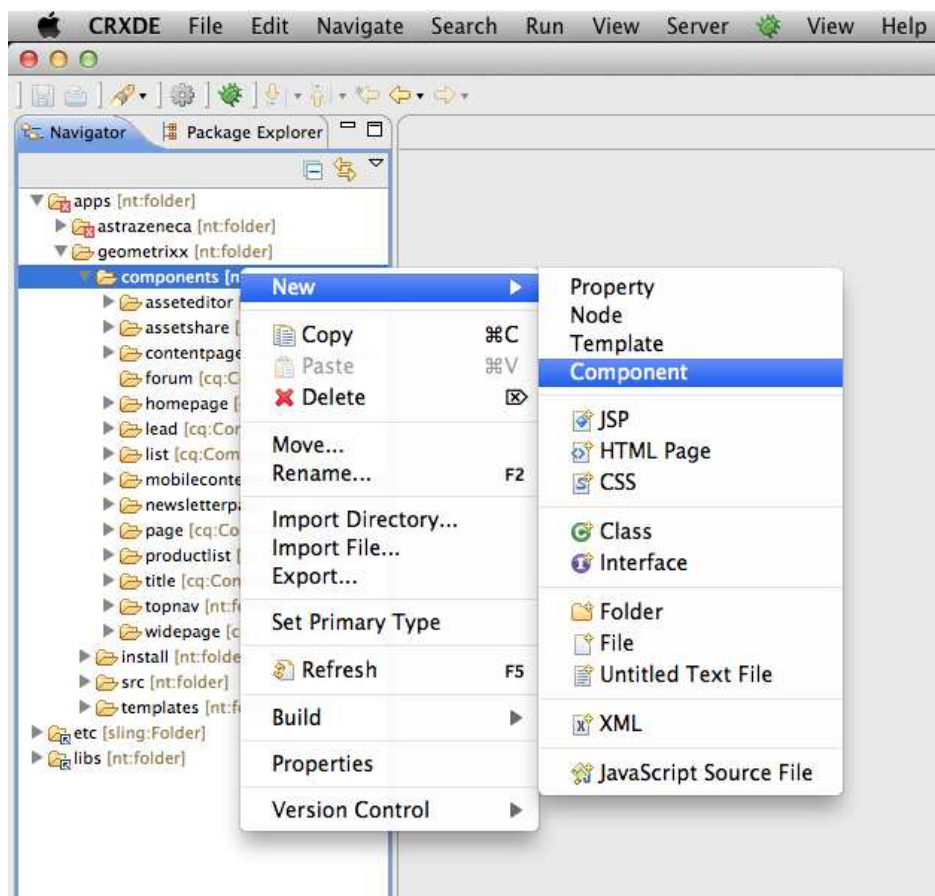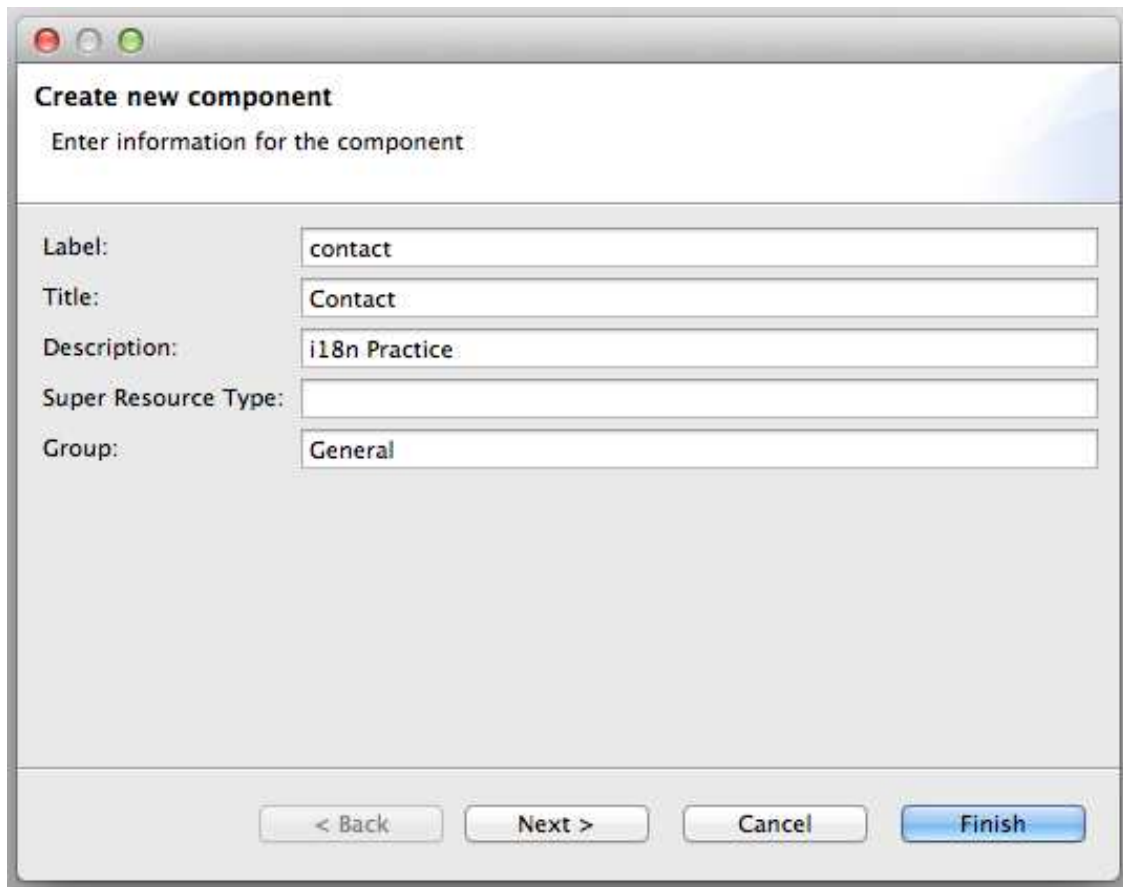# CQ5 Internationalization (i18n)

i18n capabilities allow you to provide dynamic "messages" based on the visitors language.

