

WDNetView v1.0

Visualiseur de contrôle .Net dans une application Windev

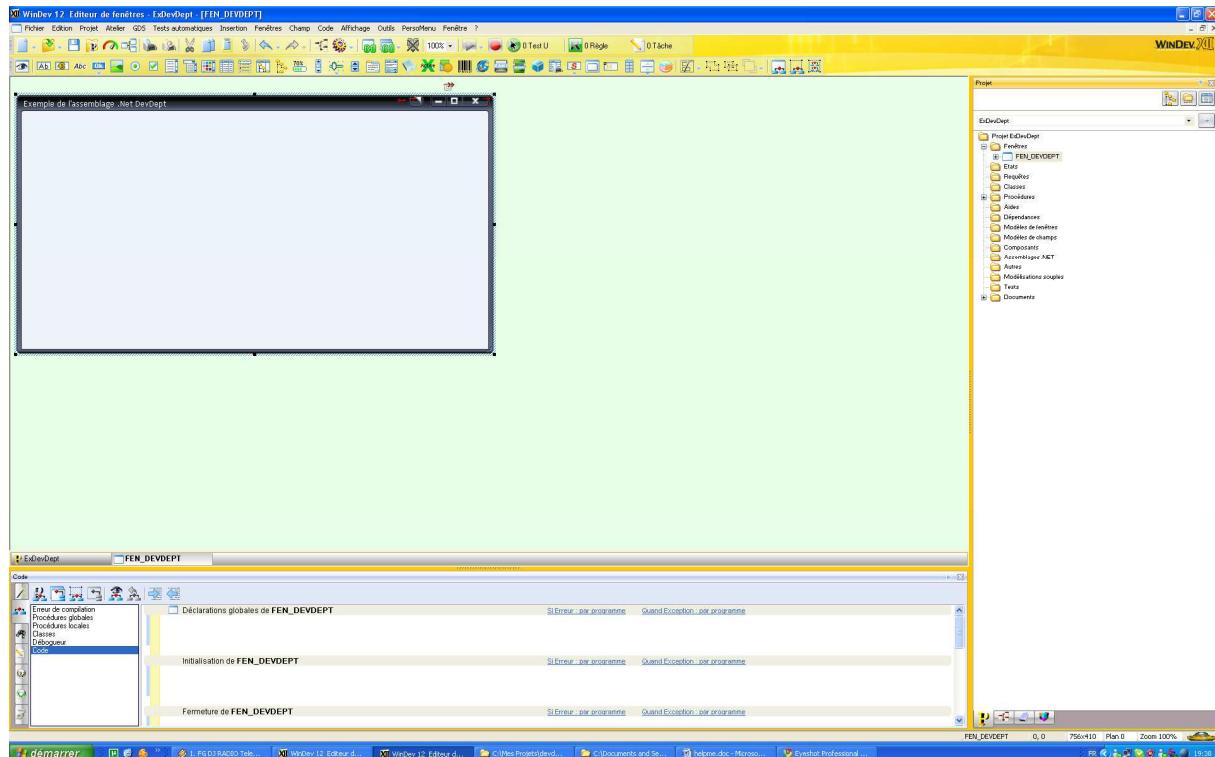
But de ce document : Montrez à l'utilisateur, via un exemple d'assemblage .Net, comment visualiser ce dernier dans une application Windev.

A/ Pré-Requis

- Pour cet exemple, j'ai choisi le composant .Net DevDept. Il s'agit d'un assemblage .Net de visualisation et de création d'objets 3D (entre autre). Vous devez donc le télécharger à cette adresse http://www.devdept.com/eyeshot/free_trial_download.php?edition=pro et l'avoir installé.
- Vous devez également créer un projet de base Windev. Je ne détaillerais pas ce point.

