



```
//For getting the CT Preset data i.e CT_BONE, CT_ABDOMEN, CT_PULMONARY,  
//CT_BRAIN etc with the user defines WindowCenter & WindowWidth.
```

```
m_iMSTDiCoDec->getPresetData((unsigned char*)&pchImgBuffer, sFrameNum,  
                             nKey,& nWindowWidth, &nWindowCenter);
```

```
//For getting file paths of all child of a node with key = nKey.
```

```
typedef vector<string> FILELIST;  
typedef FILELIST::iterator FILE_ITERATOR;  
FILELIST* m_vFilelist;
```

```
LPSAFEARRAY* psaTreeStruct = new SAFEARRAY;  
m_iMSTDiCoDec->GetFilePath(nKey, psaTreeStruct);  
m_vFileList = (FILELIST*)psaTreeStruct->pvData;
```

### 3D Generation

```
//for filling the 3D information in the below structure for the series with key = //nSeriesKey
```

```
struct INFO_3D  
{  
    int nHeight;  
    int nWidth;  
    int nNumSlices;  
    float nPixelSpacing;  
    float nSliceSpacing;  
    float nZSpacing;  
    string sRAHCoordinate;  
};
```

```
IThreeDim* m_VitualEye;
```

```
INFO_3D * m_3dInfo= new INFO_3D;  
LPSAFEARRAY lpSafeArr = new SAFEARRAY;  
m_iMSTDiCoDec->Stack3D(&lpSafeArr,nSeriesKey);  
m_3dInfo = (*lpSafeArr)->pvdata;
```

```
//for allocating memory to the volume data according to Info_3d
```

```
unsigned char* VolData=NULL;  
m_VitualEye->CreateVolume(&Info_3d,&VolData);
```

```
//// for getting the volume data
```

```
m_iMSTDiCoDec->getVolumeData(unsigned char*& uchVolumeData);
```

```

// For creating render window.
m_VitualEye->CreateRenderWindow(int* InstanceHandle,HWND WindowHandle);

//routing window procedures to virtual eye
m_VitualEye->RenderWindowProc(HWND WindowHandle, UINT message, WPARAM
                               wParam, LPARAM lParam,int nCropOn);

//Render the 3D Operation
enum Render_Operation
{
    Volume_Rendering = 1,
    MIP,
    Orthogonal_2D,
    MPR_2D,
    MPR_3D,
    Surface_Rendering
};
m_VitualEye->RenderOperation(int* InstanceHandle,HWND WindowHandle,
                             int OpIndex, unsigned char* pLUT,
                             int LutEntries,
                             unsigned char* pOpacityLUT,
                             int OpacityEntries);

//Deleting Render window
m_VitualEye->DeleteRenderWindow(int* InstanceHandle,HWND WindowHandle);

//Deleting volume
m_VitualEye->DeleteVolume();

```