|  |
| --- |
|  |

|  |  |
| --- | --- |
| **[http://www.functionx.com/design/logo1.gif](http://www.functionx.com/visualc/index.htm)** | **Windows Explorer Application** |
|  | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | http://www.functionx.com/visualc/applications/images/champions2.gif |  |  | | --- | | **Introduction** | |  |  |  | | --- | | A tree explorer is the type of application that displays a tree control on one side and another type of control on another side of a main window. Many applications nowadays use this scenario. Windows Explorer, although not the only one, is the most popular of them:  http://www.functionx.com/visualc/applications/images/explorer1.gif  In Windows Explorer, a tree control displays on the left side and a list control on the right side. In some other applications, the tree may be positioned to the top, the right, the bottom, or a corner of the parent window.  When the user clicks an item in the tree, another view changes to display items that are related to the tree. There are no preset rules as to what to display or what kind of item to display. Everything depends on the person or the company that created the application. |  |  | | --- | | **The Tree Explorer in MFC Applications** | |  |  |  | | --- | | Because of the complexities of the document/view architecture, the MFC probably doesn't provide the least difficult approach to creating a tree explorer. In an MFC application, you usually have to involve each view around the document class in order to get the necessary functionality. For example, if you plan to create an object like Windows Explorer, you may want to use a **CTreeView** object on the left side and a **CListView** on the right side. To implement their functionality, that is, to get information to flow from one view to another, you would consider a **CDocument** variable as the central object. | | |  | | --- | |  | |

|  |
| --- |
| **Practical Learning: Starting the Exercise** |
|  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. Start Microsoft Visual C++ 2. Create a new project using MFC AppWizard or as an MFC Application 3. Name it **Championships** 4. Create the project as a **Single Document** 5. Click the Windows Explorer radio button   http://www.functionx.com/visualc/applications/dlgboxes/mfcappwiz1.gif 6. Change the file names in the dialog box as follows:  |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Class Name | Header File | Implementation File | Base Class | | CChampionshipsView | **CRightView** | **RightView**.h | **RightView**.cpp | CListView | | CChampionshipsApp | **CExerciseApp** | *No Change* | *No Change* | *No Change* | | CChampionshipsDoc | **CExerciseDoc** | **ExerciseDoc**.h | **ExerciseDoc**.cpp | *No Change* | | CLeftView | *No Change* | *No Change* | *No Change* | *No Change* |  1. Click Finish   http://www.functionx.com/visualc/applications/windows/workspace1.gif 2. In the Class View, double-click CLeftView and notice that is has a method called **GetDocument()** that returns a CExerciseDoc pointer 3. In the Class View, double-click CRightView and notice that is has a method called **GetDocument()** that returns a CExerciseDoc pointer This means that the document class will plan as the intermediary between the views. Now you need to let the document class know about the view classes 4. In Class View, double-click CExerciseDoc to access its header file 5. Declare a pointer to each view class as follows:  |  | | --- | | // ExerciseDoc.h : interface of the CExerciseDoc class  //  /////////////////////////////////////////////////////////////////////////////  #pragma once  class CLeftView;  class CRightView;  class CExerciseDoc : public CDocument  {  protected: // create from serialization only  CExerciseDoc();  DECLARE\_DYNCREATE(CExerciseDoc)  // Attributes  public:  // Operations  public:  CLeftView \*pLeftView;  CRightView \*pRightView;  // Overrides  // ClassWizard generated virtual function overrides  //{{AFX\_VIRTUAL(CExerciseDoc)  public:  virtual BOOL OnNewDocument();  virtual void Serialize(CArchive& ar);  //}}AFX\_VIRTUAL  // Implementation  public:  virtual ~CExerciseDoc();  #ifdef \_DEBUG  virtual void AssertValid() const;  virtual void Dump(CDumpContext& dc) const;  #endif  protected:  // Generated message map functions  protected:  //{{AFX\_MSG(CExerciseDoc)  //}}AFX\_MSG  DECLARE\_MESSAGE\_MAP()  }; |  1. Using the ClassWizard dialog box or the Properties window, generate the **WM\_CREATE** message of the CLeftView class   http://www.functionx.com/visualc/applications/dlgboxes/mfcclasswiz3.gif 2. Whenever a view goes through the document class to access the left view, you need to communicate to the document view what object is being called. To indicate this, implement the OnCreate() event as follows:  |  | | --- | | int CLeftView::OnCreate(LPCREATESTRUCT lpCreateStruct)  {  if (CTreeView::OnCreate(lpCreateStruct) == -1)  return -1;    // TODO: Add your specialized creation code here  GetDocument()->pLeftView = this;  return 0;  } |  1. Using the ClassWizard dialog box or the Properties window, generate the **WM\_CREATE** message of the CLeftView class 2. Whenever a view goes through the document class to access the right view, you need to communicate to the document view what object is being called. To indicate this, implement the **OnCreate()** event as follows:  |  | | --- | | int CRightView::OnCreate(LPCREATESTRUCT lpCreateStruct)  {  if (CListView::OnCreate(lpCreateStruct) == -1)  return -1;    // TODO: Add your specialized creation code here  GetDocument()->pRightView = this;  return 0;  } |  1. On the main menu, click Insert -> Resource... 