Internationalization is stored in the repository under nodes (nodeType sling:Folder) named i18n. The children nodes (nodeType sling:Folder + mixin mix:language) of the i18n node represent languages and are named after the ISO code of the languages that should be supported (e.g. es, en, de, fr, etc.). Below these language nodes are the message nodes (nodeType sling:MessageEntry), which will contain a simple key-message pair.

Practice:

1. In order to test internationalization features, we're going to create a new Contact component under the existing geometrixx demo site
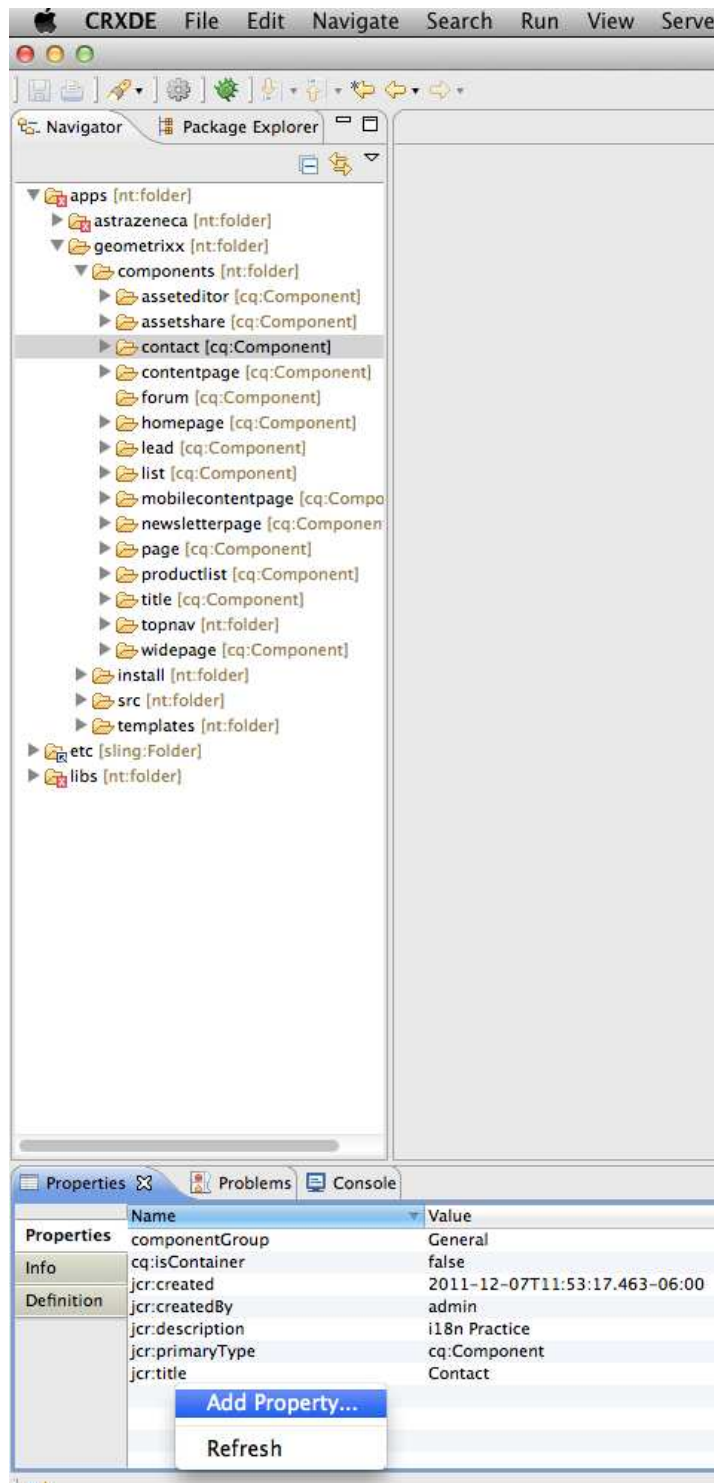
Create new component

Enter information for the component

Label:	contact
Title:	Contact
Description:	i18n Practice
Super Resource Type:	
Group:	General

