CoderForm Specification

Version 3.0.0.0

Contents

Contents 1
Introduction 3
Object model 3
  clsCoderFormManager 5
  clsDataControl (abstract base class) 6
    clsDataContainer (abstract sub class) 11
      clsDataMenu 12
      clsDataLeaves 13
      clsDataBlock 14
        clsDataBlockListBase 15
          clsDataBlockList 16
            clsDataForm 18
            clsDataBlockListIcon 20
          clsDataButtonBlock 20
    clsUDType (abstract sub class) 23
      clsUDBoolean 24
      clsUDInteger 25
      clsUDFloat 26
      clsUDDate 27
      clsUDString 28
      clsUDBitmap 29
      clsDataButton 30
      clsUDTypeList (abstract sub class) 33
        clsUDStringList 33
      clsUDTypeTable (abstract sub class) 35
        clsUDViewSet 35
Core Classes 37
  clsCoreStringBlock 37
  clsCoreCheckBlock 37
Data Control Value Properties 39
Menus 41
Pods 41
Duties and Constraints 42
  Duty Binding Times - enmDutyBinding 42
  Examples of Duties 42
Duty classes 44
  clsDuty 44
  clsDutyAction 45
Constraint classes 46
  clsConstraint (base class) 46
  clsConstraintNumeric 47
  clsConstraintDate 48

clsConstraintString 49
clsConstraintStringList 51
Visual Classes 53
Object Model 53
clsVisual 53
Public Properties 53
clsVisualContainer 54
IPortedControl Interface 55
IPortedControl 55
Plugin Controls 56
Figure 3. Example code for demonstrating plugin 57
Appendix 1. 58
Appendix 2. Documentation note 59

Introduction

CoderForm is a .Net assembly that provides presentation and layout functions to the application writer. It lets the programmer create a user interface in code, rather than by using the Visual Studio form designer.

This document provides the specification for the CoderForm classes. It is best used as a reference rather than as a primer. The best way to learn how to use CoderForm is to use the QuickStart document or the demo solution. The demo solution for VB.Net is called CoderFormVisual and for C#, CoderFormSharp.

Object model

A simplified object model for CoderForm is as follows:

clsDataControl (abstract base class)

- clsDataContainer (abstract base class)
  - clsDataMenu
  - clsDataLeaves
  - clsDataBlock
    - clsDataBlockListBase (abstract base class)
      - clsDataBlockList
        - clsDataForm
        - clsCoreStringBlock
        - clsCoreCheckBlock
  - clsDataButtonBlock
- clsUDType (abstract base class)
  - clsUDBoolean
  - clsUDInteger
  - clsUDFloat
  - clsUDDate
  - clsUDString
  - clsUDStringList

The abstract base class *clsDataControl* is an abstraction of a control. In this document, objects of class clsDataControl and its heirs may be referred to as *Data Controls* or *Controls*. Data Controls may either be containers, which may contain other Data Controls, or UDTypes, which represent primitive data types such Boolean or String. The user interface is defined by adding Data Controls to a clsDataForm object (highlighted), which itself renders as a Windows form.

Controls are added to a container in a specified direction, which defaults to top-down. They may be positioned relative to one another by specifying one of nine positional relations – TopLeft, TopMiddle etc., and offsets added. This provides a means of specification that is often easier and faster than direct manipulation of control elements. The positional relations are illustrated in Appendix 1.

clsCoderFormManager

clsCoderFormManager manages other objects in the assembly. For the assembly user, it is sufficient to instantiate an object of this class before any use of CoderForm objects and to dispose the object after they are no longer needed.

Public Methods

New ()
New (venmMenuStyle)
New (venmMenuStyle, vblnHDCFix)
New (venmMenuStyle, vblnHDCFix, vstrKey)

The constructor takes parameters of menu style, HDC fix and licence key.
The menu style is of type enmMenuStyleSet and defaults to enmMenuStyleSet.VisualStudio, which is a style similar to that used in the Visual Studio IDE.
The parameter vblnHDCFix is a boolean value, which when true fixes a problem in the DotNet Framework prior to version 1.1 SP1. This problem manifests with a slow display of the created form, caused by unnecessary .Net garbage collections. vblnHDCFix defaults to false.
The license key is a 16 digit string provided by Five Tuple. If you do not have a licence key, leave this string empty.