2. In the Insert Resource dialog box, double-click Bitmap 3. Change the ID of the bitmap to IDB\_TREE\_BMP 4. Set its Width to 96 5. Set its Height to 16 6. Design the bitmap as follows:   http://www.functionx.com/visualc/applications/windows/bitmap1.gif 7. In the header file of the CLeftView class, declare a CImageList variable named:  |  | | --- | | #pragma once  class CExerciseDoc;  class CLeftView : public CTreeView  {  protected: // create from serialization only  CLeftView();  DECLARE\_DYNCREATE(CLeftView)  // Attributes  public:  CExerciseDoc\* GetDocument();  // Operations  public:  CImageList m\_TreeImages;  // Overrides  // ClassWizard generated virtual function overrides  //{{AFX\_VIRTUAL(CLeftView)  public:  virtual void OnDraw(CDC\* pDC); // overridden to draw this view  virtual BOOL PreCreateWindow(CREATESTRUCT& cs);  protected:  virtual BOOL OnPreparePrinting(CPrintInfo\* pInfo);  virtual void OnBeginPrinting(CDC\* pDC, CPrintInfo\* pInfo);  virtual void OnEndPrinting(CDC\* pDC, CPrintInfo\* pInfo);  virtual void OnInitialUpdate(); // called first time after construct  //}}AFX\_VIRTUAL  // Implementation  public:  virtual ~CLeftView();  #ifdef \_DEBUG  virtual void AssertValid() const;  virtual void Dump(CDumpContext& dc) const;  #endif  protected:  // Generated message map functions  protected:  //{{AFX\_MSG(CLeftView)  afx\_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);  //}}AFX\_MSG  DECLARE\_MESSAGE\_MAP()  };  #ifndef \_DEBUG // debug version in LeftView.cpp  inline CExerciseDoc\* CLeftView::GetDocument()  { return (CExerciseDoc\*)m\_pDocument; }  #endif |  1. To create some items for the tree, in Class View, right-click CLeftView -> Add Member Function... 2. Set the return type to void 3. Set the Function Name to CreateLeagues   http://www.functionx.com/visualc/applications/dlgboxes/addmbrfn1.gif 4. Press Enter and implement the method as follows:  |  | | --- | | void CLeftView::CreateLeagues()  {  // If there is anything in the tree, remove it  GetTreeCtrl().DeleteAllItems();  CTreeCtrl &ctlLeagues = this->GetTreeCtrl();  m\_TreeImages.Create(IDB\_TREE\_BMP, 16, 1, RGB(255, 255, 255));  ctlLeagues.SetImageList(&m\_TreeImages, TVSIL\_NORMAL);  HTREEITEM hRoot;  HTREEITEM hSeason;  hRoot = ctlLeagues.InsertItem(\_T("World Football Championships"), 0, 1);  hSeason = ctlLeagues.InsertItem("English Premier League", 2, 3, hRoot);  ctlLeagues.InsertItem(\_T("2004-2005"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2003-2004"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2002-2001"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2000-2001"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("1999-2000"), 4, 5, hSeason);  hSeason = ctlLeagues.InsertItem("Lega Nazionale Professionisti", 2, 3, hRoot);  ctlLeagues.InsertItem(\_T("2004-2005"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2003-2004"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2002-2001"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2000-2001"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("1999-2000"), 4, 5, hSeason);    hSeason = ctlLeagues.InsertItem("Ligue Nationale", 2, 3, hRoot);  ctlLeagues.InsertItem(\_T("2004-2005"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2003-2004"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2002-2001"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2000-2001"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("1999-2000"), 4, 5, hSeason);  hSeason = ctlLeagues.InsertItem("Major League Soccer", 2, 3, hRoot);  ctlLeagues.InsertItem(\_T("2004-2005"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2003-2004"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2002-2001"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2000-2001"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("1999-2000"), 4, 5, hSeason);  hSeason = ctlLeagues.InsertItem("Primera Liga", 2, 3, hRoot);  ctlLeagues.InsertItem(\_T("2004-2005"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2003-2004"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2002-2001"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2000-2001"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("1999-2000"), 4, 5, hSeason);  ctlLeagues.Expand(hRoot, TVE\_EXPAND);  } |  1. In the Class View, under the CExerciseDoc node, double-click OnNewDocument 2. To display the tree when the application comes up, call the above method the pointer to the left