B/ Ajout des objets graphiques dans Windev

Tout d'abord, ajouter une fenêtre basique dans votre projet Windev :

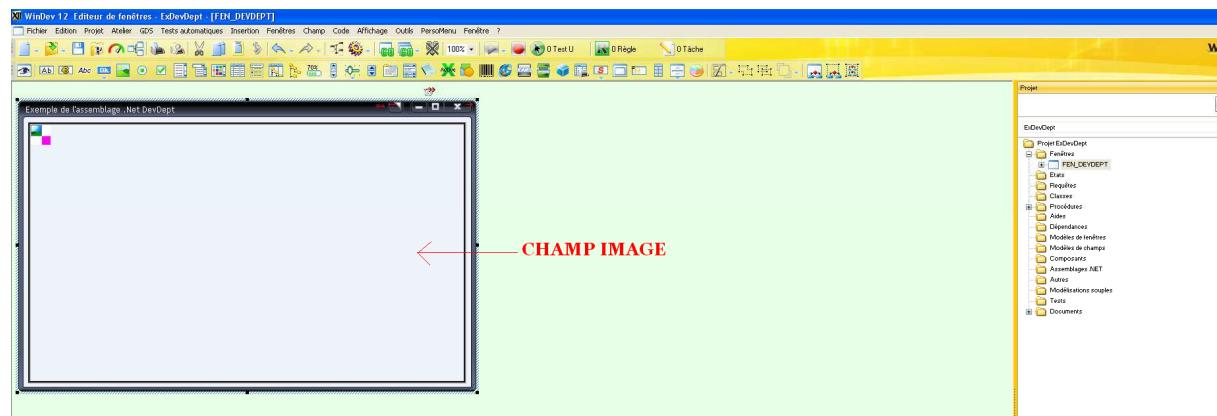


Ce document est la propriété de Vincent Roy.

Toute reproduction même partielle est interdite et ne peut se faire sans son approbation préalable.

A cette fenêtre, ajouter un champ image :

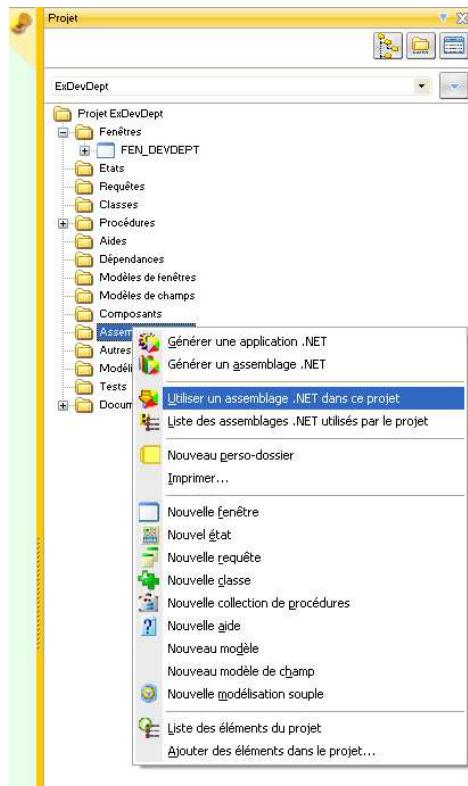
NB : C'est dans ce champ image que vous allez pouvoir visualiser le composant .Net.



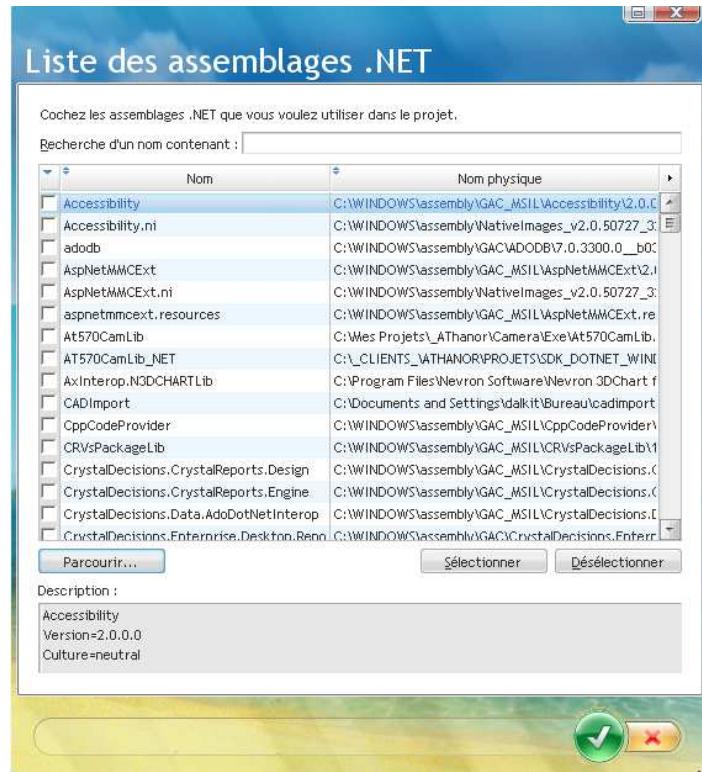
C/ Ajout de l'assemblage .Net

La partie graphique étant terminée, il faut maintenant ajouter dans votre projet l'assemblage .Net DevDept. Il s'agit de la DLL « *devDept.EyeshotNurbs.dll* » que vous trouverez dans le dossier d'installation (par défaut dans C:\Program Files\devDept Software\Eyeshot Nurbs 3.0 Eval\Bin).

