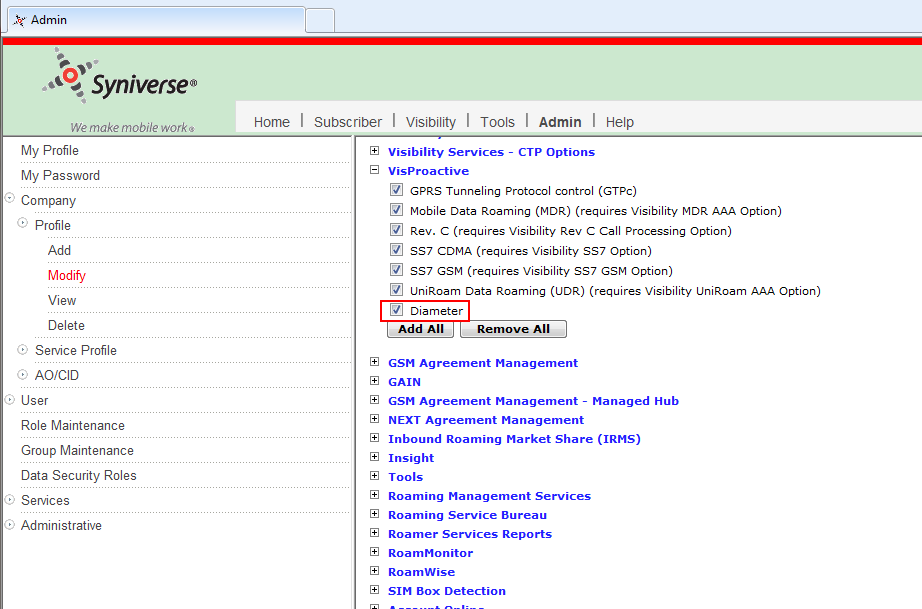
## Integrate VisProactive-Diameter into xroads 6.0 design document

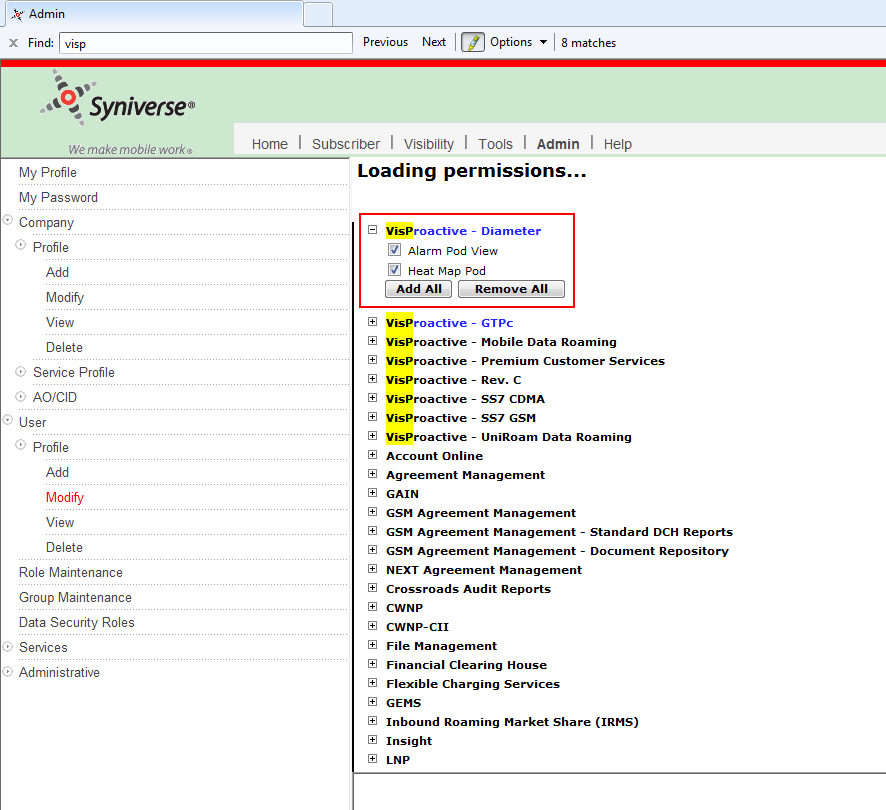
### Purpose of Integrating VisPro\_Diameter into xroads 6.0