view class:  |  | | --- | | BOOL CExerciseDoc::OnNewDocument()  {  if (!CDocument::OnNewDocument())  return FALSE;  this->pLeftView->CreateLeagues();  return TRUE;  } |  1. In the top section of the file, add a #include "LeftView.h" line:  |  | | --- | | // ExerciseDoc.cpp : implementation of the CExerciseDoc class  //  #include "stdafx.h"  #include "Championships.h"  #include "ExerciseDoc.h"  #include "LeftView.h" |  1. To add the options to the tree list, access the OnInitialUpdate() event and type the following line that sets the style:  |  | | --- | | void CLeftView::OnInitialUpdate()  {  GetTreeCtrl().ModifyStyle(NULL, TVS\_HASBUTTONS | TVS\_HASLINES | TVS\_LINESATROOT);  CTreeView::OnInitialUpdate();  } |  1. To give more room to the list tree, access the OnCreateClient and change the width of the left frame as follows:  |  | | --- | | BOOL CMainFrame::OnCreateClient(LPCREATESTRUCT /\*lpcs\*/,  CCreateContext\* pContext)  {  // create splitter window  if (!m\_wndSplitter.CreateStatic(this, 1, 2))  return FALSE;  if (!m\_wndSplitter.CreateView(0, 0, RUNTIME\_CLASS(CLeftView), CSize(200, 100), pContext) ||  !m\_wndSplitter.CreateView(0, 1, RUNTIME\_CLASS(CRightView), CSize(100, 100), pContext))  {  m\_wndSplitter.DestroyWindow();  return FALSE;  }  return TRUE;  }  BOOL CMainFrame::PreCreateWindow(CREATESTRUCT& cs)  {  if( !CFrameWnd::PreCreateWindow(cs) )  return FALSE;  cs.style &= ~FWS\_ADDTOTITLE;  return TRUE;  } |  1. Test the application   http://www.functionx.com/visualc/applications/images/champions1.gif 2. After using it, close it and return to your programming environment 3. When an item is selected in the tree, whether by the user or else, you can specify what should happen in another view, such as the right frame for our example. To make this exercise simpler and clearer, we will call a different function of the right frame when a category of node is selected in the left frame (of course, there are other ways to do this but for now, we will use this simplistic approach). First of all, we need a mechanism to erase the list view every time it must be updated, otherwise, the items to display would just be added to the existing ones. In the Class View, right-click the CRightView class -> Add Member Function... 4. Set the return type to **void** and the function name to **ResetLeagues**. Keep it public and press Enter 5. Implement the method as follows:  |  | | --- | | void CRightView::ResetLeagues()  {  CListCtrl &ctlLView = GetListCtrl();  ctlLView.DeleteAllItems();  while(ctlLView.DeleteColumn(0))  ;  UpdateWindow();  } |  1. When the root node is selected in the left frame, we will display a list of leagues. In the Class View, right-click the CRightView class -> Add Member Function... 2. Set the return type to **void** and the function name to **DisplayLeagues**. Keep it public and press Enter 3. Implement the method as follows:  |  | | --- | | void CRightView::DisplayLeagues()  {  CListCtrl &ctlLeagues = this->GetListCtrl();  ResetLeagues();  ctlLeagues.InsertColumn(0, \_T("Country"), LVCFMT\_LEFT, 80);  ctlLeagues.InsertColumn(1, \_T("League Name"), LVCFMT\_LEFT, 160);  ctlLeagues.InsertColumn(2, \_T("Address"), LVCFMT\_LEFT, 180);  ctlLeagues.InsertColumn(3, \_T("Web Site"), LVCFMT\_LEFT, 200);  int nItem;  nItem = ctlLeagues.InsertItem(0, "England");  ctlLeagues.SetItemText(nItem, 1, "English Premier League");  ctlLeagues.SetItemText(nItem, 2, "Don't Know");  ctlLeagues.SetItemText(nItem, 3, "http://www.premierleague.com");    nItem = ctlLeagues.InsertItem(0, "Italie");  ctlLeagues.SetItemText(nItem, 1, "Lega Nazionale Professionisti");  ctlLeagues.SetItemText(nItem, 2, "Forgot");  ctlLeagues.SetItemText(nItem, 3, "http://www.lega-calcio.it");  nItem = ctlLeagues.InsertItem(0, "Kamerun");  ctlLeagues.SetItemText(nItem, 1, "Ligue Nationale");  ctlLeagues.SetItemText(nItem, 2, "Can't Remember");  ctlLeagues.SetItemText(nItem, 3, "http://www.cameroon.fifa.com");  nItem = ctlLeagues.InsertItem(0, "U.S.A.");  ctlLeagues.SetItemText(nItem, 1, "Major League Soccer");  ctlLeagues.SetItemText(nItem, 2, "Can't Remember");  ctlLeagues.SetItemText(nItem, 3, "http://www.mlsnet.com");  nItem = ctlLeagues.InsertItem(0, "Espagne");  ctlLeagues.SetItemText(nItem, 1, "Primera Liga");  ctlLeagues.SetItemText(nItem, 2, "Where?");  ctlLeagues.SetItemText(nItem, 3, "Need to find out");  } |  1. In the same way, when an item is selected in the list, we will find out what its string is, pass that string to the list view and then update the list view accordingly. In the Class View, right-click the CRightView class -> Add Member Function... 