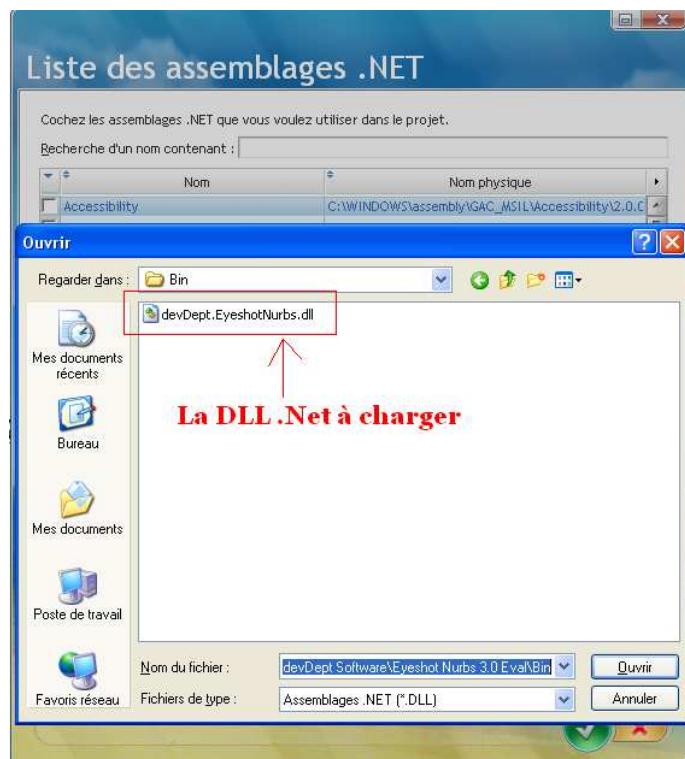
Pour ce faire, dans la fenêtre du projet, ajouter l'assemblage comme ceci :



Une fenêtre avec les assemblages disponibles sur votre machine s'affiche :



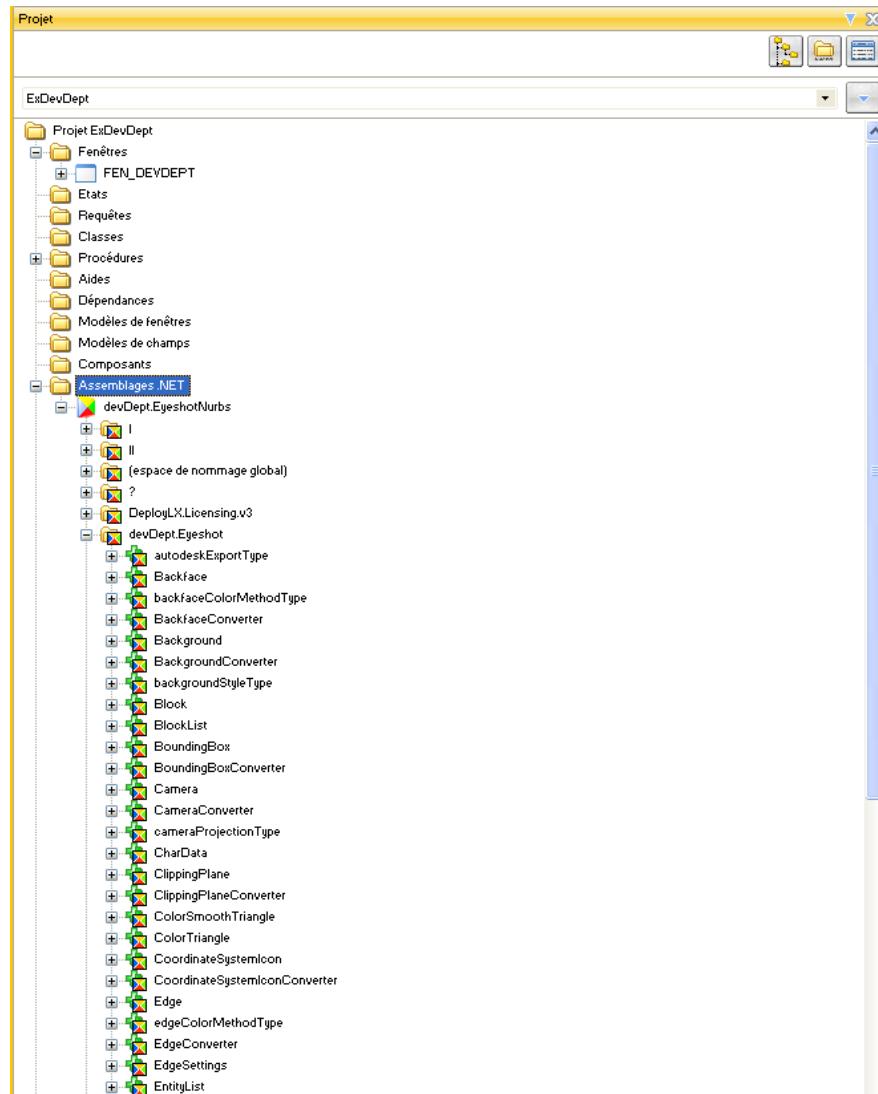
Cliquez sur le bouton Parcourir pour choisir la DLL DevDept citée plus haut et valider :