Public Properties

WriteOnly enmMenuStyle as enmMenuStyleSet

Sets the menu style. This currently sets the selected and background colours depending on the enumeration provided.

WriteOnly colMenuSelected as Color

Sets the colour of the currently selected menu item.

WriteOnly colMenuBackground as Color

Sets the background colour of the non selected menu item.

ReadOnly objCorePod () As IPod

Gets the core pod for the application. The core pod hold core image resources for the Coderform dll. For more information, see Pods. This property is exposed in order to allow a container user pod to be set to the core pod. Since, however clsUDBitmap and clsDataMenu constructors are

provided which default to the core pod, it should not be necessary to use this property.

---

clsDataControl (abstract base class)

clsDataControl is a base class which manages data associated with a Windows control. It holds a reference to a *ctlVisual* object which handles formatting and wraps the Windows.Forms.Control object,.

Public Properties

strTag

The unique identifier of the data control

strCaption

A descriptive caption for the data control

ReadOnly ctlVisual as clsVisual

Returns the visual associated with this Data Control. This is exposed to allow formatting to be set.

objContainer As clsDataContainer

Returns the container in which this control is held, if any. This property should not be set. It is used typically for identifying a container from a more readily accessible contained control.

ReadOnly objDataControl (vstrTag) As clsDataControl

Gets the clsDataControl whose tag is given. The first object in the hierarchy collection which matches the provided tag is returned.

ReadOnly objUDType (vstrTag) As clsUDType

Gets the clsUDType whose tag is given. The first object matching the provided tag is returned.

ReadOnly objGetDataMenu (vstrTag) As clsDataMenu

Gets the clsDataMenu whose tag is given. The first object matching the provided tag is returned. The search is not case sensitive.

Value properties

blnValue()
intValue()
intSelection()
aintValue()
aintSelection()
dblValue()
datValue()
strValue()
astrValue()

objGamut()
objValue()
objValueUI()

These properties return the type representation of this data control. They are typically valid only for objects of clsUDType, but they may be overriden by other objects of type Data Control. See the section Data Control Value Properties for a more detailed explanation.

Public Methods

Spatial Format Methods

Most spatial formatting is performed using a Data Control's ctlVisual object. Some methods for relative positioning are provided in clsDataControl.

Locate (vobjBlockFixed, venmRelation)
Locate (vobjBlockFixed, venmRelation, vblnInside)

This method physically locates this Data Control with respect to another. Locate may be called only once on a Data Control, and it may not be called after Fix. The position is specified by venmRelation – see **enmPosition**. The parameter vblnInside defaults to false. If it is true, then the data control, on which Locate is called, is placed inside the fixed data control. This overlays two controls and is typically used for layering panels. Only data controls which are in the same container may be positioned relative to one another. Note that adding to a clsDataBlockList calls Locate on an object.

Fix (venmPosition)

Fixes this block relative to its container. Fix may be called on a Data Control which has not already had a call to Locate or Fix. Note that adding an object to a clsDataBlockList object calls Locate on the added object.

Duty Methods

See the section on Duties and Constraints for an explanation of duties, actions and constraints.

objAddDuty (venmBinding, venmAction)

Adds a *duty* to this Data Control without specifying a constraint. The action is on the Data Control.

objAddDuty (venmConstraintString, venmBinding, venmAction)

Adds a *duty* to this Data Control using a predefined string constraint. See the following method for a description of other parameters.

objAddDuty (vobjConstraint, venmBinding, venmAction)

Parameter	Meaning	Default
vobjConstraint	clsConstraint object	
venmBinding	The event at which point this duty is performed.	
venmAction	Action associated with duty	

Adds a *duty* to this Data Control.

objAddDuty (vobjConstraint, [vintBinding], [venmAction], [vobjTarget])

Parameter	Meaning	Default
vobjConstraint	clsConstraint object	
vintBinding	The event at which point this duty is performed. Events may be ANDed to allow the duty to run at more than one event time.	enmDutyBinding .None
venmAction	Action associated with duty	enmControlStat e.Message
vobjTarget	clsDataControl object on which action is applied.	Owner

Adds a *duty* to this Data Control. See the

Duties and Constraints section for an explanation.

blnDoDuty (venmBinding, venmAction, [vobjTarget])

Parameter	Meaning	Default
venmBindTime	The event at which point this duty is performed.	
rblnMessageShown	If True on input, no message is shown , should any action result in one. If a message is shown by an action, this parameter is set True.	

