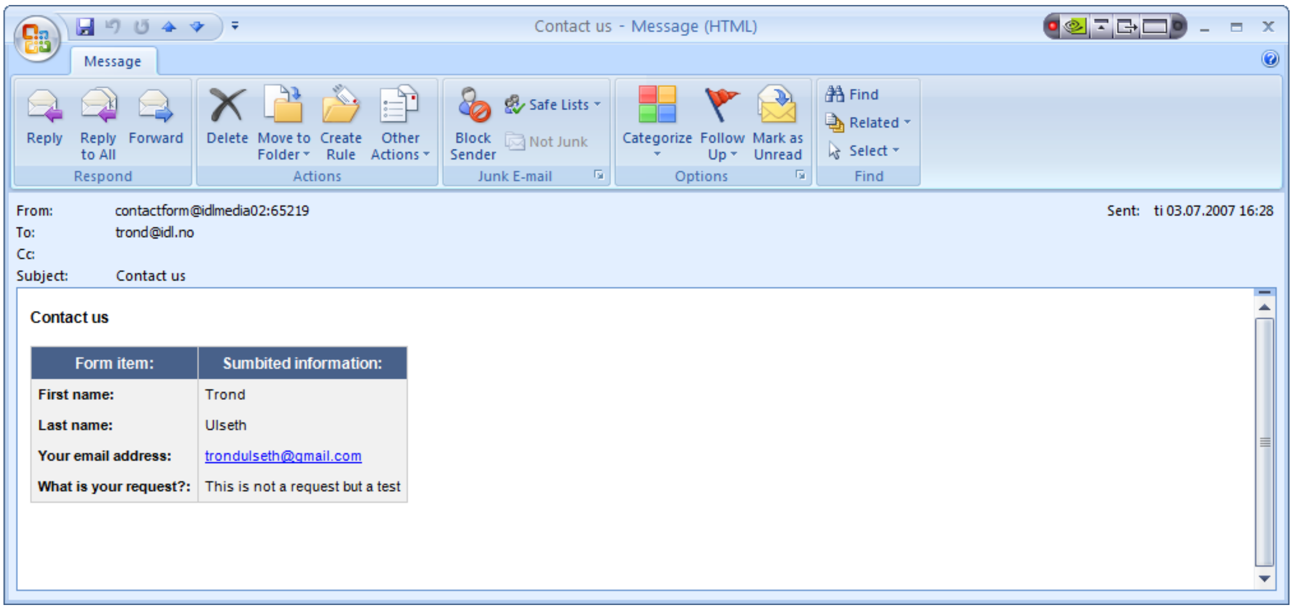
```
<cffunction name="submit" access="public" output="false" returntype="void">
            <cfargument name="objectid" required="yes" type="uuid">
            <cfargument name="formData" required="yes" type="struct">

            <!--- getData for object --->
            <cfset var stObj = this.getData(arguments.objectid)>

            <!--- create structure to hold submited files --->
            <cfset var uploadfile = StructNew()>

            <!--- upload all submited files and add them to the uploadfile
            structure  --->
            <cfloop from="1" to="#arrayLen(stObj.aFormItems)#" index="i">
                  <cfset oFormItemService =
createObject("component","farcry.plugins.idlForm.packages.types.idlFormItem")>
                  <cfset oFormItem =
                  oFormItemService.getData(objectID=stObj.aFormItems[i])>
            <cfif oFormItem.type is "filefield" and
            Len(formData[oFormItem.objectid])>
                  <cffile action="upload" filefield="#oFormItem.objectid#"
                  destination="#application.defaultfilepath#"
                  nameconflict="makeunique">
                  <cfset uploadfile[#oFormItem.objectid#] =
                  "#cffile.serverDirectory#\#cffile.serverFile#">
            </cfif>
            </cfloop>
            <!--- invoke method to send an email with the submited data --->
            <cfinvoke method="sendMail">
                  <cfinvokeargument name="objectId"
                        value="#arguments.objectId#"/>
                  <cfinvokeargument name="formData"
                        value="#arguments.formData#"/>
                  <cfinvokeargument name="stObj" value="#stObj#">
                  <cfinvokeargument name="uploadfile" value="#uploadfile#">
            </cfinvoke>

      </cffunction>

      <cffunction name="sendMail" access="private" output="false"
      returntype="void">
            <cfargument name="objectid" required="yes" type="uuid">
            <cfargument name="formData" required="yes" type="struct">
            <cfargument name="stObj" required="yes" type="struct">
            <cfargument name="uploadfile" required="yes" type="struct">

            <cfif Len(arguments.stObj.receiver)>
            <cfmail to="#arguments.stObj.receiver#"
            from="contactform@#cgi.HTTP_HOST#"
            subject="#arguments.stObj.title#" type="html">

                  <style type="text/css">
                  <!--
                  body {
                        font: 12px Arial, Helvetica, sans-serif;
                  }
                  th, td {
                        padding: 5px;
                  }
                  th {
                        background-color: ##48618A;
                        color: ##FFFFFF;
                        font-size: 12px;
                  }
                  td {
                        background-color: ##F1F1F1;
                        font-size: 11px;
                  }
                  .or {
                        color: ##E17000;
                  }