Ce document est la propriété de Vincent Roy.

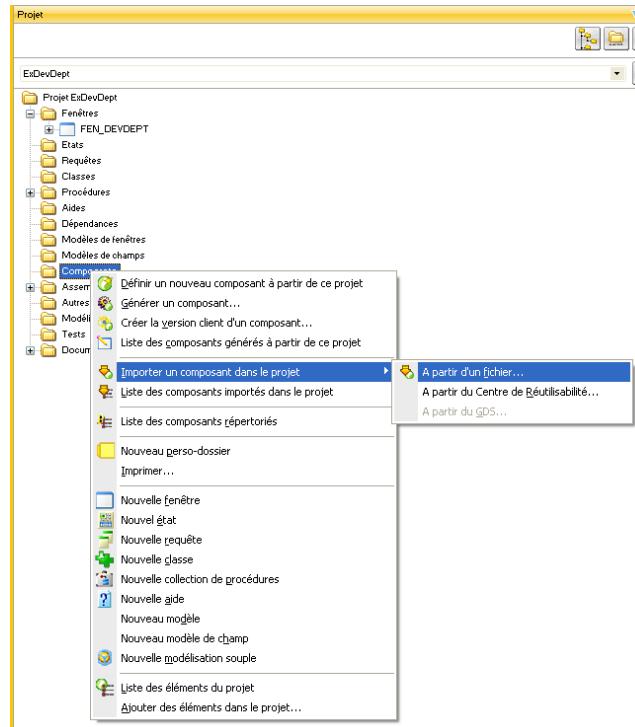
Toute reproduction même partielle est interdite et ne peut se faire sans son approbation préalable.

A ce stade, vous devriez voir les classes et membres de l'assemblage :