Performs all duties defined on this Data Control for the given bind time.

SetState (venmState)
SetState (venmState, vstrMessage)

Sets the state of this control using the enmControlState specified.

Menu Methods

AddDataMenu(vobjDataMenu)

This method takes an object of type clsDataMenu and adds it to the data controls's menu collection. An error is thrown if the object does not support the tyep of menu added; a main menu (a clsDataMenu object set with type enmMenuMain) may be added only to a clsDataForm object or to another main menu; context menus may not be added to a main menu.

objAddDataMenu(vstrText)
objAddDataMenu(vstrText, vstrTag)
objAddDataMenu(vstrText, vstrTag, venmMenuSiting)

This factory method generates, adds and returns a clsDataMenu object, with errors thrown as described in AddDataMenu. The tag defaults to the text and the siting defaults to Main.

objAddMenu(vstrText, vstrItems)
objAddMenu(vstrText, vstrItems, venmMenuSiting)
objAddMenu(vstrText, vstrItems, venmMenuSiting, strDelimiter)

This factory method generates, adds and returns a clsDataMenu object with the supplied header text and items. Item values are delimited by string strDelimiter, defaulting to comma, and are trimmed of spaces. A menu spacer can be added simply by including a hyphen in the list. Errors are thrown as

described in AddDataMenu. The tag defaults to the text and the siting defaults to Main.

Public Events

When an event occurs in a Data Control, that event is raised in its container Data Control to any number of levels, so that the event may be seen in the top level form.

```
Public Event evtDataChanged(ByVal vobjSender As Object, _
                            ByVal vobjCommandEventArgs As
                            clsDataChangedEventArgs)
```

This event is raised when a UDType's data changes.
clsDataChangedEventArgs has the following properties

Parameter	Meaning
strTag	Tag of UDType whose data has changed
objUDType	clsUDType whose data has changed

```
Public Event evtDataChosen (ByVal vobjSender As Object, _
                            ByVal vobjCommandEventArgs As
                            clsDataChosenEventArgs)
```

This event is raised when a UDType's data is selected.
clsDataChosenEventArgs has the following properties

Parameter	Meaning
strTag	Tag of UDType whose data has changed
objUDType	clsUDType whose data has changed

```
Public Event evtCommandClicked(ByVal vobjSender As Object, _
                               ByVal vobjCommandEventArgs As
                               clsCommandClickedEventArgs)
```

This event is raised when a command control that is, a button, is clicked.

clsCommandClickedEventArgs has the following properties

Parameter	Meaning
strTag	Tag of command control clicked
objCommandData	clsDataControl associated with command.

```
Public Event evtMenuClicked(ByVal vobjSender As Object, _
                            ByVal vobjEventArgs As
                            clsMenuClickedEventArgs)
```

This event is raised when a menu item on the form is clicked.

vobjEventArgs has the following properties

Parameter	Meaning
strTag	Text of menu item clicked excluding any embedded ampersand.
objCommandData	clsDataControl associated with menu
strValue	Value of clsDataControl associated with menu

---

clsDataContainer (abstract sub class)

```
Inherits Protronics.User.clsDataControl
```

Class clsDataContainer is a representation of a control container. It keeps a collection of clsDataControl objects and can generate a control from them. The two children of this class are *clsDataLeaves*, which renders each contained Data Control as a tab page, and *clsDataBlock*, which renders each Data Control in a rectangular block.

Public Properties

```
Default Public Property Item(vstrTag) As String
```

Returns the string representation of the first contained control with the given tag.

```
WriteOnly  objUserPod () As IPod
```

Sets the user pod for this container. The user pod holds the bitmap or icon images for clsDataMenu or clsUDBitmap objects which are added to the container. Pods hold image resources. See section Pods for further explanation.

```
ReadOnly ctlVisualContainer() As clsVisualContainer
```

Returns the ctlVisualContainer object to allow setting of margin and grid properties.

