New Flow Graph

The new Flow Graph control is a big step forward from the previous version. It introduces a lot of new features while every feature of the old Flow Graph works as before (such as controls inside nodes, page transitions, etc). The Flow Graph architecture has been extended to support many new features such as XML, turning it into a fully featured diagram control.

IMPORTANT: Section #6 has important migration information when moving from version 15.3.1 and below to a newer version.

1. XAML for nodes

Fundamental idea of the new control is giving up of manual drawing of nodes. Any node is now defined by its own markup (written with XAML) and is drawn by the effective Codejock markup engine. This approach enhances the control and gives more freedom for developers. Other flow graph objects such as connection points' circles, connections, node groups are still drawn in the old way.

2. Default nodes; connection points idea reconsideration

CXTPFlowGraphNode(FlowGraphNode) class represents a default node.

Default nodes here and after are known as rectangular tables with caption as a first row. Other rows were called connection points and were added manually. Default nodes are now defined with XAML templates that can be found in resources. Generally a user is not supposed to edit those templates but may take into consideration their code when designing his custom nodes.

A very important idea of the new control is that a connection point is not a table row now, it's a small circle used for connections (default nodes XAML markup defines a connection point with a rectangle actually, a circle is drawn inside). A user must not create connection points by his own: they are created automatically in the Flow Graph after adding a new table row. There's a new convenient method of creating a table row with 0,1,2 connection points (depending on the second argument):

MFC:

```
pNode->AddNamedConnectionPoints(_T("ID"), xtpFlowGraphPointInput);
```

COM:

```
Dim pTableCustomers As FlowGraphNode
Set pTableCustomers = Page.Nodes.AddNode
pTableCustomers.AddNamedConnectionPoints "ID", xtpFlowGraphPointInput
Dim pConnection As FlowGraphConnection
```

A connection point (and its corresponding table row) for default nodes is defined by its unique name. It also has a caption (that appears in a table row). The caption is the same as name by default.

When creating a connection with default nodes a user must not rely on connection points indexes in the code. Instead one should rely on a connection point name (and type as an optional second parameter, when you have 2 actual connection points – input/output in this table row).

MFC:

```
    pConnection->SetOutputPoint(pTableOrders->GetConnectionPoints()-
>FindConnectionPoint(_T("Customer ID")));
    pConnection->SetInputPoint(pTableCustomers->GetConnectionPoints()-
>FindConnectionPoint(_T("ID")));
```

COM:

```
    Set pConnection = Page.Connections.AddConnection
    pConnection.OutputPoint =
pTableOrders.ConnectionPoints.FindConnectionPointByCaption("Customer ID")
    pConnection.InputPoint =
pTableCustomers.ConnectionPoints.FindConnectionPointByCaption("ID")
```

Connection point (table row) can still have an image inside:

MFC:

```
    pTableEmployees->GetConnectionPoints()->FindConnectionPoint(_T("ID"))->SetImageIndex(2);
```

COM:

```
    Dim pTableOrderDetails As FlowGraphNode
    Set pTableOrderDetails = Page.Nodes.AddNode
    pTableOrderDetails.ImageIndex = 1
```

3. Custom nodes

Custom shape is represented as *CXTPFlowGraphNodeCustom(FlowGraphNodeCustom)* class (inherited from *CXTPFlowGraphNode*) or *FlowGraphNodeCustom* in ActiveX.

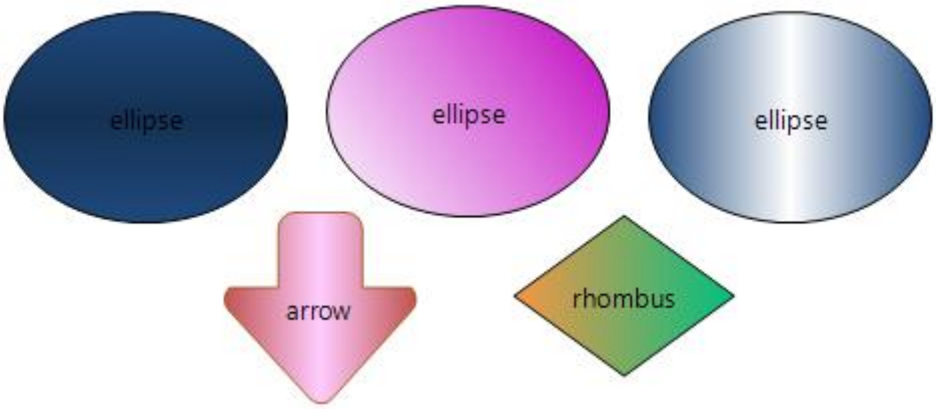
New control makes it possible to add custom nodes of any shape and look. The Codejock Flow Graph sample includes some basic shapes such as rectangle, rhombus, ellipse and some others. Basically a custom node is defined by Path or Shape in its XAML markup. When a custom node is defined by Path node’s connection points are path’s supporting points. When a custom node is a shape (like ellipse or custom image) connection points’ disposition is defined by the **ConnectionPointsIn** property (see *ellipse.xaml* and *image.xaml* samples for details).