< Back   Next >   Cancel   Finish
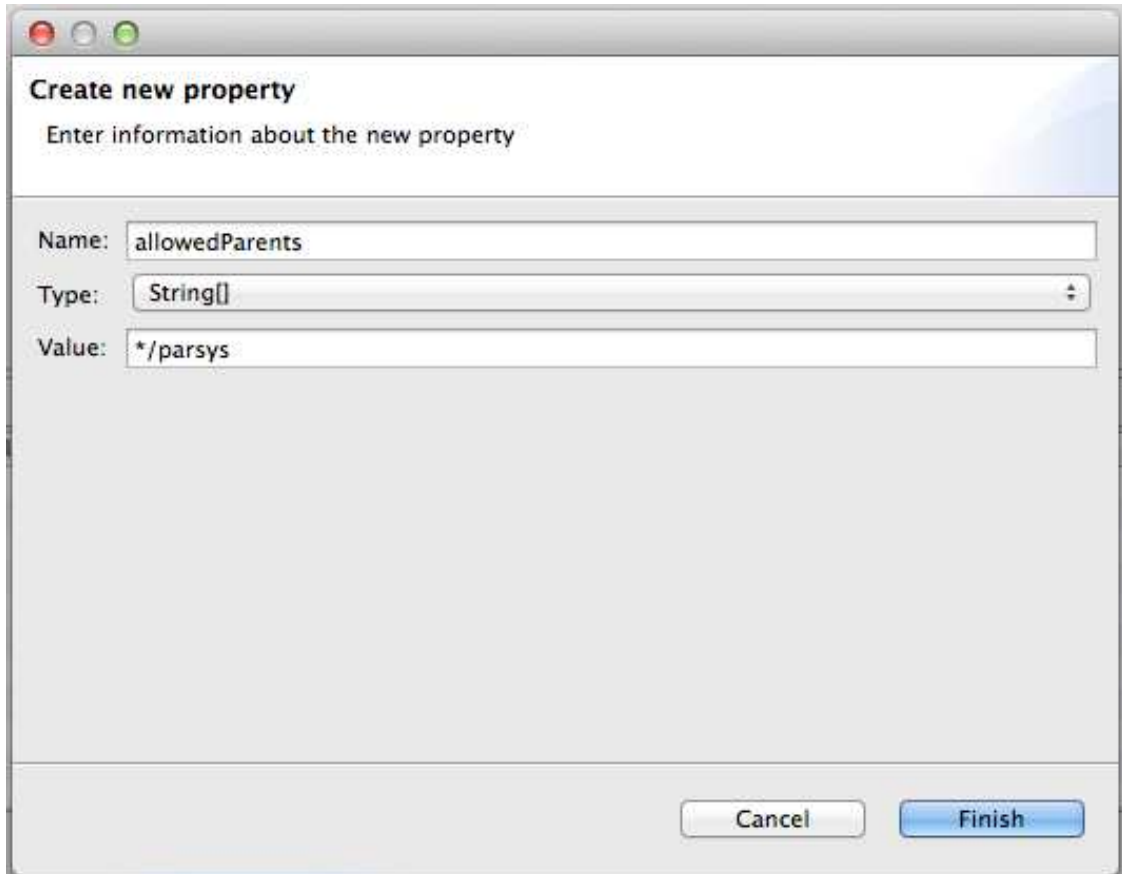
2. Then add a new property for the Contact component:

Name : allowedParents
Type : String[]
Value : */parsys

3. Create a new Dialog for the Contact Component.

4. Create an items node (nodeType cq:WidgetCollection) under the tab 1 node of the

5. Open the **contact.jsp** file and fill it with the following content:

```jsp
<%@include file="/libs/foundation/global.jsp"%>

<table>
    <tr>
        <td>
            <label for="Name">Name</label>
        </td>
        <td>
            <input type="text" name="Name" id="Name" />
        </td>
    </tr>
    <tr>
        <td>
            <label for="City">City</label>
        </td>
        <td>
            <input type="text" name="City" id="City" />
        </td>
    </tr>
    <tr>
        <td>
            <label for="Email">Email</label>
        </td>
        <td>
            <input type="text" name="Email" id="Email" />
        </td>
    </tr>
    <tr>
        <td>
            <label for="Message">Message</label>
        </td>
        <td>
            <textarea name="Message" id="Message" rows="10"
cols="15"></textarea>
        </td>
    </tr>
    <tr>
        <td colspan="2" align="right">
            <input type="submit" name="Submit" id="Submit" />
        </td>
    </tr>
</table>
```
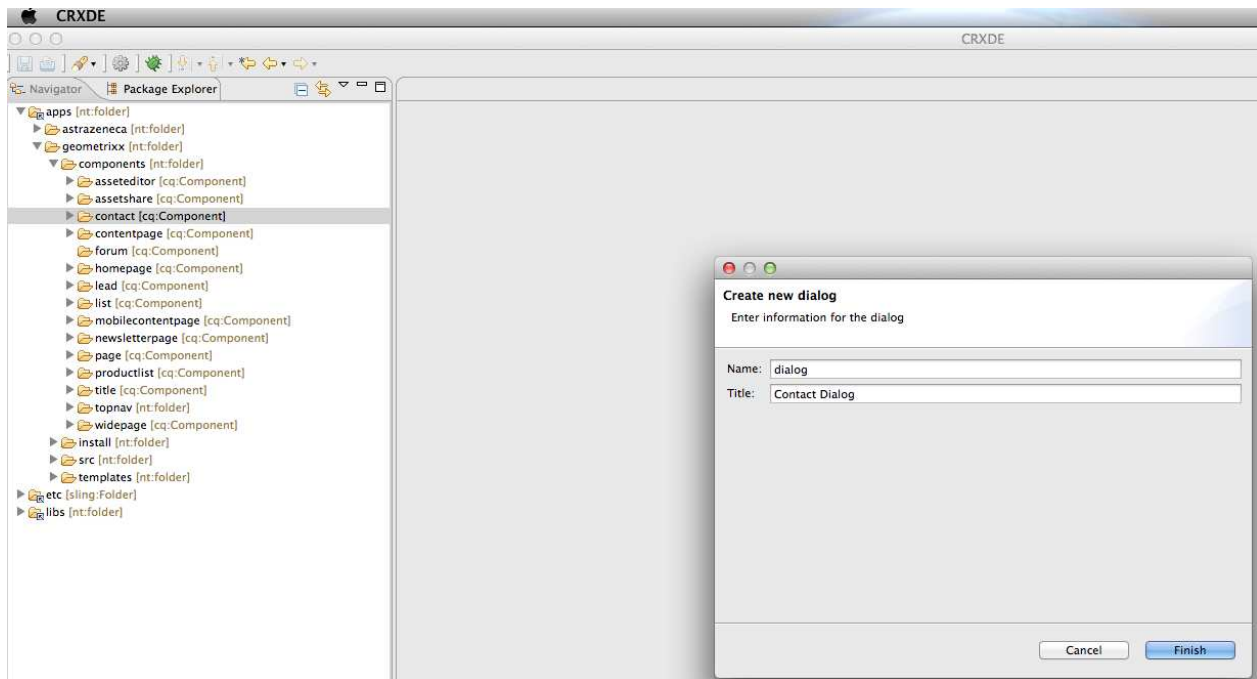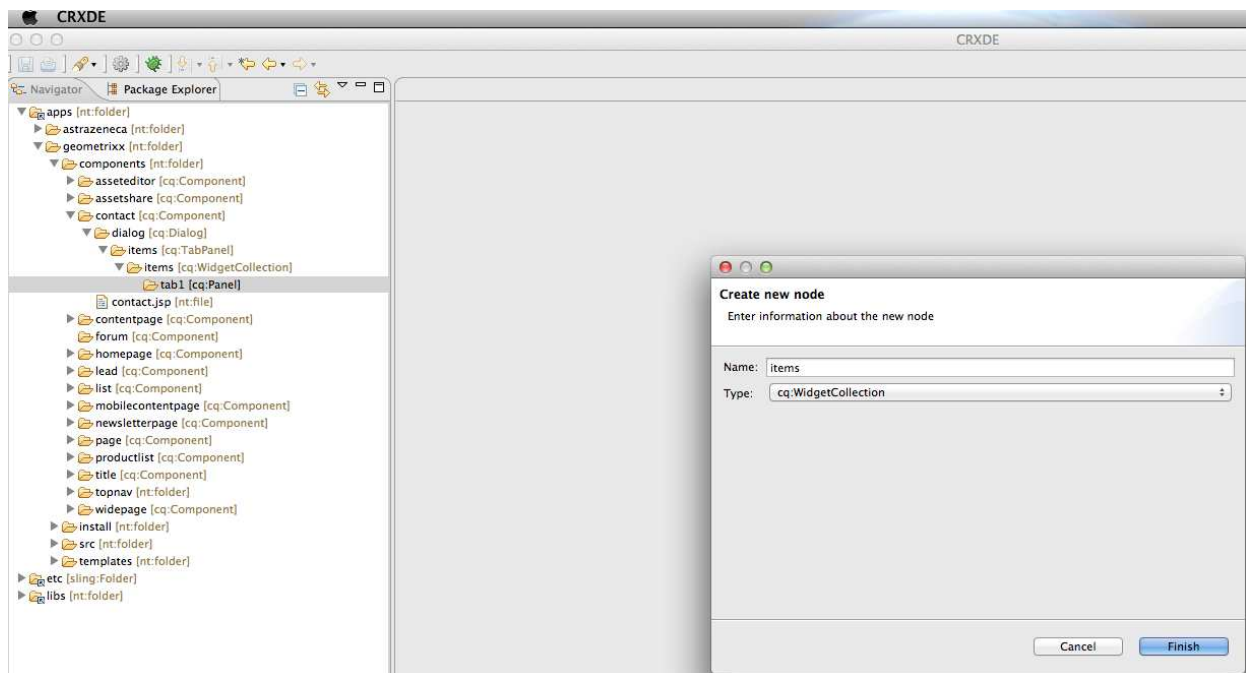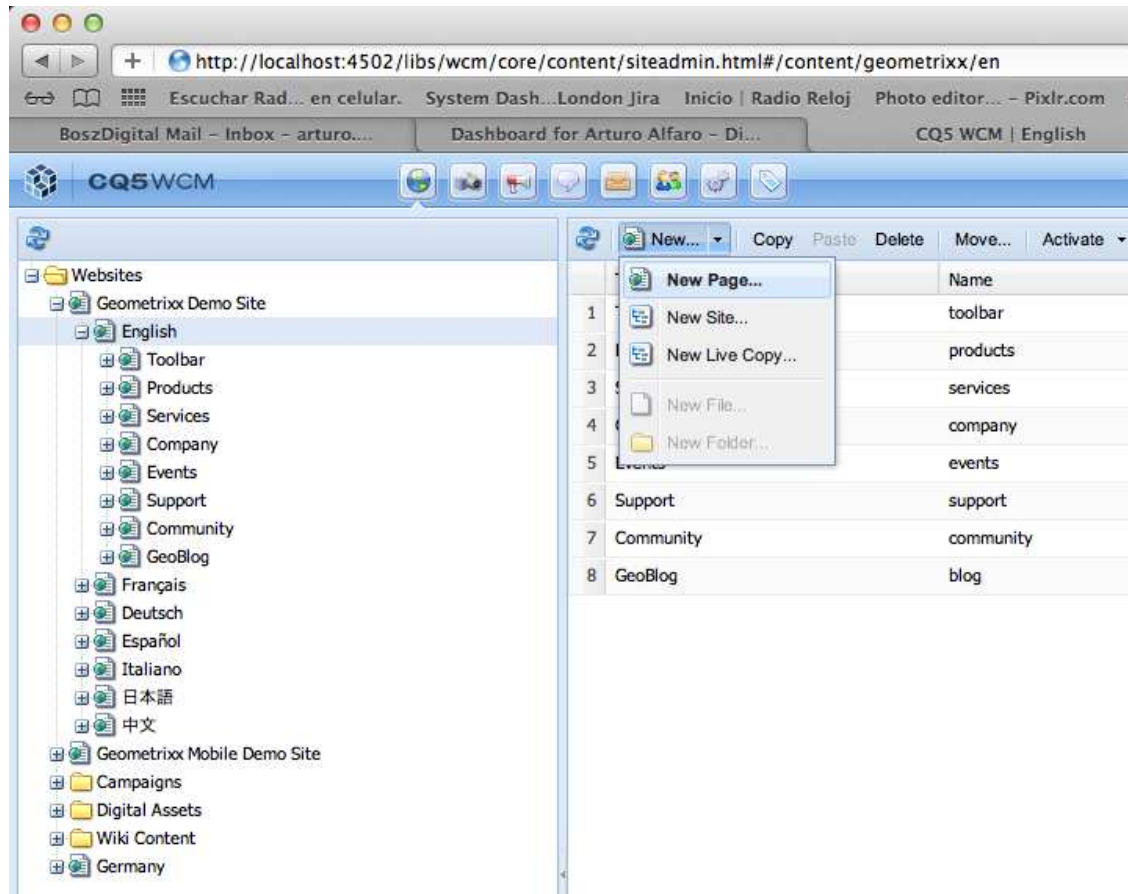
6. Then go to CQ5 WCM and under : Websites/Geometrixx Demo Site/English
Create a new Page "Contact"
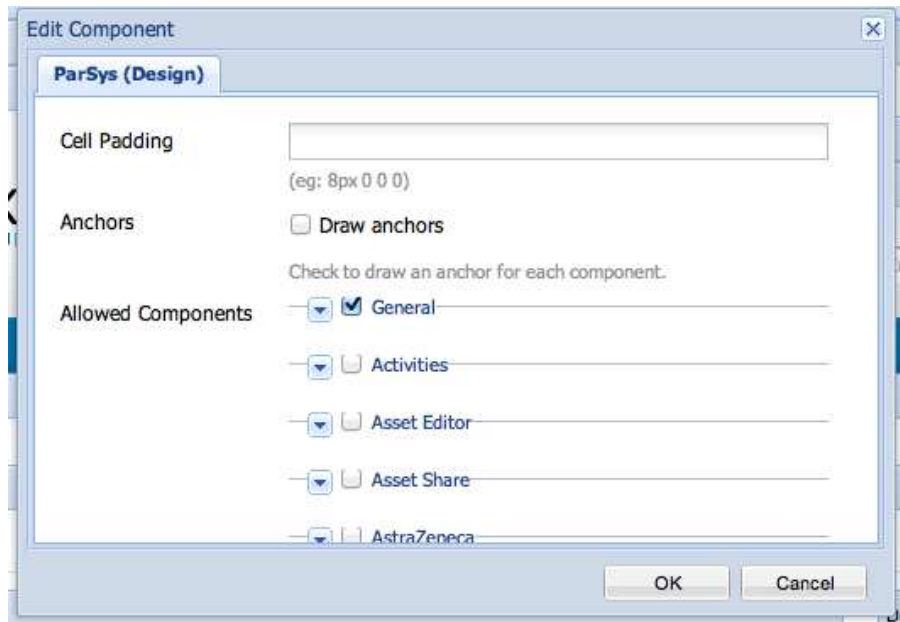
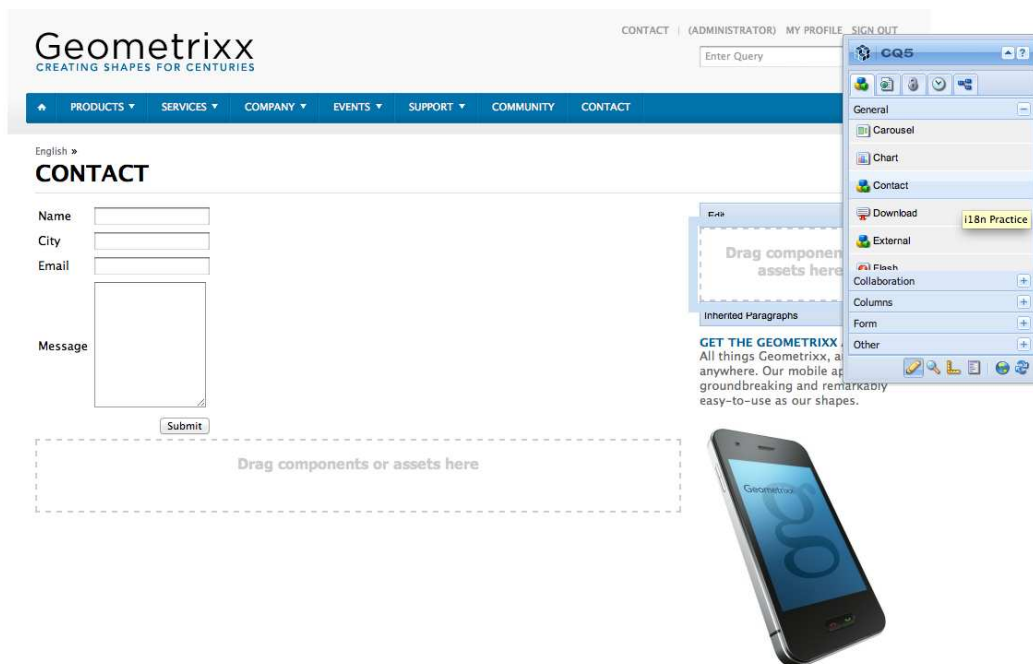7. Open the just created Contact page, go to design mode on the sidekick, then edit Design of par