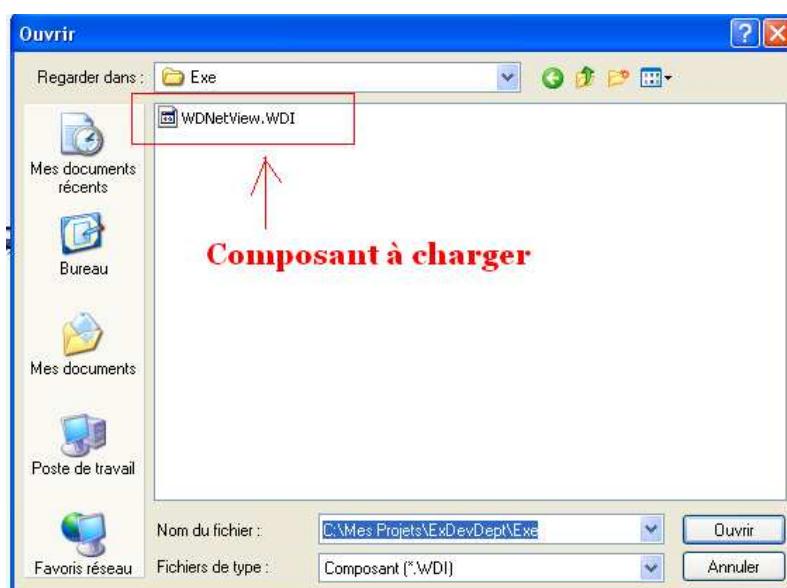


D/ Ajout du composant WDNetView

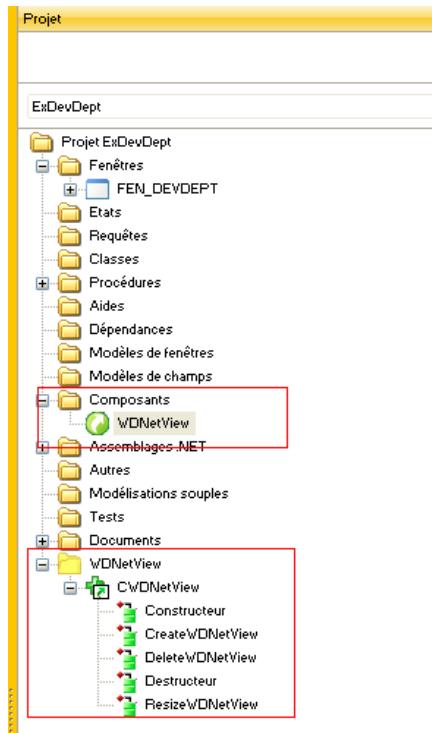
C'est exactement la même démarche que pour l'assemblage .Net, sauf que c'est un composant. Sur l'explorateur de projet, cliquez sur « *composant* », puis « *importer un composant dans le projet* » et enfin « *à partir d'un fichier* » :



Une fenêtre de sélection s'affiche. Vous devez choisir le fichier WDNetView.WDI :



A ce stade, vous devez voir la classe de WDNetView dans Windev :



E/ Ajout du code de visualisation

La commence les choses intéressantes ☺

Le composant WDNetView ne se compose que d'une seule classe composée elle même de trois méthodes :

- CreateWDNetView : méthode de création de l'interface de visualisation entre Windev et l'assemblage .Net
- DeleteWDNetView : méthode d'arrêt de l'interface (la visualisation est stoppée)
- ResizeWDNetView : méthode de gestion du redimensionnement du contrôle .Net

Détail des paramètres :

- CreateWDNetView se compose de deux paramètres
 - o Le 1^{er} est l'objet du contrôle .Net à visualiser
 - o Le 2^{ème} est le nom de l'image windev de destination (sous format chaîne)
- DeleteWDNetView : Aucun paramètre.
- ResizeWDNetView : Aucun paramètre.

Commençons dès à présent à écrire le code nécessaire.

Ce document est la propriété de Vincent Roy.

Toute reproduction même partielle est interdite et ne peut se faire sans son approbation préalable.

Les étapes sont très simples :

- Il faut tout d'abord créer un objet correspondant au nom de votre contrôle .Net (Pour connaître le nom du contrôle .Net, lisez la documentation du constructeur. La plupart du temps, il est livré des exemples en VB.Net, C# ...)

Exemple :

```
// Création de l'objet du contrôle .Net
ObjDevDept est un ViewportProfessional
```

Et en image :



NB : Vous le déclarez où vous voulez. Je l'ai mis dans la partie « déclarations globales de la fenêtre ».

- Ensuite vient la création de l'objet du composant WDNetView

Exemple :

```
// Création de l'objet du composant WDNetView
objWDNetView est un CWDNetView ()
```

Et en image :



Ce document est la propriété de Vincent Roy.