Custom shapes can be transparent or filled with a solid color. One can specify its own filling rule in XAML.

Custom shape’s caption is drawn in the center (default) or below the node (xtpFlowGraphCustomNodeCaptionBelow).

Controls inside custom nodes are not supported.

Custom nodes support filling with solid color, as well 6 types of linear gradients. See **SecondColor** and **FillType** properties for more information.



Custom XAML rules

XAML markup of a custom node mush have:

```
                <Canvas x:Name='nodeCanvas'>

                Or for shapes like custom image, ellipse, etc:
                <Canvas x:Name='nodeCanvas' Width='340' Height='172'
ConnectionPointsIn='Rhombus'>
```

As a root element. The first child element of canvas must be a shape (defined by Path, Image, etc.). A Path must have a Stretch='Fill' property for correct resizing of a node and should have a certain name:

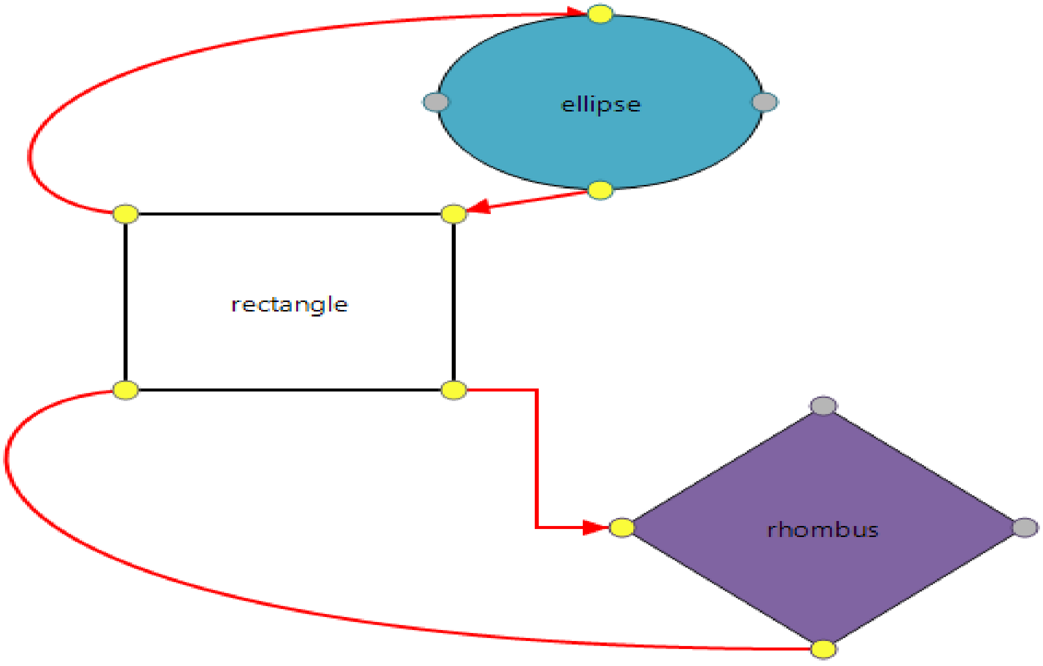
```
                <Path x:Name='nodePath' Stretch="Fill"
```