Public Methods

AddDataControl (vobjControl)

This method adds the supplied data control to the container.

---

clsDataMenu

```
Inherits Protronics.User.clsDataContainer
```

The class clsDataMenu represents a menu object. See Menus for more detail.

Public Properties

mnuMenuItem As MenuItem

Returns the Windows menu item object

icoMenuIcon As Icon

Windows icon object to be shown with menu option. The icon may be created and set by the user.

enmIcon As enmMenuIcon

Icon from core pod to be shown with menu option.

intMenuIcon As Integer

Icon from user pod to be shown with menu option. The value is the index of the icon, which is its enumeration value.

enmMenuSiting As enmMenuSitingSet

This property sets or returns the menu type as main or context. A menu can be both, in which case it will be displayed as both kinds on a form.

objGetDataMenu (vintIndex) As clsDataMenu

Searchs for and returns the menu with the zero-based index. This method is an overloaded form of that in clsDataControl, which takes a tag value.

objCommandData As clsDataControl

Returns command data associated with the menu. The command data is the Data Control returned with the event when this menu item is clicked

objValidatedControl As clsDataControl

Control which is to be validated when menu is selected

Public Methods

This constructor allows the user to specify a handler which will be invoked when the menu is clicked. The handler has no parameters.

New (vstrCaption)
New (vstrCaption, vstrTag)
New (vstrCaption, vstrTag , vobjCommandData, [venmMenuIcon])

Parameter	Meaning	Default
vstrCaption	Caption, used if the container is bounded.	
vstrTag	Unique identifier for data control	vstrCaption
vobjCommandData	Data Control to be returned with event when this menu item is clicked	Nothing
venmMenuIcon	Icon of type enmMenuIcon from core pod displayed with menu caption	Nothing

New (vstrCaption, vstrTag, vdlgHandler, [venmMenuIcon])

Parameter	Meaning	Default
vstrCaption	Caption, used if the container is bounded.	
vstrTag	Unique identifier for data control	
vdlgHandler	Address of delegate to be invoked when menu is clicked	
venmMenuIcon	Icon of type enmMenuIcon from core pod displayed with menu caption	Nothing

---

clsDataLeaves

```
Inherits Protronics.User.clsDataContainer
```

The class `clsDataLeaves` is a mapping to the tabbed dialog control container. Each contained control is placed on a tab whose caption is the caption of the contained control. Contained controls may themselves be containers.

Public Methods

New (vstrCaption)

The caption is not used.

AddDataControl (vobjControl, vstrLeafCaption)

This method adds the supplied data control to the container and sets the leaf, that is the tab, caption to vstrHeaderCaption.

---

clsDataBlock

```
Inherits Protronics.User.clsDataContainer
```

This container class represents a rectangular block. Controls are added to the container in a direction specified by the property *enmDirection*. If the object's ctlVisualContainer object has its property *intListLength* set, then no more than *intListLength* controls will be added in the specified direction: subsequent controls will start a new line. Similarly, if the added object has a ctlVisual with the *blnCR* property true, a new line will be started.

Thus controls are added, in effect, to a rectangular grid which defaults to a single column and unlimited rows. Where there is more than one column or row, controls are added in a raster scan of length intListLength. The relative positioning of controls, defined by the clsDataControl methods Locate and Fix may be used in conjunction with the grid layout.

Public Properties

enmDirection() As enmDirectionSet

This enumeration determines the direction in which the controls are placed in the block by the packer. It defaults to `enmDirectionSet.TopDown` if not set.

Public Methods

New (vstrCaption)

New (vstrCaption, venmBorder)
New (vstrCaption, venmBorder, vstrTag)

The constructor parameter venmBorder determines how the data is presented. The tag defaults to the caption. The border defaults to a rectangle.

---

clsDataBlockListBase

```
Inherits Protronics.User.clsDataBlock
```

This abstract class overrides clsDataBlock, adding additional formatting properties.

Public Properties

enmCompaction As enmCompactionSet

This property represents spacing between contained controls, and controls and the border.