The purpose is to add company permission, user permission and menu items as following

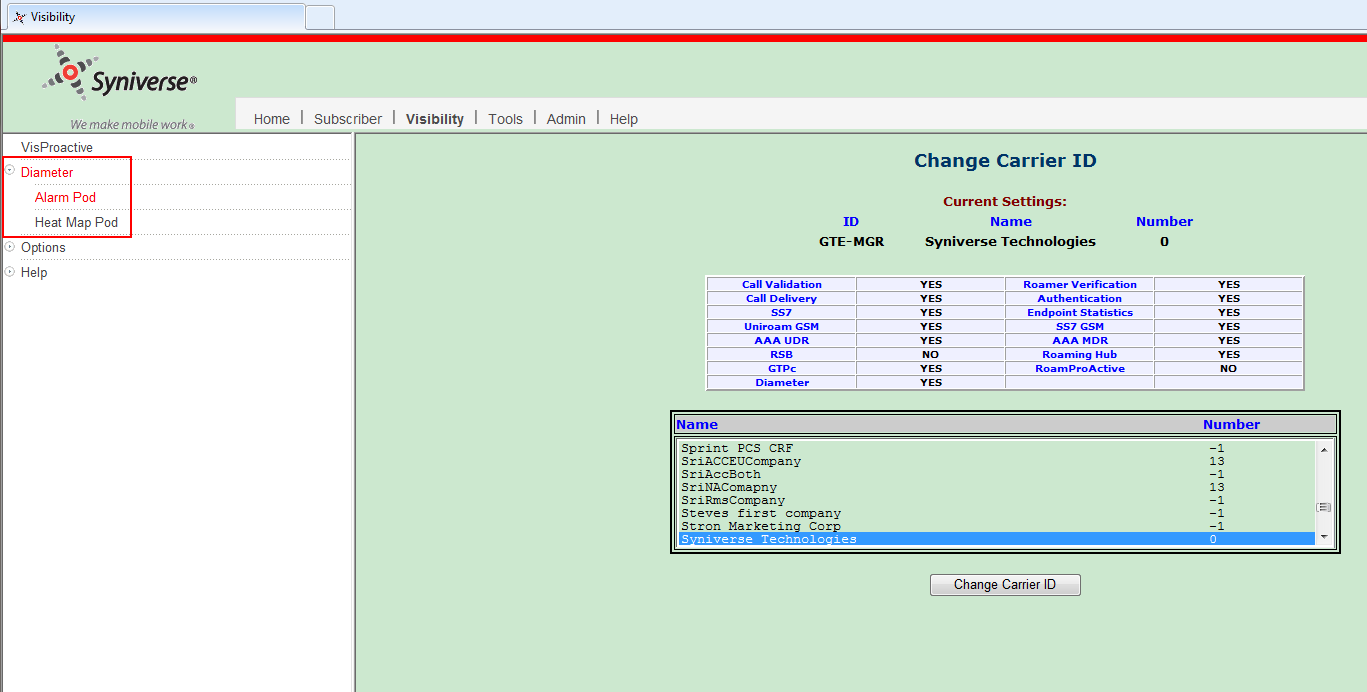
**Company Permission**



**User Permission**



**Menu items**



### How to achieve this purpose

Do two things to achieve this purpose

1. Insert records into database

After insert appropriate records into database, company permission, user permission and menu items can be shown on GUI

1. Add ldif data into LDAP server

When third party accesses xroads6.0, instead of fetching data from database, the data is fetched from LDAP server. So the data we insert into database should also be inserted into LDAP