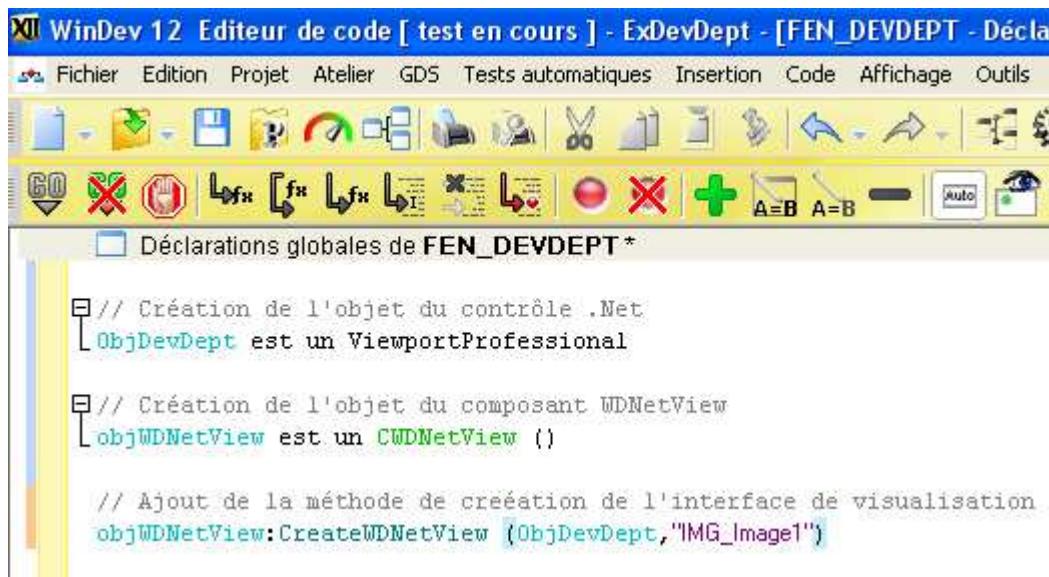
Toute reproduction même partielle est interdite et ne peut se faire sans son approbation préalable.

- Ensuite, ajouter la méthode de création de l'interface de visualisation :

Exemple :

```
// Ajout de la méthode de création de l'interface de visualisation
objWDNetView:CreateWDNetView (ObjDevDept,"IMG_Image1")
```

Et en image:



- Ensuite, ajouter le code de gestion du redimensionnement :

Exemple :

```
// Méthode de gestion du redimensionnement du contrôle .Net
objWDNetView:ResizeWDNetView()
```

Et en image :



- Enfin, ajouter la méthode d'arrêt de l'interface de visualisation dans la fermeture de la fenêtre :

Exemple :

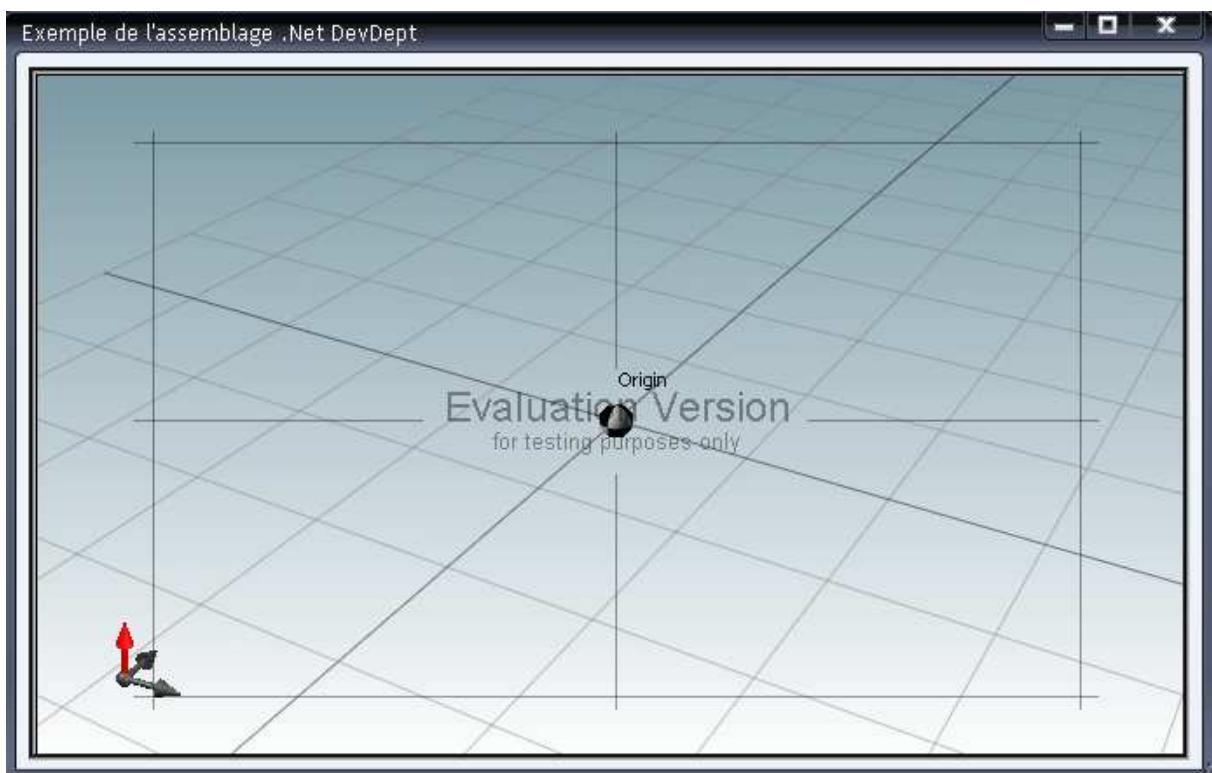
```
// Arret de l'interface de visualisation  
objWDNetView:DeleteWDNetView ()
```