Public Methods

New (vstrCaption, venmBorder, venmCompaction, vstrTag)

Parameter	Meaning	Default
vstrCaption	Caption, used if the container is bounded.	
venmBorder	Border style of type enmBorderStyle	
venmCompaction	Enumeration determining inner margin of block and spacing of contained controls	
vstrTag	Unique identifier for data control	

AddDataControl (vobjControl)
AddDataControl (vobjControl, venmPosition)
AddDataControl (vobjControl, venmPosition, venmCompaction)

Parameter	Meaning	Default
vobjControl	clsDataControl to be added to block	
venmPosition	Enumeration determining position relative to last added control.	enmPositionSet. DefaultPosition
venmCompaction	Determines spacing from last control. If not specified, the default for the object is used.	enmCompaction Set.DefaultComp action

This is an overriden implementation of the function in clsDataContainer which allows the position of a control to be determined relative to the last added control. If venmCompaction is not set to None, then padding is added to one or two sides using a best guess heuristic.

---

clsDataBlockList

```
Inherits Protronics.User.clsDataBlockListBase
```

This class overrides clsDataBlockListBase to create a formatted container with the compactness of the contained controls defined by this class, and the direction of their placement defined by the ancestor clsDataBlock. If controls have a caption, this is shown to the left of the control. The class includes factory methods which creating a UDType and adds it automatically to the container collection.

Public Properties

enmLabelPosition As enmPositionSet

This property represents the position of the label relative to its control.

Public Methods

New (vstrCaption)

Constructor defaulting to a rectangular border style and with other defaults as below.

New (vstrCaption, venmBorder, [venmCompaction], [vstrTag])

Parameter	Meaning	Default
vstrCaption	Caption, used if the container is bounded.	
venmBorder	Border style of type enmBorderStyle	
venmCompaction	Enumeration determining inner margin of block and spacing of contained controls.	enmCompactionSet.Regular
vstrTag	Unique identifier for data control	vstrCaption

AddDataControl (vobjControl, vblnAddCaption)

This is an overriden implementation which allows addition of a caption

Factory Methods

The following factory methods all return an object which is a child of clsUDType. Each method takes a caption, and optional initial data and tag. The parameter vstrTag always defaults to the caption.

objAddBoolean (vstrCaption) As clsUDBoolean
objAddBoolean (vstrCaption, vblnInitialData, vstrTag) As clsUDBoolean

Adds a boolean UDType to the container. Initial data is False

objAddInteger (vstrCaption) As clsUDInteger
objAddInteger (vstrCaption, vintInitialData, vstrTag) As clsUDInteger

Adds an integer UDType to the container. Initial data is 0.

objAddFloat (vstrCaption) As clsUDFloat
objAddFloat (vstrCaption, vdblInitialData, vstrTag) As clsUDFloat

Adds a float UDType to the container. Initial data is 0.

objAddDate (vstrCaption) As clsUDDate
objAddDate (vstrCaption, vdatInitialData, vstrTag) As clsUDDate

Adds a date UDType to the container. Initial data defaults to the current date.

objAddBitmap(venmIcon) As clsUDImage
objAddBitmap(venmIcon, vstrCaption, vstrTag) As clsUDImage

Adds a bitmap UDType to the container. The type of venmIcon is enmLib, which represents a bitmap from the core pod.

objAddBitmap(vintIcon, vstrCaption, vstrTag) As clsUDImage

Adds a bitmap UDType to the container. The parameter vintIcon represents an index of the bitmap for the current container user pod. This parameter is set to a bitmap enumeration value which is supplied with the chosen user pod.

objAddString(vstrCaption) As clsUDString

Adds an empty string UDType to the container, using a label. The tag and the initial data is set to the caption.

objAddString(vstrCaption, venmVisualType) As clsUDString

Adds an empty string UDType to the container, using the specified visual type. The tag is set to the caption.

objAddString(vstrCaption, vstrInitialData, venmVisualType) As clsUDString

Adds a string UDType to the container, using the specified visual type, setting initial data and using a label. The tag is set to the caption.

objAddString(vstrCaption, [vblnInitialData], [vstrTag], [vintLength],
[vintMaxLength], [venmVisualType]) As clsUDString

Adds a string UDType to the container, using a textbox. Initial data is “”. The parameter vintLength determines the width of the textbox, and vintMaxLength, the maximum width of entered data. Both default to 30. The visual type defaults to a textbox.