8. Check General on Allowed Components section

9. Then go back to the author mode, and drag and drop the Contact Component to the parsys section, as result you must see the following screen:

10. Now let's start working on i18n stuff, create an *i18n* node (nodeType sling:Folder) under the root node of the title Component.
(e.g. /apps/geometrixx/components/contact)
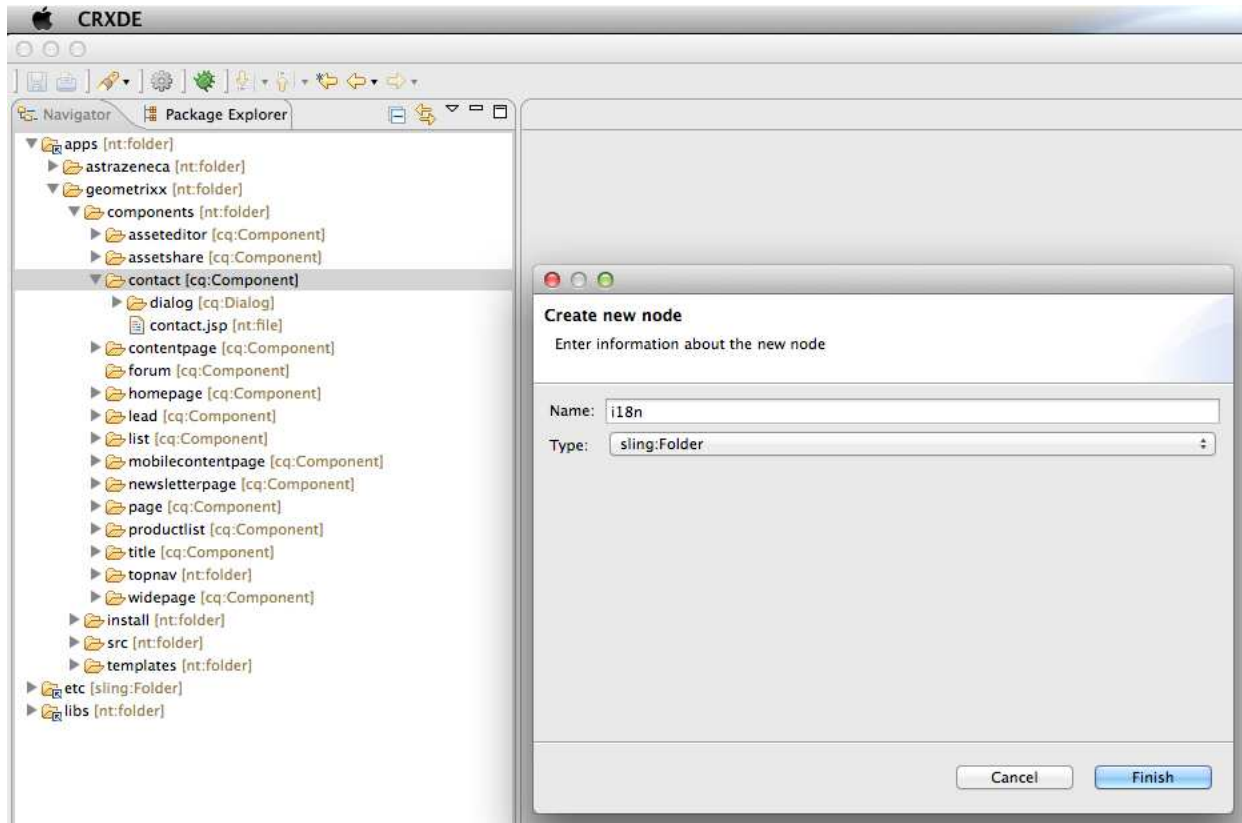
11. Create an *en* node (nodeType sling:Folder) under the newly created i18n node.

12. Open a Content Explorer window and assign the *mixin*, mix:language to the newly created en node. Although they appear to be represented as properties, *mixins* are used to add functions to a node. They are basically reusable feature sets that can be added to any node type.
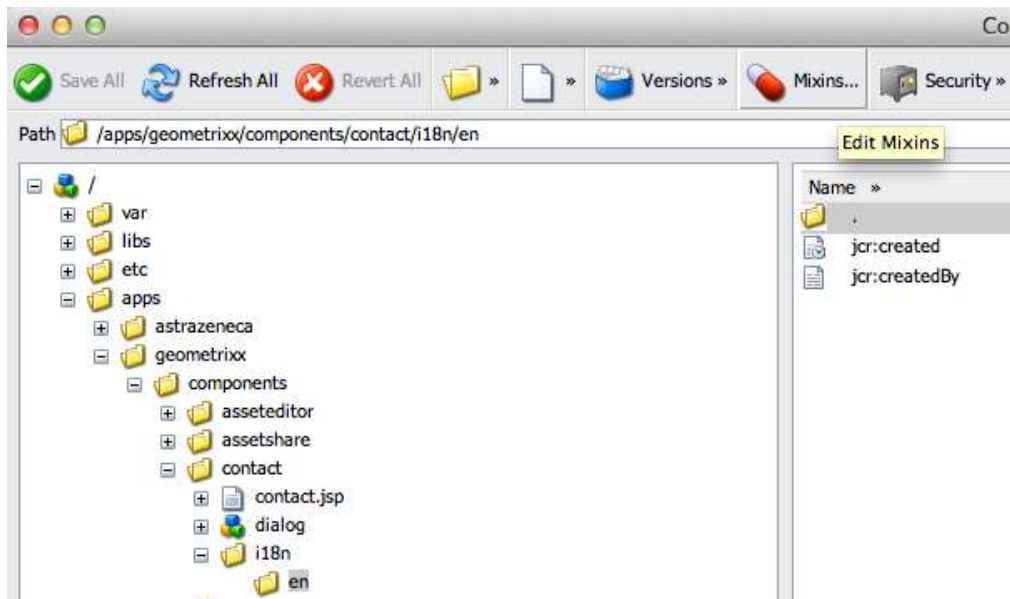
13. Choose mixin (mix:language)
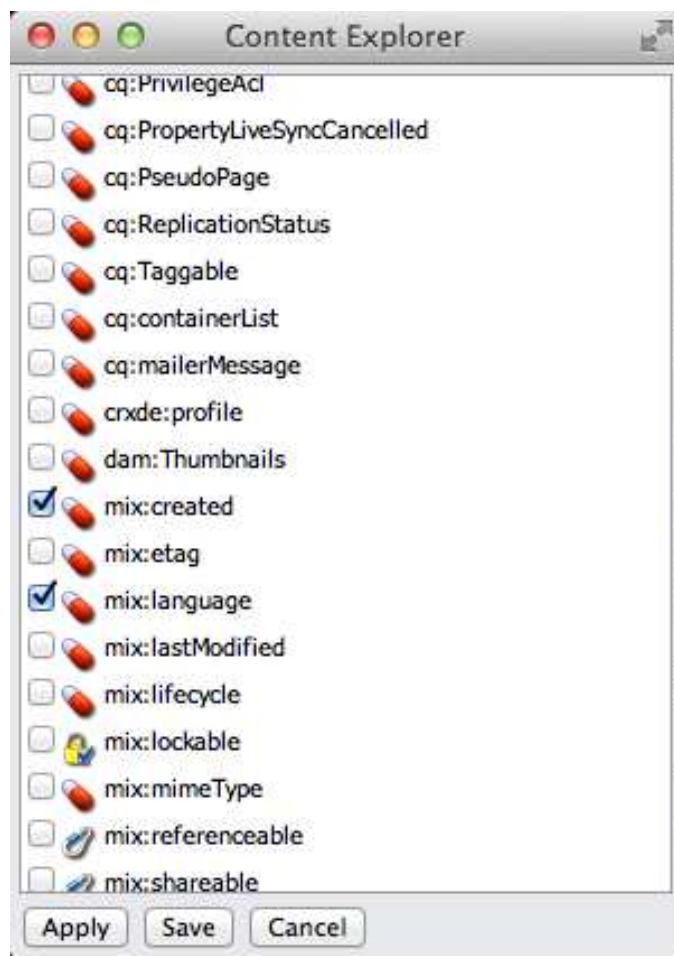
14. Return to CRXDE. Refresh the *en* node.