If you want a node to have a caption you should specify

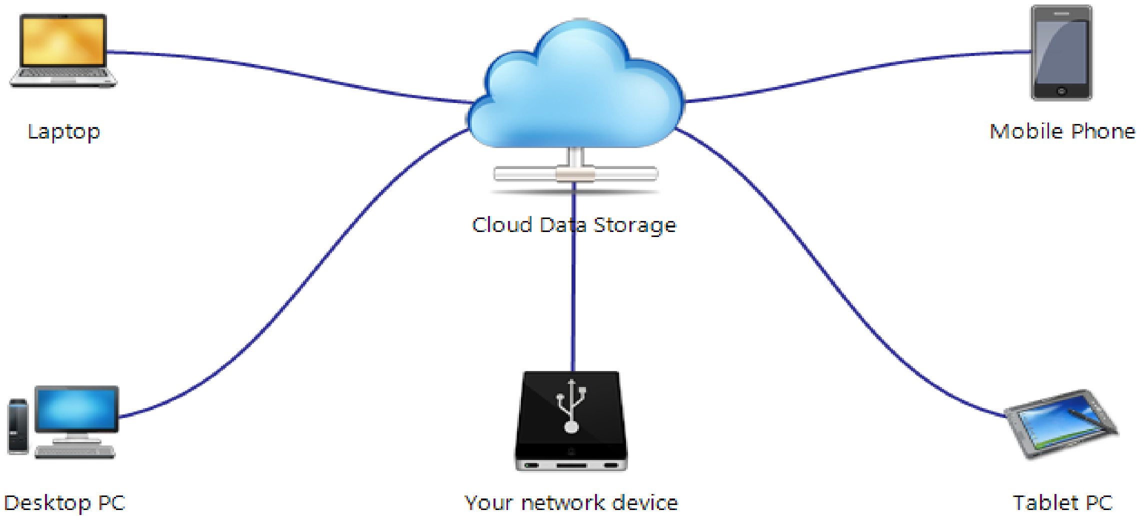
```
                <TextBlock x:Name='nodeCaption'/>
```

as the second child element of Canvas.

<u>Custom nodes sample (nodes defined by XAML paths):</u>

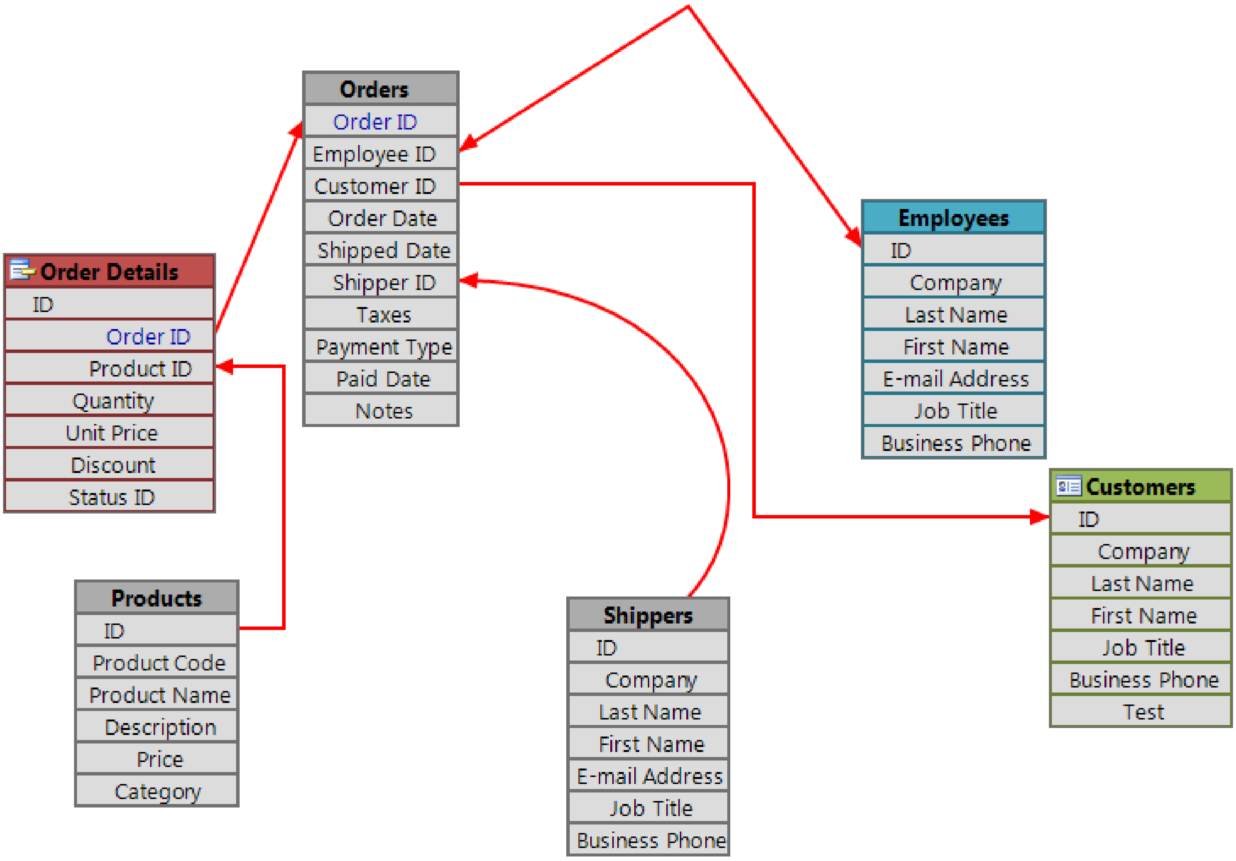


Custom nodes sample (nodes as images):