---

clsDataForm

```
Inherits Protronics.User.clsDataBlockList
```

This subclass extends clsDataBlockList to provide a form. Two significant methods are frmGenerateForm, which generates and returns a Windows.Forms.Form object which contains this control, and Expose, which shows the object.

Public Properties

blnCancelled As Boolean

This property is set toTrue when a form is cancelled.

WriteOnly AcceptButton as clsDataButton

This property sets the provided Data Button to be the accept button of Windows Form.

WriteOnly CancelButton as clsDataButton

This property sets the provided Data Button to be the cancel button of Windows Form

Public Methods

New ()
New (vstrCaption)
New (vstrCaption, venmFormStyle)
New (vstrCaption, venmFormStyle, venmButtonBlock)
New (vstrCaption, venmFormStyle, venmButtonBlock,
[venmButtonPlacement], [vstrTag])

Parameter	Meaning	Default
vstrCaption	Caption, used if the container is bounded.	“”
venmFormStyle	Type of form: fixed dialog; sizeable; options dialog.	Fixed Dialog
venmButtonBlock	Contained buttons.	enmButtonTypes. OKCancel
venmButtonPlacement	Position of button controls on form.	enmPositionSet. BottomRight
vstrTag	Unique identifier for data control	vstrCaption

frmGenerateForm As clsVisualForm

This method generates a Windows.Forms.Form object which displays the contained controls. The form is not shown.

Expose([vblnDialog])

This method generates the form if necessary, and shows it. The parameter vblnDialog defaults to false. If it is true, then the form is displayed as a modal dialog.

Close()

This method closes the form.

Cancel()

This method cancels the form.

Public Events

```
Public Event evtVisualClosed(ByVal vobjSender As Object, _
                             ByVal vobjCloseEventArgs As
                             clsCloseEventArgs)
```

This event is raised when the form is closed using the OK or Cancel buttons or the Windows close command. The Class clsCloseEventArgs has a single Boolean parameter, blnCancelled, which, if true, shows that the form has been cancelled rather than closed.

clsDataBlockListIcon

```
Inherits Protronics.User.clsDataBlockList
```

This subclass extends clsDataBlockList to add an icon to a block.

Public Methods

New (vstrCaption, venmIcon)

Constructor taking an icon from the core pod. Defaults to a rectangular border, and otherwise as constructor below.

New (vstrCaption, venmBorder, [venmCompaction], [venmIcon], [vstrTag])

Parameter	Meaning	Default
vstrCaption	Caption, used if the container is bounded.	
venmBorder	Border style of type enmBorderStyle	
venmCompaction	Enumeration determining spacing of contained controls.	enmCompactionSet.Regular
venmIcon	System defined icon.	enmIconSet.IconUndefined
vstrTag	Unique identifier for data control	vstrCaption

clsDataButtonBlock

```
Inherits Protronics.User.clsDataBlockListBase
```

This subclass extends `clsDataBlockList` to create a block that is convenient for adding a number of buttons to the block.

Public Properties

objDataButton (vstrTag) as clsDataButton

Returns the button with the specified tag from those added to a form. An exception is thrown if the specified control is not a clsDataButton

objDataButton (vintIndex) as clsDataButton

Returns the button with the specified index from those added to a form. The index is the order in which the control was added, starting at 0. An exception is thrown if the specified control is not a clsDataButton

objButtonOK () as clsDataButton

Returns the predefined OK button, if any

objButtonCancel () as clsDataButton

Returns the predefined Cancel button, if any

Public Methods

New (venmButtonBlock, venmDirection)

Constructor specifying buttons to include and direction of listing. Defaults for properties not included in this constructor are as in table below.

New (venmButtonBlock, venmBorder)

Constructor specifying buttons to include and kind of border. Defaults for properties not included in this constructor are as in table below.

New (venmButtonBlock, venmDirection, venmCompaction, [vstrCaption], [vstrTag], [venmBorder])

Parameter	Meaning	Default
venmButtonBlock	A predefined set of buttons to add.	
venmDirection	Direction in which buttons are added.	enmDirectionSet.DefaultDirection
venmCompaction	Enumeration determining spacing of contained controls.	enmCompactionSet.DefaultCompaction
vstrCaption	Caption, used if the block is bounded.	Empty string
vstrTag	Unique identifier for data control	vstrCaption
venmBorder	Border style of type enmBorderStyle	None

The default direction is RightLeft for a standard button set, and TopDown if no buttons are added by the constructor (enmButtonTypes.None). The compaction defaults to Regular.