2. Set the return type to **void** and the function name to **DisplayLeague(CString League)**. Keep it public and press Enter 3. Implement the method as follows:  |  | | --- | | void CRightView::DisplayLeague(CString League)  {  CListCtrl &ctlRightView = this->GetListCtrl();  ResetLeagues();  ctlRightView.InsertColumn(0, \_T("Season"), LVCFMT\_LEFT, 80);  ctlRightView.InsertColumn(1, \_T("Rank"), LVCFMT\_LEFT, 50);  ctlRightView.InsertColumn(2, \_T("Team"), LVCFMT\_LEFT, 100);  ctlRightView.InsertColumn(3, \_T("P"), LVCFMT\_RIGHT, 30);  ctlRightView.InsertColumn(4, \_T("W"), LVCFMT\_RIGHT, 30);  ctlRightView.InsertColumn(5, \_T("D"), LVCFMT\_RIGHT, 30);  ctlRightView.InsertColumn(6, \_T("L"), LVCFMT\_RIGHT, 30);  ctlRightView.InsertColumn(7, \_T("F"), LVCFMT\_RIGHT, 30);  ctlRightView.InsertColumn(8, \_T("A"), LVCFMT\_RIGHT, 30);  ctlRightView.InsertColumn(9, \_T("GD"), LVCFMT\_RIGHT, 30);  ctlRightView.InsertColumn(10, \_T("Pts"), LVCFMT\_RIGHT, 30);  int nItem;  if( League == "English Premier League" )  {  nItem = ctlRightView.InsertItem(0, "2004-2005");  ctlRightView.SetItemText(nItem, 1, "6");  ctlRightView.SetItemText(nItem, 2, "Aston Villa");  ctlRightView.SetItemText(nItem, 3, "16");  ctlRightView.SetItemText(nItem, 4, "6");  ctlRightView.SetItemText(nItem, 5, "7");  ctlRightView.SetItemText(nItem, 6, "3");  ctlRightView.SetItemText(nItem, 7, "21");  ctlRightView.SetItemText(nItem, 8, "17");  ctlRightView.SetItemText(nItem, 9, "4");  ctlRightView.SetItemText(nItem, 10, "25");  nItem = ctlRightView.InsertItem(0, "2004-2005");  ctlRightView.SetItemText(nItem, 1, "5");  ctlRightView.SetItemText(nItem, 2, "Middlesbro");  ctlRightView.SetItemText(nItem, 3, "16");  ctlRightView.SetItemText(nItem, 4, "8");  ctlRightView.SetItemText(nItem, 5, "4");  ctlRightView.SetItemText(nItem, 6, "4");  ctlRightView.SetItemText(nItem, 7, "27");  ctlRightView.SetItemText(nItem, 8, "20");  ctlRightView.SetItemText(nItem, 9, "7");  ctlRightView.SetItemText(nItem, 10, "28");  nItem = ctlRightView.InsertItem(0, "2004-2005");  ctlRightView.SetItemText(nItem, 1, "4");  ctlRightView.SetItemText(nItem, 2, "Man Utd");  ctlRightView.SetItemText(nItem, 3, "16");  ctlRightView.SetItemText(nItem, 4, "8");  ctlRightView.SetItemText(nItem, 5, "6");  ctlRightView.SetItemText(nItem, 6, "2");  ctlRightView.SetItemText(nItem, 7, "22");  ctlRightView.SetItemText(nItem, 8, "10");  ctlRightView.SetItemText(nItem, 9, "12");  ctlRightView.SetItemText(nItem, 10, "30");    nItem = ctlRightView.InsertItem(0, "2004-2005");  ctlRightView.SetItemText(nItem, 1, "3");  ctlRightView.SetItemText(nItem, 2, "Everton");  ctlRightView.SetItemText(nItem, 3, "16");  ctlRightView.SetItemText(nItem, 4, "10");  ctlRightView.SetItemText(nItem, 5, "3");  ctlRightView.SetItemText(nItem, 6, "3");  ctlRightView.SetItemText(nItem, 7, "20");  ctlRightView.SetItemText(nItem, 8, "14");  ctlRightView.SetItemText(nItem, 9, "6");  ctlRightView.SetItemText(nItem, 10, "33");    nItem = ctlRightView.InsertItem(0, "2004-2005");  ctlRightView.SetItemText(nItem, 1, "2");  ctlRightView.SetItemText(nItem, 2, "Arsenal");  ctlRightView.SetItemText(nItem, 3, "16");  ctlRightView.SetItemText(nItem, 4, "10");  ctlRightView.SetItemText(nItem, 5, "4");  ctlRightView.SetItemText(nItem, 6, "2");  ctlRightView.SetItemText(nItem, 7, "42");  ctlRightView.SetItemText(nItem, 8, "20");  ctlRightView.SetItemText(nItem, 9, "22");  ctlRightView.SetItemText(nItem, 10, "34");  nItem = ctlRightView.InsertItem(0, "2004-2005");  ctlRightView.SetItemText(nItem, 1, "1");  ctlRightView.SetItemText(nItem, 2, "Chelsea");  ctlRightView.SetItemText(nItem, 3, "16");  ctlRightView.SetItemText(nItem, 4, "12");  ctlRightView.SetItemText(nItem, 5, "3");  ctlRightView.SetItemText(nItem, 6, "1");  ctlRightView.SetItemText(nItem, 7, "31");  ctlRightView.SetItemText(nItem, 8, "6");  ctlRightView.SetItemText(nItem, 9, "25");  ctlRightView.SetItemText(nItem, 10, "39");  nItem = ctlRightView.InsertItem(0, "2003-2004");  ctlRightView.SetItemText(nItem, 1, "10");  ctlRightView.SetItemText(nItem, 2, "Birmingham");  ctlRightView.SetItemText(nItem, 3, "38");  ctlRightView.SetItemText(nItem, 4, "12");  ctlRightView.SetItemText(nItem, 5, "14");  ctlRightView.SetItemText(nItem, 6, "12");  ctlRightView.SetItemText(nItem, 7, "43");  ctlRightView.SetItemText(nItem, 8, "48");  ctlRightView.SetItemText(nItem, 9, "-5");  ctlRightView.SetItemText(nItem, 10, "50");  nItem = ctlRightView.InsertItem(0, "2003-2004");  ctlRightView.SetItemText(nItem, 1, "9");  ctlRightView.SetItemText(nItem, 2, "Fulham");  ctlRightView.SetItemText(nItem, 3, "38");  ctlRightView.SetItemText(nItem, 4, "14");  ctlRightView.SetItemText(nItem, 5, "10");  ctlRightView.SetItemText(nItem, 6, "14");  ctlRightView.SetItemText(nItem, 7, "52");  ctlRightView.SetItemText(nItem, 8, "46");  ctlRightView.SetItemText(nItem, 9, "6");  ctlRightView.SetItemText(nItem, 10, "52");    nItem = ctlRightView.InsertItem(0, "2003-2004");  ctlRightView.SetItemText(nItem, 1, "8");  ctlRightView.SetItemText(nItem, 2, "Bolton");  ctlRightView.SetItemText(nItem, 3, "38");  ctlRightView.SetItemText(nItem, 4, "14");  ctlRightView.SetItemText(nItem, 5, "11");  ctlRightView.SetItemText(nItem, 6, "13");  ctlRightView.SetItemText(nItem, 7, "48");  ctlRightView.SetItemText(nItem, 8, "56");  ctlRightView.SetItemText(nItem, 9, "-8");  ctlRightView.SetItemText(nItem, 10, "53");  nItem = ctlRightView.InsertItem(0, "2003-2004");  ctlRightView.SetItemText(nItem, 1, "7");  ctlRightView.SetItemText(nItem, 