

```
vector<Body*> sectionCopyTool::getAllVisibleBodies()
{
    vector<Body *> outputBodies;
    outputBodies.clear();
    //all visible bodies in the display part

    vector<DisplayableObject *> visibleObjects = theSession->Parts()->Display()->Views()->WorkView()->AskVisibleObjects();
    if (visibleObjects.empty())
    {
        sectionCopyTool::theUI->NXMessageBox()->Show("Block Styler", NXOpen::NXMessageBox::DialogTypeError, "Copy Bodies Failed");
    }
    else
    {
        for (int i = 0 ; i<visibleObjects.size();i++)
        {
            Body *body = dynamic_cast<Body*>(visibleObjects.at(i));
            if (body!=NULL && !body->IsBlanked())
            {
                //load the component fully
                if (body->IsOccurrence())
                {
                    bool option1;
                    option1 = theSession->Parts()->LoadOptions()->UsePartialLoading();
                    theSession->Parts()->LoadOptions()->SetUsePartialLoading(false);
                    std::vector<NXOpen::Assemblies::Component *> componentsToOpen1(1);
                    componentsToOpen1[0] = body->OwningComponent();
                    std::vector<NXOpen::Assemblies::ComponentAssembly::OpenComponentStatus> openStatus1;
                    NXOpen::PartLoadStatus *partLoadStatus1;
                    partLoadStatus1 = theSession->Parts()->Display()->ComponentAssembly()->OpenComponents(NXOpen::Assemblies::ComponentAssembly::OpenOptionComponentOnly,
                    componentsToOpen1, openStatus1);
                    theSession->Parts()->LoadOptions()->SetUsePartialLoading(true);
                    delete partLoadStatus1;
                }
                outputBodies.push_back(body);
            }
        }
    }

    return outputBodies;
}
```