                  -->
                  </style>
                  <body>
                  <strong>#arguments.stObj.title#</strong><br>
                  &nbsp;<br>
                  <table cellspacing="1" bgcolor="##CCCCCC">
                  <tr>
                  <th scope="col">Form item:</th>
                  <th scope="col">Sumbited information:</th>
                  </tr>
                  <cfloop from="1" to="#arrayLen(arguments.stObj.aFormItems)#"
                  index="i">
                        <cfset oFormItemService =
createObject("component","farcry.plugins.idlForm.packages.types.idlFormItem")>
                        <cfset oFormItem =
            oFormItemService.getData(objectID=arguments.stObj.aFormItems[i])>
                        <cfif oFormItem.type is "radiobutton">
                        <tr><td><strong>#oFormItem.title#:</strong></td>
                        <td>#arguments.formData[oFormItem.name]#</td></tr>
                        <cfelseif oFormItem.type is "filefield" and
                  StructKeyExists(arguments.uploadfile,oFormItem.objectid)>
                              <cfmailparam
                        file="#arguments.uploadfile[oFormItem.objectid]#">
                              <tr><td><strong>#oFormItem.title#:</strong></td>
                              <td class="or">
                  #ListLast(arguments.uploadfile[oFormItem.objectid],"\")#
                              </td></tr>
                        <cfelse>
                        <tr><td><strong>#oFormItem.title#:</strong></td><td>
                        <cfif
                        StructKeyExists(arguments.formData,oFormItem.objectid)>
                        #arguments.formData[oFormItem.objectid]#
                        </cfif></td></tr>
                        </cfif>
                  </cfloop>
                  </table>
                  </body>
            </cfmail>
            </cfif>
      </cffunction>
```

Ok – now if we submit the form we get this:


farcry: Contact us - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://idlmedia02:65219/index.cfm?objectid=879B8C91-DD04-3BF4-C49EFD580C63DBDB
farcry
Site Search:
Go
Home
Support
Contact Us
Done
192.168.50.9

Thank you very much!

And checking my email I have this in my inbox:



From: contactform@idlmedia02:65219 Sent: ti 03.07.2007 16:28
To: trond@idl.no
Cc:
Subject: Contact us

Contact us

Form item:	Sumbited information:
First name:	Trond
Last name:	Ulseth
Your email address:	trondulseth@gmail.com
What is your request?:	This is not a request but a test

Ok – that should cover it for this lesson. There are both enhansments and error fixes that is not included in this little tutorial, but the code for the complete plugin will be made available from IDLmedia AS (my company) – both through my blog and through the farcrycms.org website.

I hope I've managed to convey how easy it is to use FarCry to develop plugins – and in fact full applications. I am still very new to plugins and formtols, and I've barely scratched the surface here.

Happy FarCry'ing

Trond Ulseth
IDLmedia AS
www.idl.no