2, "Charlton");  ctlRightView.SetItemText(nItem, 3, "38");  ctlRightView.SetItemText(nItem, 4, "14");  ctlRightView.SetItemText(nItem, 5, "11");  ctlRightView.SetItemText(nItem, 6, "13");  ctlRightView.SetItemText(nItem, 7, "51");  ctlRightView.SetItemText(nItem, 8, "51");  ctlRightView.SetItemText(nItem, 9, "0");  ctlRightView.SetItemText(nItem, 10, "53");  nItem = ctlRightView.InsertItem(0, "2003-2004");  ctlRightView.SetItemText(nItem, 1, "6");  ctlRightView.SetItemText(nItem, 2, "Aston Villa");  ctlRightView.SetItemText(nItem, 3, "38");  ctlRightView.SetItemText(nItem, 4, "15");  ctlRightView.SetItemText(nItem, 5, "11");  ctlRightView.SetItemText(nItem, 6, "12");  ctlRightView.SetItemText(nItem, 7, "48");  ctlRightView.SetItemText(nItem, 8, "44");  ctlRightView.SetItemText(nItem, 9, "4");  ctlRightView.SetItemText(nItem, 10, "56");  nItem = ctlRightView.InsertItem(0, "2003-2004");  ctlRightView.SetItemText(nItem, 1, "5");  ctlRightView.SetItemText(nItem, 2, "Newcastle");  ctlRightView.SetItemText(nItem, 3, "38");  ctlRightView.SetItemText(nItem, 4, "13");  ctlRightView.SetItemText(nItem, 5, "17");  ctlRightView.SetItemText(nItem, 6, "8");  ctlRightView.SetItemText(nItem, 7, "52");  ctlRightView.SetItemText(nItem, 8, "40");  ctlRightView.SetItemText(nItem, 9, "12");  ctlRightView.SetItemText(nItem, 10, "56");  nItem = ctlRightView.InsertItem(0, "2003-2004");  ctlRightView.SetItemText(nItem, 1, "4");  ctlRightView.SetItemText(nItem, 2, "Liverpool");  ctlRightView.SetItemText(nItem, 3, "38");  ctlRightView.SetItemText(nItem, 4, "16");  ctlRightView.SetItemText(nItem, 5, "12");  ctlRightView.SetItemText(nItem, 6, "10");  ctlRightView.SetItemText(nItem, 7, "55");  ctlRightView.SetItemText(nItem, 8, "37");  ctlRightView.SetItemText(nItem, 9, "18");  ctlRightView.SetItemText(nItem, 10, "60");  nItem = ctlRightView.InsertItem(0, "2003-2004");  ctlRightView.SetItemText(nItem, 1, "3");  ctlRightView.SetItemText(nItem, 2, "Man Utd");  ctlRightView.SetItemText(nItem, 3, "38");  ctlRightView.SetItemText(nItem, 4, "23");  ctlRightView.SetItemText(nItem, 5, "6");  ctlRightView.SetItemText(nItem, 6, "9");  ctlRightView.SetItemText(nItem, 7, "64");  ctlRightView.SetItemText(nItem, 8, "35");  ctlRightView.SetItemText(nItem, 9, "29");  ctlRightView.SetItemText(nItem, 10, "75");  nItem = ctlRightView.InsertItem(0, "2003-2004");  ctlRightView.SetItemText(nItem, 1, "2");  ctlRightView.SetItemText(nItem, 2, "Chelsea");  ctlRightView.SetItemText(nItem, 3, "38");  ctlRightView.SetItemText(nItem, 4, "24");  ctlRightView.SetItemText(nItem, 5, "7");  ctlRightView.SetItemText(nItem, 6, "7");  ctlRightView.SetItemText(nItem, 7, "67");  ctlRightView.SetItemText(nItem, 8, "30");  ctlRightView.SetItemText(nItem, 9, "37");  ctlRightView.SetItemText(nItem, 10, "79");  nItem = ctlRightView.InsertItem(0, "2003-2004");  ctlRightView.SetItemText(nItem, 1, "1");  ctlRightView.SetItemText(nItem, 2, "Arsenal");  ctlRightView.SetItemText(nItem, 3, "38");  ctlRightView.SetItemText(nItem, 4, "26");  ctlRightView.SetItemText(nItem, 5, "12");  ctlRightView.SetItemText(nItem, 6, "0");  ctlRightView.SetItemText(nItem, 7, "73");  ctlRightView.SetItemText(nItem, 8, "26");  ctlRightView.SetItemText(nItem, 9, "47");  ctlRightView.SetItemText(nItem, 10, "90");  }  else if( League == "Lega Nazionale Professionisti" )  {  nItem = ctlRightView.InsertItem(0, "2004-2005");  ctlRightView.SetItemText(nItem, 1, "2");  ctlRightView.SetItemText(nItem, 2, "Milan");  ctlRightView.SetItemText(nItem, 3, "28");  ctlRightView.SetItemText(nItem, 4, "13");  ctlRightView.SetItemText(nItem, 5, "8");  ctlRightView.SetItemText(nItem, 6, "4");  ctlRightView.SetItemText(nItem, 7, "1");  ctlRightView.SetItemText(nItem, 8, "19");  ctlRightView.SetItemText(nItem, 9, "8");  ctlRightView.SetItemText(nItem, 10, "11");  nItem = ctlRightView.InsertItem(0, "2004-2005");  ctlRightView.SetItemText(nItem, 1, "1");  ctlRightView.SetItemText(nItem, 2, "Juventus");  ctlRightView.SetItemText(nItem, 3, "32");  ctlRightView.SetItemText(nItem, 4, "13");  ctlRightView.SetItemText(nItem, 5, "10");  ctlRightView.SetItemText(nItem, 6, "2");  ctlRightView.SetItemText(nItem, 7, "1");  ctlRightView.SetItemText(nItem, 8, "25");  ctlRightView.SetItemText(nItem, 9, "6");  ctlRightView.SetItemText(nItem, 10, "19");  }  } |  1. In the Class View, right-click the CRightView class -> Add Member Function... 