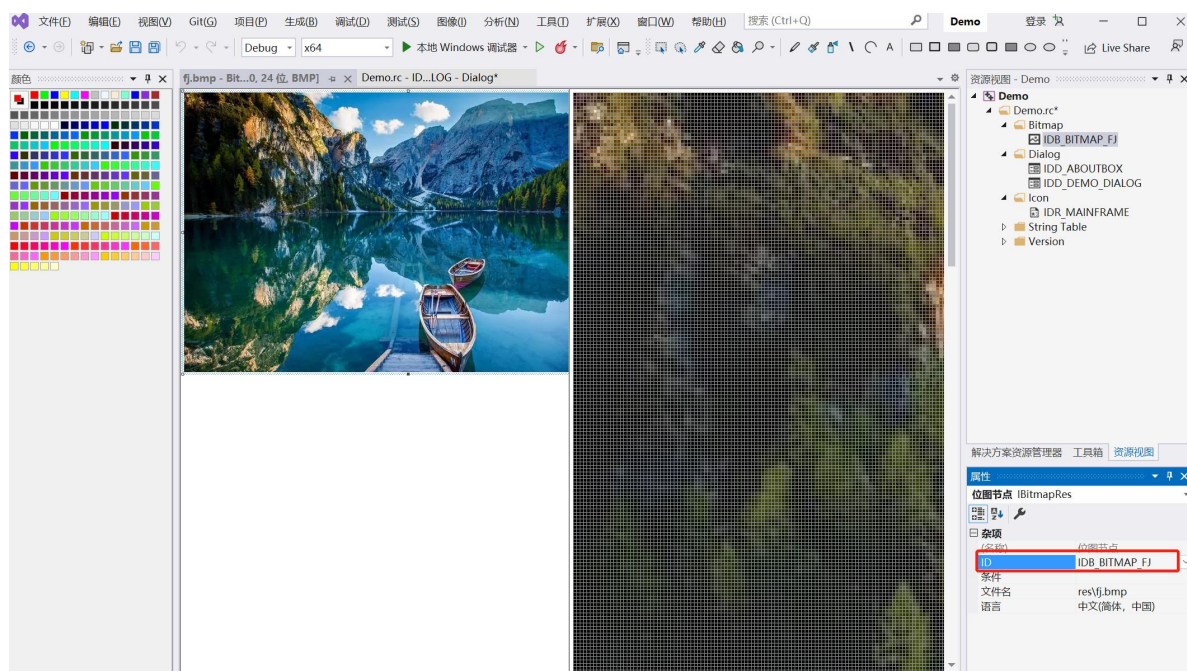


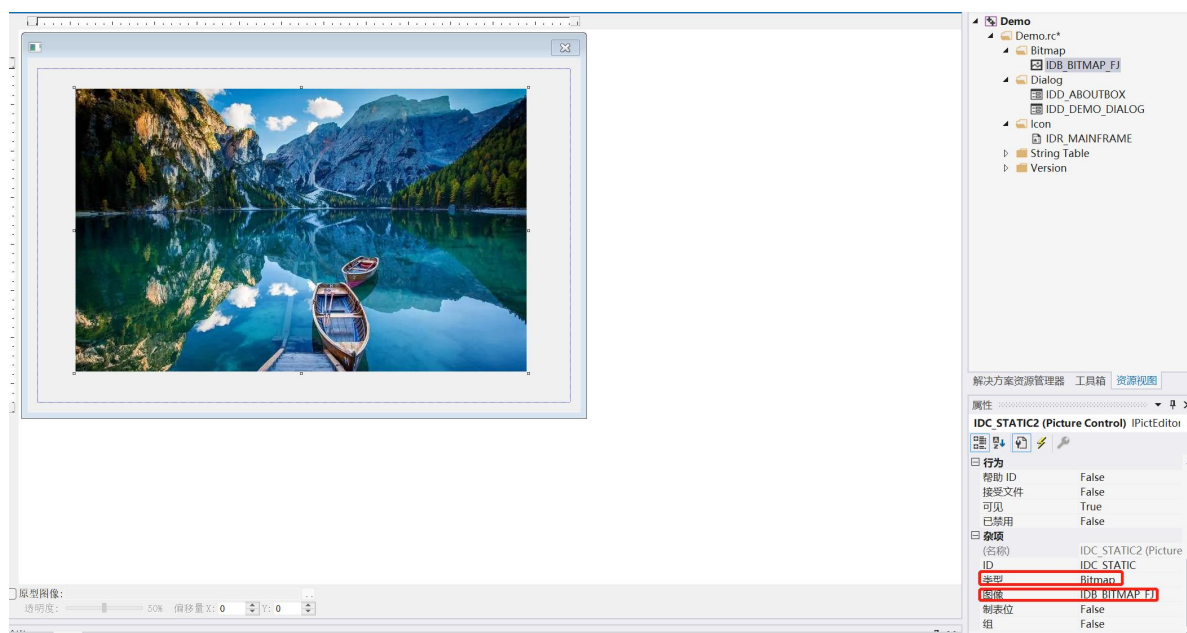
# 图形控件

图像控件Picture Control用于显示图像以美化界面。

首先导入风景图片资源，并重命名ID： IDB\_BITMAP\_FJ。



位图属性选择类型：Bitmap，图像为资源ID： IDB\_BITMAP\_FJ。



## 动态加载位图

导入一张新图片资源，添加一个加载新图片按钮，并给图形控件绑定一个控件类型变量。



加载新图片

```
1 void CDemoDlg::OnBnClickedBtnLoadNewPic()
2 {
3     HBITMAP hBmp = LoadBitmap(AfxGetInstanceHandle(),
4     MAKEINTRESOURCE(IDB_BITMAP_FJ2));
5     m_pic.SetBitmap(hBmp);
6 }
```

## 什么是双缓冲技术？

在我们的电脑中，屏幕中显示的东西都会被放在一个显示缓存中，即单缓冲。在单缓冲中任何绘图的过程都会被立即显示在屏幕中，而双缓冲就是在这个显示的缓冲区外再建一个不显示的缓冲区，所有的绘图都在这个不显示的缓冲区进行，当一帧都绘制完毕后再拷贝到真正的显示缓冲区，这样就不会出现闪烁的现象。



```
1 void CDemoDlg::OnPaint()
2 {
3     if (IsIconic())
4     {
5         .....
6     }
7     else
8     {
```

```

9      CPaintDC dc(this);
10
11      // 1.创建第二缓冲区
12      CDC memDC;
13      memDC.CreateCompatibleDC(&dc);
14
15      CBitmap bmp;
16      bmp.LoadBitmapW(IDB_BITMAP_LOGO);
17
18      // 2.把位图放入第二缓冲区，并在画布上绘制内容
19      memDC.SelectObject(&bmp);
20      memDC.SetBkMode(TRANSPARENT);
21      memDC.TextOutW(0, 0, _T("我的logo"));
22
23      // 3.将第二缓冲区内容拷贝到第一缓冲区，立即显示在屏幕上
24      CRect rect;
25      GetClientRect(rect);
26
27      // dc.BitBlt(250, 680, rect.width(), rect.Height(), &memDC, 0, 0,
SRCCOPY);
28      dc.StretchBlt(250, 680, 200, 100, &memDC, 0, 0, 448, 194, SRCCOPY);
29
30      CDialogEx::OnPaint();
31  }

```



加载新图片

我的logo