We need to generate sql script to insert records into database and generate ldif file to insert data into LDAP server

We maintain a file used to generate sql script and ldif file, so what we now need to do is add a new instance into this file, with this new instance, we can generate both sql script and ldif file

Refer to IntegrateVisPro\_DiameterToXroads6.0\_RealseNotes.docx to see how to run sql&ldif

### How to add new instance

We add a new instance VisProactive\_Diameter) into GenerateLDIF file, with this new instance, we can generate both sql script and ldif file

In different environment, the URLs the menu items pointing to are different, so we define a variable VISPRO\_DIAMETER, and assign corresponding URL values to this variable in different environment.

Here is the matrix of environments and URL

|  |  |  |
| --- | --- | --- |
| Environments | Server box name | URL |
| 6.xx DEV | xroads-ofmd1 | vis-diameter-portal-dev-alt.syniverse.com |
| 6.xx TEST | xroads-oamt1|xroads-oamt2 | vis-diameter-portal-test-alt.syniverse.com |
| 7.xx DEV | vapp013-hw1 | vis-diameter-portal-dev.syniverse.com |
| 7.xx TEST | hw-xroads-t3|hw-xroads-t4|hw-xroads-t5|hw-xroads-t6 | vis-diameter-portal-test.syniverse.com |
| production | \* | vis-diameter.syniverse.com |
|  |  |  |

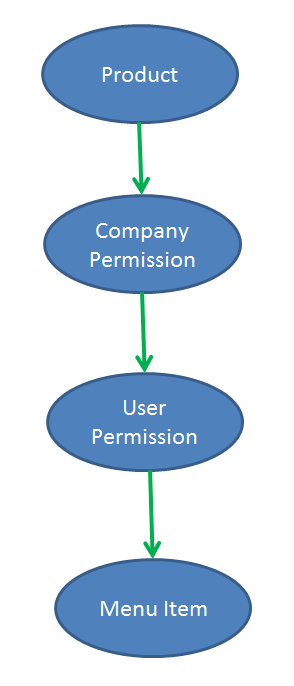
### How Company permission, user permission and menu items are organized

Relationship likes this

a). Company permission should be tied to product

b). User permission should be tied to company permission

c). Menu item should be tied to user permission



**Company permission**

VisPro\_Diameter\_cmpyP

1. It’s tied to a product called VisPro
2. It’s under a permission group called VisProactive\_Co

**User permission**

VisPro\_Diameter\_userP is a permission group

VisPro\_Diameter\_Alarm\_Pod\_View\_userP and VisPro\_Diameter\_Heat\_Map\_Pod\_View\_userP are

1. menu items
2. under permission group VisPro\_Diameter\_userP
3. tied to Company permission VisPro\_Diameter\_cmpyP
4. Belongs to product VisSvc

**Menu item**

VisPro\_Diameter\_Menu is a presentation group, and it is under presentation group VisSvc

VisPro\_Diameter\_Alarm\_Pod\_menuitem and VisPro\_Diameter\_Heat\_Map\_Pod\_menuitem are menu items, they are assigned user permission by database table column PRESENTATION.PRSNTTN\_PERMISSIONID, two user permissions VisPro\_Diameter\_Alarm\_Pod\_View\_userP and VisPro\_Diameter\_Heat\_Map\_Pod\_View\_userP are assigned to these two menu items respectively

In database table RSRC\_TO\_RSRC\_REL we store relationship between rsrc\_id1 and rsrc\_id2, and rsrc\_id2, stress the following points

1. rsrc\_typ\_cd binds to rsrc\_id2
2. rsrc\_id2 is the parent id
3. Most time, the values stored in both rsrc\_id1 and rsrc\_id2 are from database table RESOURC.RSRC\_ID, but it’s not this case if rel\_typ\_cd is” PresGroup”, at this time rsrc\_id1 is still from database table RESOURC.RSRC\_ID, however, rsrc\_id2 is from PRESENTATION.PRSNTTN\_ID