2. Set the return type to **void** and the function name to **DisplaySeason(CString League, CString Season)**. Keep it public and press Enter 3. Implement the method as follows:  |  | | --- | | void CRightView::DisplaySeason(CString League, CString Season)  {  CListCtrl &ctlRightView = this->GetListCtrl();  ResetLeagues();  ctlRightView.InsertColumn(1, \_T("Rank"), LVCFMT\_RIGHT, 50);  ctlRightView.InsertColumn(2, \_T("Team"), LVCFMT\_LEFT, 100);  ctlRightView.InsertColumn(3, \_T("P"), LVCFMT\_RIGHT, 30);  ctlRightView.InsertColumn(4, \_T("W"), LVCFMT\_RIGHT, 30);  ctlRightView.InsertColumn(5, \_T("D"), LVCFMT\_RIGHT, 30);  ctlRightView.InsertColumn(6, \_T("L"), LVCFMT\_RIGHT, 30);  ctlRightView.InsertColumn(7, \_T("F"), LVCFMT\_RIGHT, 30);  ctlRightView.InsertColumn(8, \_T("A"), LVCFMT\_RIGHT, 30);  ctlRightView.InsertColumn(9, \_T("GD"), LVCFMT\_RIGHT, 30);  ctlRightView.InsertColumn(10, \_T("Pts"), LVCFMT\_RIGHT, 30);  if( League == "English Premier League" )  {  if( Season == "2004-2005" )  {  int nItem;    nItem = ctlRightView.InsertItem(0, "4");  ctlRightView.SetItemText(nItem, 1, "Man Utd");  ctlRightView.SetItemText(nItem, 2, "16");  ctlRightView.SetItemText(nItem, 3, "8");  ctlRightView.SetItemText(nItem, 4, "6");  ctlRightView.SetItemText(nItem, 5, "2");  ctlRightView.SetItemText(nItem, 6, "22");  ctlRightView.SetItemText(nItem, 7, "10");  ctlRightView.SetItemText(nItem, 8, "12");  ctlRightView.SetItemText(nItem, 9, "30");    nItem = ctlRightView.InsertItem(0, "3");  ctlRightView.SetItemText(nItem, 1, "Everton");  ctlRightView.SetItemText(nItem, 2, "16");  ctlRightView.SetItemText(nItem, 3, "10");  ctlRightView.SetItemText(nItem, 4, "3");  ctlRightView.SetItemText(nItem, 5, "3");  ctlRightView.SetItemText(nItem, 6, "20");  ctlRightView.SetItemText(nItem, 7, "14");  ctlRightView.SetItemText(nItem, 8, "6");  ctlRightView.SetItemText(nItem, 9, "33");    nItem = ctlRightView.InsertItem(0, "2");  ctlRightView.SetItemText(nItem, 1, "Arsenal");  ctlRightView.SetItemText(nItem, 2, "16");  ctlRightView.SetItemText(nItem, 3, "10");  ctlRightView.SetItemText(nItem, 4, "4");  ctlRightView.SetItemText(nItem, 5, "2");  ctlRightView.SetItemText(nItem, 6, "42");  ctlRightView.SetItemText(nItem, 7, "20");  ctlRightView.SetItemText(nItem, 8, "22");  ctlRightView.SetItemText(nItem, 9, "34");  nItem = ctlRightView.InsertItem(0, "1");  ctlRightView.SetItemText(nItem, 1, "Chelsea");  ctlRightView.SetItemText(nItem, 2, "16");  ctlRightView.SetItemText(nItem, 3, "12");  ctlRightView.SetItemText(nItem, 4, "3");  ctlRightView.SetItemText(nItem, 5, "1");  ctlRightView.SetItemText(nItem, 6, "31");  ctlRightView.SetItemText(nItem, 7, "6");  ctlRightView.SetItemText(nItem, 8, "25");  ctlRightView.SetItemText(nItem, 9, "39");  }  else if( Season == "2003-2004" )  {  int nItem;  nItem = ctlRightView.InsertItem(0, "10");  ctlRightView.SetItemText(nItem, 1, "Birmingham");  ctlRightView.SetItemText(nItem, 2, "38");  ctlRightView.SetItemText(nItem, 3, "12");  ctlRightView.SetItemText(nItem, 4, "14");  ctlRightView.SetItemText(nItem, 5, "12");  ctlRightView.SetItemText(nItem, 6, "43");  ctlRightView.SetItemText(nItem, 7, "48");  ctlRightView.SetItemText(nItem, 8, "-5");  ctlRightView.SetItemText(nItem, 9, "50");  nItem = ctlRightView.InsertItem(0, "9");  ctlRightView.SetItemText(nItem, 1, "Fulham");  ctlRightView.SetItemText(nItem, 2, "38");  ctlRightView.SetItemText(nItem, 3, "14");  ctlRightView.SetItemText(nItem, 4, "10");  ctlRightView.SetItemText(nItem, 5, "14");  ctlRightView.SetItemText(nItem, 6, "52");  ctlRightView.SetItemText(nItem, 7, "46");  ctlRightView.SetItemText(nItem, 8, "6");  ctlRightView.SetItemText(nItem, 9, "52");  nItem = ctlRightView.InsertItem(0, "8");  ctlRightView.SetItemText(nItem, 1, "Bolton");  ctlRightView.SetItemText(nItem, 2, "38");  ctlRightView.SetItemText(nItem, 3, "14");  ctlRightView.SetItemText(nItem, 4, "11");  ctlRightView.SetItemText(nItem, 5, "13");  ctlRightView.SetItemText(nItem, 6, "48");  ctlRightView.SetItemText(nItem, 7, "56");  ctlRightView.SetItemText(nItem, 8, "-8");  ctlRightView.SetItemText(nItem, 9, "53");  nItem = ctlRightView.InsertItem(0, "7");  ctlRightView.SetItemText(nItem, 1, "Charlton");  ctlRightView.SetItemText(nItem, 2, "38");  ctlRightView.SetItemText(nItem, 3, "14");  ctlRightView.SetItemText(nItem, 4, "11");  ctlRightView.SetItemText(nItem, 5, "13");  ctlRightView.SetItemText(nItem, 6, "51");  ctlRightView.SetItemText(nItem, 7, "51");  ctlRightView.SetItemText(nItem, 8, "0");  ctlRightView.SetItemText(nItem, 9, "53");  nItem = ctlRightView.InsertItem(0, "6");  ctlRightView.SetItemText(nItem, 1, "Aston Villa");  ctlRightView.SetItemText(nItem, 2, "38");  ctlRightView.SetItemText(nItem, 3, "15");  ctlRightView.SetItemText(nItem, 4, "11");  ctlRightView.SetItemText(nItem, 5, "12");  ctlRightView.SetItemText(nItem, 6, "48");  ctlRightView.SetItemText(nItem, 7, "44");  ctlRightView.SetItemText(nItem, 8, "4");  ctlRightView.SetItemText(nItem, 9, "56");  nItem = ctlRightView.InsertItem(0, "5");  ctlRightView.SetItemText(nItem, 1, "Newcastle");  ctlRightView.SetItemText(nItem, 2, "38");  ctlRightView.SetItemText(nItem, 3, "13");  ctlRightView.SetItemText(nItem, 4, "17");  ctlRightView.SetItemText(nItem, 5, "8");  ctlRightView.SetItemText(nItem, 6, "52");  ctlRightView.SetItemText(nItem, 7, "40");  ctlRightView.SetItemText(nItem, 8, "12");  ctlRightView.SetItemText(nItem, 9, "66");  nItem = ctlRightView.InsertItem(0, "4");  ctlRightView.SetItemText(nItem, 1, "Liverpool");  ctlRightView.SetItemText(nItem, 2, "38");  ctlRightView.SetItemText(nItem, 3, "16");  ctlRightView.SetItemText(nItem, 4, "12");  ctlRightView.SetItemText(nItem, 5, "10");  ctlRightView.SetItemText(nItem, 6, "55");  