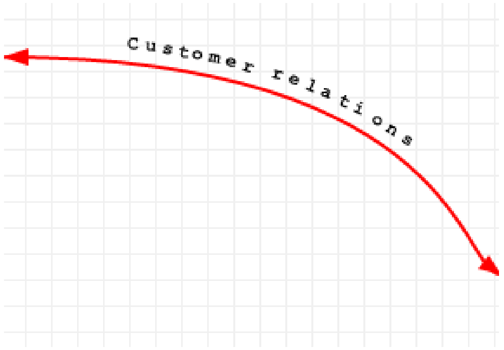
4. New connection types

New Flow Graph supports 9 connection types: Straight, Straight Arrow, Straight Double Arrow, Curved, Curved Arrow, Curved Double Arrow, Elbow, Elbow Arrow, Elbow Double Arrow. The paint manager has a default connection type (Curved) but each connection can have its own style:



5. Connection annotation (caption)

When creating a connection one can define its caption (also known as annotation). When defined, this text will be drawn above the connection line as a shape repeating the line's bending:



A connection can also have a 'Tag' property.

6. Backward compatibility and migration notes

A user has to upgrade his code in order to work with the new Flow Graph. Good news is that it's quite easy.

Creating a table row with connection point(s)

Connection point constructor is protected in MFC now, and ActiveX version has no 'AddConnectionPoint' method. Creating connection points for default nodes(actually a table row with connection points) is easier now.

C++

<u>Old code:</u>

```
CXTPFlowGraphConnectionPoint* pConnectionPoint = pTableCustomers-
>GetConnectionPoints()->AddConnectionPoint(new CXTPFlowGraphConnectionPoint());
pConnectionPoint->SetCaption(_T("ID"));
pConnectionPoint->SetType(xtpFlowGraphPointInput);
```

<u>Node code</u>

```
pNode->AddNamedConnectionPoints(_T("ID"), xtpFlowGraphPointInput);
```

VB

<u>Old code:</u>

```
Dim pConnectionPoint As FlowGraphConnectionPoint
Set pConnectionPoint = pTableCustomers.ConnectionPoints.AddConnectionPoint
pConnectionPoint.Caption = "ID"
pConnectionPoint.Type = xtpFlowGraphPointInput
```

New code

```
pTableCustomers.AddNamedConnectionPoints "ID", xtpFlowGraphPointInput
```

Connection points for custom nodes are also created automatically based on their markup. When a custom node is defined by some Path (most common case), connection points are points of that path.

Themes support

Old FG control supported 2 themes – 'metallic' (default) and 'normal'. The new control's theme for default nodes is now 'metallic', the control doesn't support themes the same way that was used in previous versions. Now users can use custom brushes/colors/etc in XAML to achieve custom drawing of nodes (custom theme).

VisualTheme property has been removed.

Paint Manager

The FG paint manager class is extended and supports some new properties.

AlwaysDrawConnectionPoints property specifies whether to draw connection points always or draw only on selected nodes. TRUE by default.

RecalcConnectionLayoutWhenNeeded property applies only when **AlwaysDrawConnectionPoints** is FALSE. When selection state of nodes in FG changes, it requires to recalculate connection end points. TRUE by default.

ChangeNodeColorWhenSelected property applies to only to default nodes and specifies whether a node changes its color when selected. Otherwise node's caption text changes its color. TRUE by default.

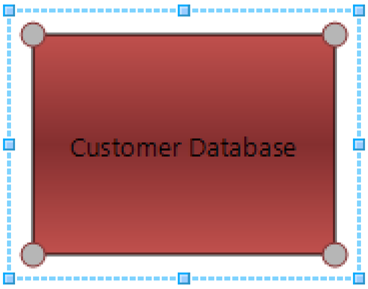
ConnectorType property specifies default connection type for new nodes. xtpFlowGraphConnectorCurved by default.

DrawArrow property is removed and replaced with **ConnectorType** to support all new types of connections.

7. Other

Resizing a node

New FG now uses Visio-like resizing mechanism. Resizing rectangle will appear when a node is selected:



Show/hide a node

A node can be hidden now. Once a node is hidden all connections that involve this node become invisible as well.

Connection point (table row) color

Now ‘Color’ property of a connection point is supported by default nodes. It changes text color of corresponding table row.

Resizing support

Both default and custom nodes can be resized: a resize gripper will appear in the bottom right corner of the node after it’s selected.

Saving a graph to XML file

This feature still works though the format of XML is changed, custom nodes XAML is saved directly to XML file.

Saving a graph to a image file

Use *SaveToFile* method to save the contents of the graph to a PNG/GIF/JPG or BMP file.