

## Linking Interactors

Contributed by Mark Wyszomierski  
Monday, 27 August 2007

Download article attachment: DevSample\_LinkInteractors\_MDI.zip(83.9kb)

This article shows how to link interactors so it appears that camera changes in one render window are copied to all other 'linked' render windows. The sample application is created as a win32 MDI application but the actual supporting classes are cross platform.

The solution presented in this sample application is encapsulated in one observer class called VtkObserverLinkInteractors. This observer class listens for interaction events invoked by the renderer's render window interactor style. Any type of interaction through mouse or keyboard should be acted upon.

The VtkObserverLinkInteractors class uses two static member variables to achieve linking. Here they are in the source code:

Static member variables of VtkObserverLinkInteractors

```
// Static list shared between all instances. When the static flag is set to true,
```

```
// we can iterate over all instances in this list and copy interactor actions
```

```
// between everyone.
```

```
list<vtkRenderWindowInteractor*> VtkObserverLinkInteractors::StaticInstances;
```

```
// The static flag to set when linking should be performed.
```

```
bool VtkObserverLinkInteractors::StaticLinkInteractors = false;
```

The list of static instances allows any instance of VtkObserverLinkInteractors to keep track of all other instantiated observers. When an interaction event is observed, we can simply have the 'control' observer iterate over the list and copy the camera parameters to all other interactors.

The static bool variable StaticLinkInteractors is the flag to set when you want to turn linking on or off. In the sample MDI application, I linked it to the main menu.

That's really all there is to it. In the sample application you can create a few different render windows (ctrl+n) and you will see a world axes in the center of each view. From the main menu you can go to the 'Test' menu and choose 'Link Interactors'. Now try interacting with the scene with the mouse or keyboard and the same camera actions should be copied between all opened render windows.

The VtkObserverLinkInteractors class can be added to any scene regardless of its contents and linking should work (ie. it does not matter if you're rendering volumes vs. a world axes like in the sample). It is very generic.

I have seen on the VTK mailing list a possible shortcut to copying over the camera parameters between interactors. In the suggestion, a user had been doing:

Possible shortcut

```
pDstRenderer->SetActiveCamera(pSrcRenderer->GetActiveCamera());
```

I gave this a try and it seemed to work - the camera positions were copied for all linked interactors. However after a few minutes my machine would completely crash into the windows blue screen of no return. I'm pretty sure the above call was the cause, ever since changing to copying over camera properties 'manually' as in the sample application, no such

crashes have occurred.

Download the sample application and have a look at the source code.