ctlRightView.SetItemText(nItem, 7, "37");  ctlRightView.SetItemText(nItem, 8, "18");  ctlRightView.SetItemText(nItem, 9, "60");  nItem = ctlRightView.InsertItem(0, "3");  ctlRightView.SetItemText(nItem, 1, "Man Utd");  ctlRightView.SetItemText(nItem, 2, "38");  ctlRightView.SetItemText(nItem, 3, "23");  ctlRightView.SetItemText(nItem, 4, "6");  ctlRightView.SetItemText(nItem, 5, "9");  ctlRightView.SetItemText(nItem, 6, "64");  ctlRightView.SetItemText(nItem, 7, "35");  ctlRightView.SetItemText(nItem, 8, "29");  ctlRightView.SetItemText(nItem, 9, "75");  nItem = ctlRightView.InsertItem(0, "2");  ctlRightView.SetItemText(nItem, 1, "Chelsea");  ctlRightView.SetItemText(nItem, 2, "38");  ctlRightView.SetItemText(nItem, 3, "24");  ctlRightView.SetItemText(nItem, 4, "7");  ctlRightView.SetItemText(nItem, 5, "7");  ctlRightView.SetItemText(nItem, 6, "67");  ctlRightView.SetItemText(nItem, 7, "30");  ctlRightView.SetItemText(nItem, 8, "37");  ctlRightView.SetItemText(nItem, 9, "79");  nItem = ctlRightView.InsertItem(0, "1");  ctlRightView.SetItemText(nItem, 1, "Arsenal");  ctlRightView.SetItemText(nItem, 2, "38");  ctlRightView.SetItemText(nItem, 3, "26");  ctlRightView.SetItemText(nItem, 4, "12");  ctlRightView.SetItemText(nItem, 5, "0");  ctlRightView.SetItemText(nItem, 6, "73");  ctlRightView.SetItemText(nItem, 7, "26");  ctlRightView.SetItemText(nItem, 8, "47");  ctlRightView.SetItemText(nItem, 9, "90");  }  }  else if( League == "Lega Nazionale Professionisti" )  {  if( Season == "2004-2005" )  {  int nItem;    nItem = ctlRightView.InsertItem(0, "2");  ctlRightView.SetItemText(nItem, 1, "Milan");  ctlRightView.SetItemText(nItem, 2, "28");  ctlRightView.SetItemText(nItem, 3, "13");  ctlRightView.SetItemText(nItem, 4, "8");  ctlRightView.SetItemText(nItem, 5, "4");  ctlRightView.SetItemText(nItem, 6, "1");  ctlRightView.SetItemText(nItem, 7, "19");  ctlRightView.SetItemText(nItem, 8, "8");  ctlRightView.SetItemText(nItem, 9, "11");  nItem = ctlRightView.InsertItem(0, "1");  ctlRightView.SetItemText(nItem, 1, "Juventus");  ctlRightView.SetItemText(nItem, 2, "32");  ctlRightView.SetItemText(nItem, 3, "13");  ctlRightView.SetItemText(nItem, 4, "10");  ctlRightView.SetItemText(nItem, 5, "2");  ctlRightView.SetItemText(nItem, 6, "1");  ctlRightView.SetItemText(nItem, 7, "25");  ctlRightView.SetItemText(nItem, 8, "6");  ctlRightView.SetItemText(nItem, 9, "19");  }  }  } |  1. To make this exercise again simpler, we will mostly care only for the Report View Access the OnInitialUpdate() event of the CRightView class and change it as follows:  |  | | --- | | void CRightView::OnInitialUpdate()  {  CListView::OnInitialUpdate();  CListCtrl &ctlCarParts = this->GetListCtrl();  ctlCarParts.ModifyStyle(NULL, LVS\_REPORT);  ctlCarParts.SetExtendedStyle(LVS\_EX\_FULLROWSELECT | LVS\_EX\_GRIDLINES);  } |  1. Now that we have the necessary functions we can react when a node changes in the tree view. Using either the ClassWizard, the Messages section of the Properties, or whatever, generate a TVN\_SELCHANGED message for the CLeftView class   http://www.functionx.com/visualc/applications/dlgboxes/newmsg1.gif 2. Implement the OnSelchanged event as follows:  |  | | --- | | // LeftView.cpp : implementation of the CLeftView class  //  #include "stdafx.h"  #include "Championships.h"  #include "ExerciseDoc.h"  #include "LeftView.h"  #include "RightView.h"  #ifdef \_DEBUG  #define new DEBUG\_NEW  #undef THIS\_FILE  static char THIS\_FILE[] = \_\_FILE\_\_;  #endif  /////////////////////////////////////////////////////////////////////////////  // CLeftView  IMPLEMENT\_DYNCREATE(CLeftView, CTreeView)  BEGIN\_MESSAGE\_MAP(CLeftView, CTreeView)  //{{AFX\_MSG\_MAP(CLeftView)  ON\_WM\_CREATE()  ON\_NOTIFY\_REFLECT(TVN\_SELCHANGED, OnSelchanged)  //}}AFX\_MSG\_MAP  // Standard printing commands  ON\_COMMAND(ID\_FILE\_PRINT, CTreeView::OnFilePrint)  ON\_COMMAND(ID\_FILE\_PRINT\_DIRECT, CTreeView::OnFilePrint)  ON\_COMMAND(ID\_FILE\_PRINT\_PREVIEW, CTreeView::OnFilePrintPreview)  END\_MESSAGE\_MAP()  /////////////////////////////////////////////////////////////////////////////  // CLeftView construction/destruction  CLeftView::CLeftView()  {  }  CLeftView::~CLeftView()  {  }  BOOL CLeftView::PreCreateWindow(CREATESTRUCT& cs)  {  return CTreeView::PreCreateWindow(cs);  }  /////////////////////////////////////////////////////////////////////////////  // CLeftView drawing  void CLeftView::OnDraw(CDC\* pDC)  {  CExerciseDoc\* pDoc = GetDocument();  ASSERT\_VALID(pDoc);  }  /////////////////////////////////////////////////////////////////////////////  // CLeftView printing  BOOL CLeftView::OnPreparePrinting(CPrintInfo\* pInfo)  {  // default preparation  return DoPreparePrinting(pInfo);  }  void CLeftView::OnBeginPrinting(CDC\* /\*pDC\*/, CPrintInfo\* /\*pInfo\*/)  {  }  void CLeftView::OnEndPrinting(CDC\* /\*pDC\*/, CPrintInfo\* /\*pInfo\*/)  {  }  void CLeftView::OnInitialUpdate()  {  GetTreeCtrl().ModifyStyle(NULL, TVS\_HASBUTTONS | TVS\_HASLINES | TVS\_LINESATROOT);  CTreeView::OnInitialUpdate();  }  /////////////////////////////////////////////////////////////////////////////  // CLeftView diagnostics  #ifdef \_DEBUG  void CLeftView::AssertValid() const  {  CTreeView::AssertValid();  }  void CLeftView::Dump(CDumpContext& dc) const  {  CTreeView::Dump(dc);  }  CExerciseDoc\* CLeftView::GetDocument() // non-debug version is inline  {  ASSERT(m\_pDocument->IsKindOf(RUNTIME\_CLASS(CExerciseDoc)));  return (CExerciseDoc\*)m\_pDocument;  }  #endif //\_DEBUG  /////////////////////////////////////////////////////////////////////////////  // CLeftView message handlers  int CLeftView::OnCreate(LPCREATESTRUCT lpCreateStruct)  {  if (CTreeView::OnCreate(lpCreateStruct) == -1)  return -1;    // TODO: Add your specialized creation code here  GetDocument()->pLeftView = this;  return 0;  }  void CLeftView::CreateLeagues()  {  // If there is anything in the tree, remove it  GetTreeCtrl().DeleteAllItems();  // GetDocument()->pRightView->ResetLeaguesList();  CTreeCtrl &ctlLeagues = this->GetTreeCtrl();  m\_TreeImages.Create(IDB\_TREE\_BMP, 16, 1, RGB(255, 255, 255));  ctlLeagues.SetImageList(&m\_TreeImages, TVSIL\_NORMAL);  HTREEITEM hRoot;  HTREEITEM hSeason;  hRoot = ctlLeagues.InsertItem(\_T("World Football Championships"), 0, 1);  hSeason = ctlLeagues.InsertItem("English Premier League", 2, 3, hRoot);  ctlLeagues.InsertItem(\_T("2004-2005"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2003-2004"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2002-2001"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2000-2001"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("1999-2000"), 4, 5, hSeason);  hSeason = ctlLeagues.InsertItem("Lega Nazionale Professionisti", 2, 3, hRoot);  ctlLeagues.InsertItem(\_T("2004-2005"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2003-2004"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2002-2001"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2000-2001"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("1999-2000"), 4, 5, hSeason);    hSeason = ctlLeagues.InsertItem("Ligue Nationale", 2, 3, hRoot);  ctlLeagues.InsertItem(\_T("2004-2005"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2003-2004"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2002-2001"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2000-2001"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("1999-2000"), 4, 5, hSeason);  hSeason = ctlLeagues.InsertItem("Major League Soccer", 2, 3, hRoot);  ctlLeagues.InsertItem(\_T("2004-2005"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2003-2004"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2002-2001"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2000-2001"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("1999-2000"), 4, 5, hSeason);  hSeason = ctlLeagues.InsertItem("Primera Liga", 2, 3, hRoot);  ctlLeagues.InsertItem(\_T("2004-2005"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2003-2004"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2002-2001"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("2000-2001"), 4, 5, hSeason);  ctlLeagues.InsertItem(\_T("1999-2000"), 4, 5, hSeason);  ctlLeagues.Expand(hRoot, TVE\_EXPAND);  }  void CLeftView::OnSelchanged(NMHDR\* pNMHDR, LRESULT\* pResult)  {  NM\_TREEVIEW\* pNMTreeView = (NM\_TREEVIEW\*)pNMHDR;  // TODO: Add your control notification handler code here    \*pResult = 0;    CExerciseDoc \*pDoc = GetDocument();  // Get a reference to the tree control  CTreeCtrl &ctlLeagues = this->GetTreeCtrl();  // Find out what item is selected in the tree  HTREEITEM nodSelected = ctlLeagues.GetSelectedItem();  // Get the string of the selected node  CString strSelected = ctlLeagues.GetItemText(nodSelected);  // Just in case the selected node as a parent, get that parent  HTREEITEM nodParentLeague = ctlLeagues.GetParentItem(nodSelected);  // Find the string of the parent of the selected node, if it has a parent  CString strParentLeague = ctlLeagues.GetItemText(nodParentLeague);    // Now you can react depending on the selected node  if( strSelected == "2004-2005" )  {  if( strParentLeague == "English Premier League" )  {  pDoc->pRightView->DisplaySeason("English Premier League", "2004-2005" );  }  else if( strParentLeague == "Lega Nazionale Professionisti" )  {  pDoc->pRightView->DisplaySeason("Lega Nazionale Professionisti", "2004-2005");  }  }  else if( strSelected == "2003-2004" )  {  if( strParentLeague == "English Premier League" )  {  pDoc->pRightView->DisplaySeason("English Premier League", "2003-2004" );  }  }  else if( strSelected == "English Premier League" )  pDoc->pRightView->DisplayLeague("English Premier League");  else if( strSelected == "Lega Nazionale Professionisti" )  pDoc->pRightView->DisplayLeague("Lega Nazionale Professionisti");  else  pDoc->pRightView->DisplayLeagues();  } |  1. Test the application |

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
|  | | |
| [**Home**](http://www.functionx.com/visualc/index.htm) | [**Copyright © 2004-2015 FunctionX, Inc.**](http://www.functionx.com/visualc/index.htm) |  |
|  | | |