15. Set the following property on the newly created *en* node:

Name : jcr:language
Type : String
Value : en

16. Create a *label_name* node (nodeType sling:MessageEntry) under the newly created *en* node.

17. Assign the following properties to the newly created *label_name* node:

- The property that identifies the message key
    - Name = sling:key
    - Type = String
    - Value = label_name
- The property that identifies the message
    - Name = sling:message
    - Type = String
    - Value = Name

18. Create a *label_city* node (nodeType sling:MessageEntry) under the newly created *en* node.

19. Assign the following properties to the newly created *label_ city* node:

- The property that identifies the message key
    - Name = sling:key
    - Type = String
    - Value = label_city
- The property that identifies the message
    - Name = sling:message
    - Type = String
    - Value = City

20. Create a *label_email* node (nodeType sling:MessageEntry) under the newly created *en* node.

21. Assign the following properties to the newly created *label_email* node:

- The property that identifies the message key
    - Name = sling:key
    - Type = String
    - Value = label_email
- The property that identifies the message
    - Name = sling:message
    - Type = String
    - Value = Email

22. Create a *label_message* node (nodeType sling:MessageEntry) under the newly created *en* node.

23. Assign the following properties to the newly created *label_message* node:
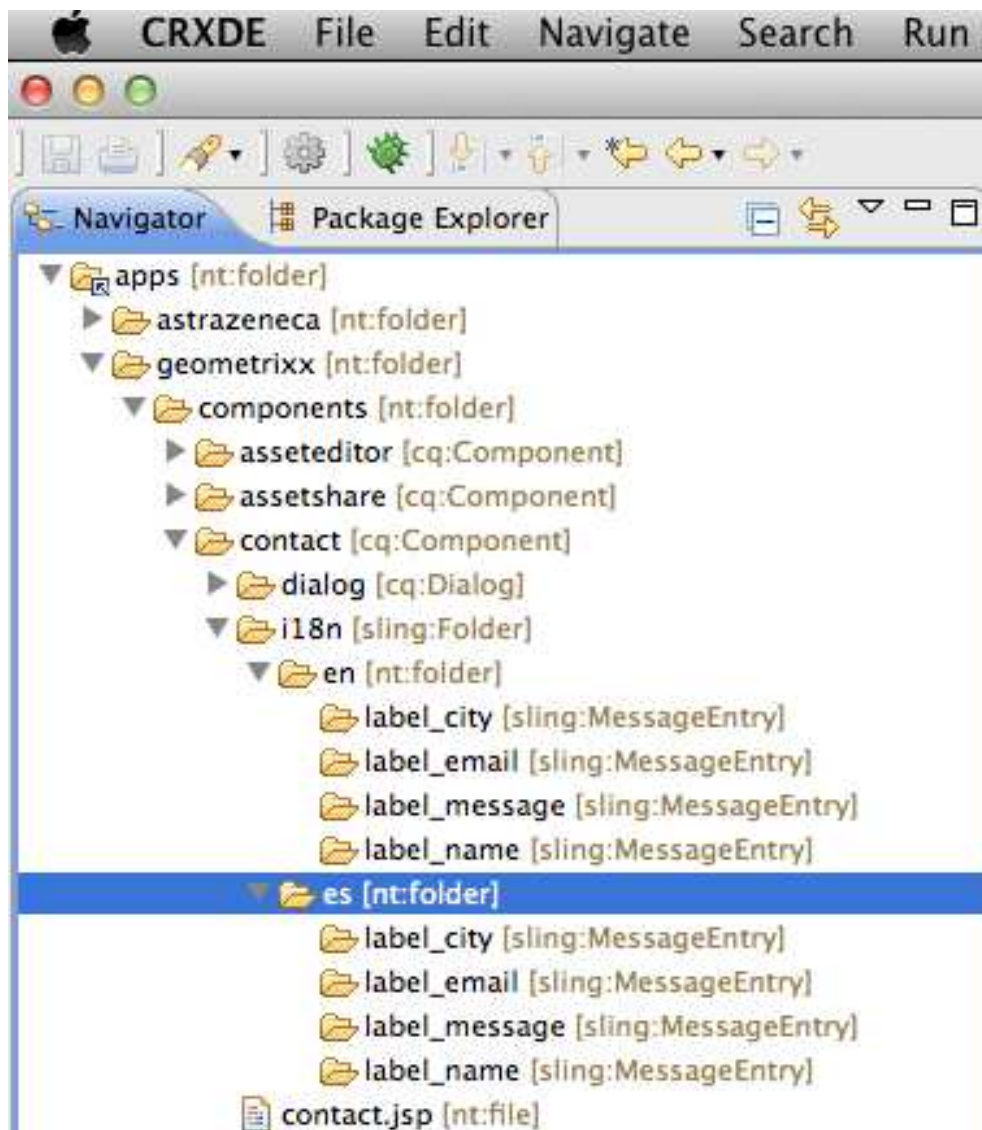
- The property that identifies the message key
  - Name = sling:key
  - Type = String
  - Value = label_message
- The property that identifies the message
  - Name = sling:message
  - Type = String
  - Value = Message

