# Classes

### DynamicPorts : UserControl

构造函数，继承与UserControl，读取xml配置文档

### Unit : GraphLinksModelNodeData<String>

### Wire : GraphLinksModelLinkData<String, String>

### CustomModel : GraphLinksModel<Unit, String, String, Wire>

### CustomPartManager : PartManager

### CustomRoute : Route

### InertListBox : ListBox

### InertListBoxItem : ListBoxItem

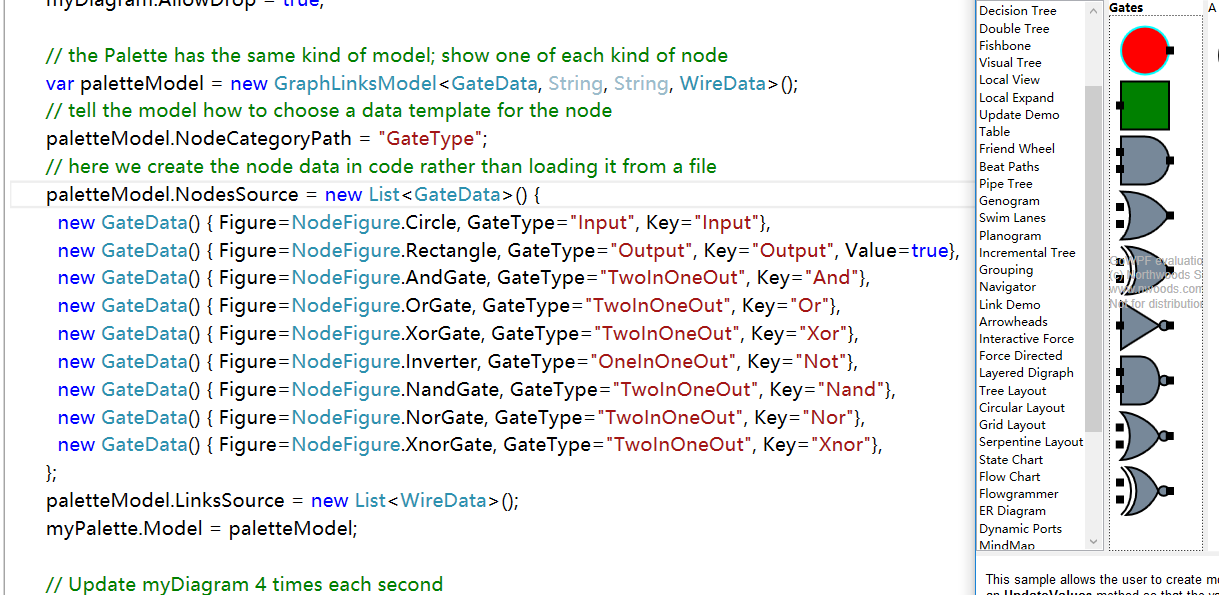
Tips：

LeftSockets（Right、Top、Buttom）可读不可写；

Unit用的是Rectangle、文字用的是textblock（双向绑定）、port用的是自定义的inertListBox

一个Node的主体和文字都定义在一个九宫格grid的中心，剩下四个方向分别是Left-01，Right-21，Top-10，Botton-12

Socket,Unit,Port,



一个节点可以有许多独立的逻辑和物理连接器元素，称为“端口”。只需在任何FrameworkElement上设置go：Node.PortId附加属性即可。

本示例演示了节点如何具有可变数量的端口以及动态添加或移除端口的能力。节点数据类Unit有四个数据列表，每个节点有一个数据表。每个这些数据，套接字，描述一个端口。 Socket.Name用作Node.PortId，Socket.Color用于定制port元素的颜色，Side和Index属性记住端口的位置。

NodeTemplate使用中央矩形周围的四个ListBox。 ListBox.ItemsSource被数据绑定到单元套接字列表。 ListBox.ItemTemplate基本上是一个Rectangle，将go：Node.PortId绑定到Socket.Name唯一字符串。

由于ListBox通常支持选择项目，因此我们使用自定义的InertListBox和InertListBoxItem来禁用鼠标事件处理。

有四个按钮可在选定节点上添加端口。这些按钮由Diagram.SelectedNode是否为非null来启用。点击每个按钮，将一个套接字添加到所选单元的一侧。通过数据绑定的魔力，这将导致相应的ListBox获得一个代表一个端口的额外的FrameworkElement。请注意，如何在事务中执行此更改，以及对模型或其数据的每次更改。

一个特定的端口可以通过右键点击来删除。 （除了不支持右键单击的Silverlight 3）。