Et en image :



F/ Aperçu du résultat

Lancez le projet et vous obtenez un beau contrôle .Net dans une fenêtre Windev ☺.



Enfin, il ne vous reste plus qu'à coder à votre guise pour afficher des objets dans le contrôle .Net ☺

```

clP2D est un Point2D dynamique
clP2D = allouer un Point2D (200,200)

gpclDevDept:Grid:Max = clP2D
gpclDevDept:Grid:step = 25

nArrows est un entier = 13;
nArcRadius est un entier = 100;

rArcSpan est un réel = 120;

clMyArc est un Arc (Point3D::Origin,nArcRadius,U
gpclDevDept:Entities:Add (clMyArc,0)

i est un entier=0

POUR i=1 _A_ nArrows
  rRadAngle est un réel = Utility::DegToRad(i*A
  m est un Mesh (meshNatureType::Smooth)

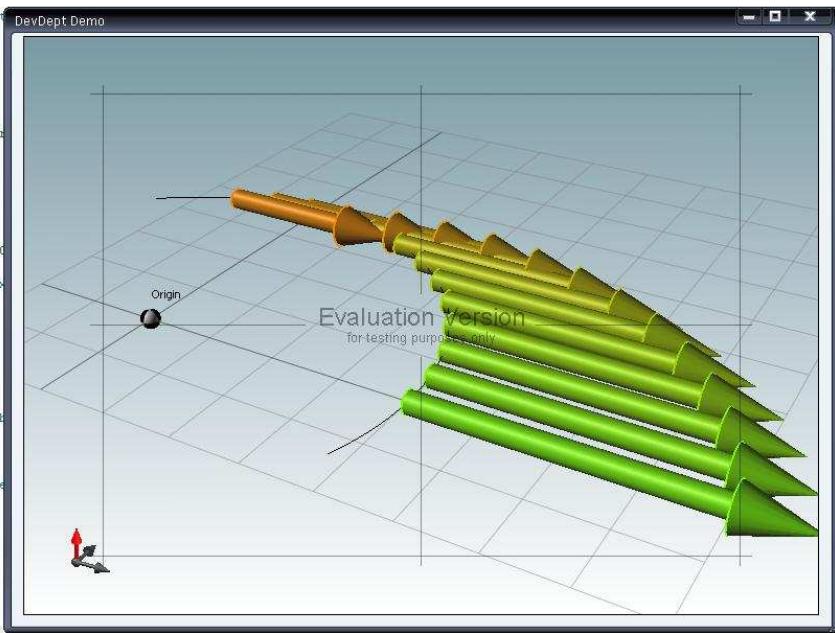
  m:EdgeStyle = meshEdgeStyleType::Sharp;

  m:MakeArrow(4, 100-(i-1)*4, 8, 24, 16);

  clTra est un Translation(nArcRadius * Math::Pi
  clRot est un Rotation(rRadAngle, Vector3D::A
  pclCombined est un Transformation dynamique
  //Combined = cltra * clrot
  m:TransformBy(clTra);
  gpclDevDept:Entities:Add (m,0,Color::FromArg
  FIN

  gpclDevDept:DisplayMode = viewportDisplayType::Re
  // sets trimetric view
  gpclDevDept:SetView(viewType::Trimetric);
  // fits the model in the viewport
  gpclDevDept:ZoomFit();

```



Bon dev.

Vincent Roy.

Pour toutes demandes d'informations: contact@ennovsys.fr