24. Copy the *en* node and paste as *es* under i18n

25. Change the value the property jcr:language under *i18n / es* node
   - Value = es

26. Change the value the property sling:message under *es / label_name* node
   - Value = Nombre

27. Change the value the property sling:message under *es / label_city* node
   - Value = Ciudad

28. Change the value the property sling:message under *es / label_email* node
   - Value = Correo Electronico

29. Change the value the property sling:message under *es / label_message* node
   - Value = Mensaje

30. Copy the *en* node and paste as *fr* under i18n

31. Change the value the property jcr:language under *i18n / fr* node
   - Value = fr

32. Change the value the property sling:message under *fr / label_name* node
   - Value = Nom

33. Change the value the property sling:message under *fr / label_city* node
   - Value = Ville

34. Change the value the property sling:message under *fr / label_email* node
   - Value = Courrier

35. Change the value the property sling:message under *fr / label_message* node
   - Value = Noun

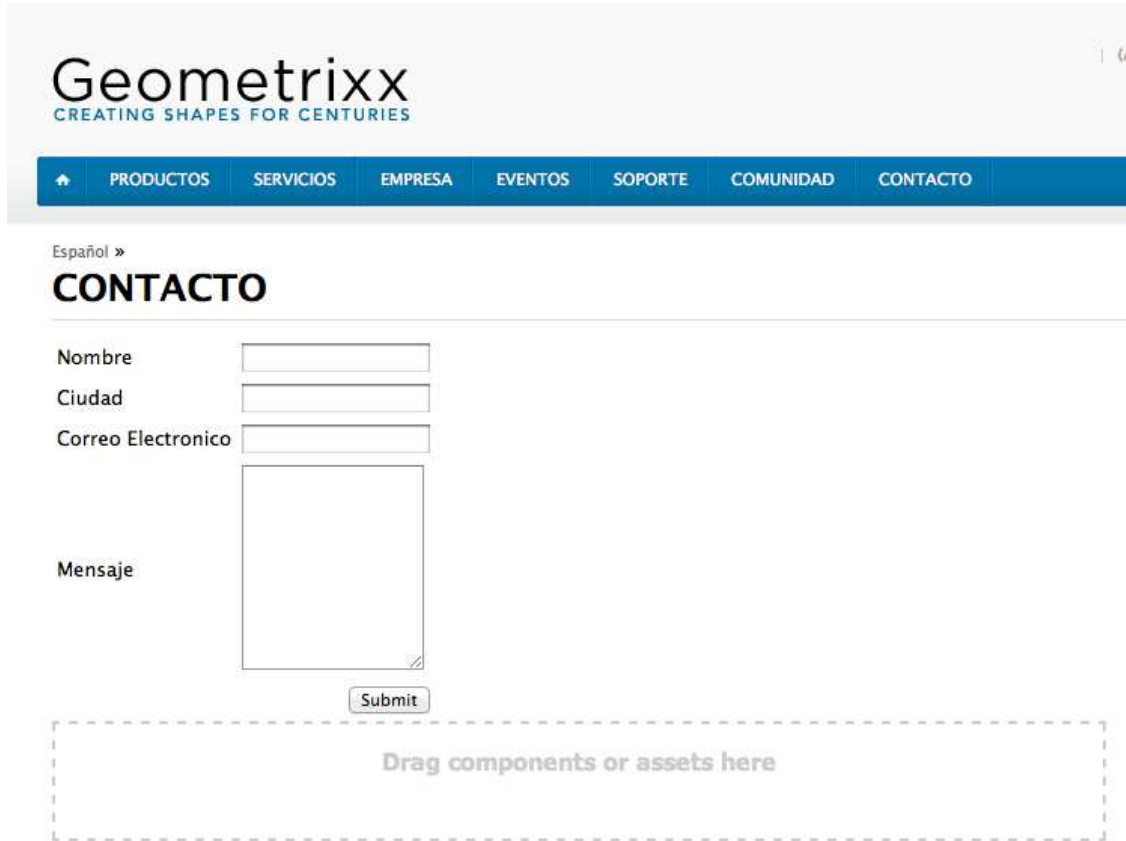36. Modify the contact.jsp file as follows: (**Changes are in red**)

```jsp
<%@include file="/libs/foundation/global.jsp"%>

<cq:setContentBundle/>

<table>
    <tr>
        <td>
            <label for="Name"><fmt:message key="label_name"/></label>
        </td>
        <td>
            <input type="text" name="Name" id="Name" />
        </td>
    </tr>
    <tr>
        <td>
            <label for="City"><fmt:message key="label_city"/></label>
        </td>
        <td>
            <input type="text" name="City" id="City" />
        </td>
    </tr>
    <tr>
        <td>
            <label for="Email"><fmt:message
key="label_email"/></label>
        </td>
        <td>
            <input type="text" name="Email" id="Email" />
        </td>
    </tr>
    <tr>
        <td>
            <label for="Message"><fmt:message
key="label_message"/></label>
        </td>
        <td>
            <textarea name="Message" id="Message" rows="10"
cols="15"></textarea>
        </td>
     </tr>
    <tr>
        <td colspan="2" align="right">
            <input type="submit" name="Submit" id="Submit" />
        </td>
    </tr>
</table>
```

37. Then go to CQ5 WCM and under : Websites/Geometrixx Demo Site/Español Create a new Page "Contacto"

38. Drag and drop the Contact Component to the parsys section of the new Page, as result you must see the following screen:

39. Then go to CQ5 WCM and under : Websites/Geometrixx Demo Site/Français Create a new Page "Contactez"

40. Drag and drop the Contact Component to the parsys section of the new Page, as result you must see the following screen:

As you can see the contact form component was dynamically translate to the correct language depending of the settings of the page.