New (vstrButtons)
New (vstrButtons, venmDirection, [venmCompaction], [vstrCaption], [vstrTag], [venmBorder])

Parameter	Meaning	Default
vstrButtons	CSV list of button captions. The delimiter may be changed from the default of comma to that specified in vstrDelimiter.	
venmDirection	Direction in which buttons are added.	enmDirectionSet. DefaultDirection
venmCompaction	Enumeration determining spacing of contained controls.	enmCompactionSet .DefaultCompaction
vstrCaption	Caption, used if the block is bounded.	Empty string
vstrTag	Unique identifier for data control	vstrCaption
venmBorder	Border style of type enmBorderStyle	None
vstrDelimiter	Delimiter used in vstrButtons.	Comma

Constructors which use a CSV list of captions for the buttons to be created. In this case, each button is assigned a tag which is the tag of this object concatenated with the button number, indexed from 0. So the second button in a clsDataButtonBlock tag named “ButtonBlock” has the tag “ButtonBlock1”. The default direction is RightLeft for a standard button set, and TopDown if no buttons are added by the constructor (enmButtonTypes.None). The default compaction is regular.

objAddDataButton (vstrCaption)
objAddDataButton (vstrCaption, vstrTag, [vintWidth], [vintHeight]) as clsDataButton

Factory method to create and add a button.

Parameter	Meaning	Default
vstrCaption	Caption to appear as button label	
vstrTag	Unique identifier for data control	vstrCaption
vintWidth	Button width	0 (Fixed default width)
vintHeight	Button height	0 (Fixed default height)

clsUDType (abstract sub class)

This is an abstract class which handles different types of user data. The heirs of this class deal with elemental types such as String and Integer.

Public Properties

Formatting properties

These methods pass though to the *IPortedControl* object held by the clsUDType. The IPortedControl object wraps the Windows control for UD Types. A value of –1, or gintAUTO, means that the size is set automatically to fit the data. A value is –2, or gintDEFAULT, sets a predefined default value. After the UDType has been generated, the automatic or default values are changed to the actual value, or an approximation to it. Although the type of these properties is Single, they are usually set to integer values, denoting a number of characters or rows.

The formatting properties are valid only after the clsUDType is instantiated or, for custom controls, when the objPortedControl object has been instantiated and associated with the clsUDType. The properties, although ReadWrite, should only be written, not read.

sglDataWidth ()

Gets or sets the width of displayed data in units appropriate to that data. For example, for a string this property determines the length of data displayed.

sglDataWidthMax()

Gets or sets the maximum width of displayed data in units appropriate to that data. For example, for a string this property determines the maximum length of data that may be entered.

sglDataHeight()

Gets or sets the height of displayed data in units appropriate to that data. For example, with a string this property determines the number of lines displayed.

sglDataHeightMax()

Gets or sets the maximum height of displayed data in units appropriate to that data.

objValue()

Overriden from clsDataControl. This property gets or sets an object which represents the definitive value of this clsUDType. If the object uses direct binding, then the value appears immediately in the UI. If not, it appears when Bind(True) is called. Direct binding is used unless binding is explicitly set by the user.

objPortedControl()

Returned the IPortedControl object for this UDType.

Shared Behaviour

All heirs of this class implement the following behaviours.
Caption text may be tagged as follows:

<nl> Line Feed - Carriage Return
<tt> Four spaces

---

clsUDBoolean

```
Inherits Protronics.User.clsUDType
```

This class manages a boolean type, rendering it as a control whose state represents the value of the associated data.

Visual Types

The only visual type implemented for the clsUDBoolean is currently a checkbox. It also supports custom controls.

Public Properties

See Data Control Value Properties

Formatting properties

Formatting properties are not implemented for this type.

Public Methods

New(vstrCaption, vblnInitialData)
New(vstrCaption, vblnInitialData, vstrTag, [venmVisualType])

Parameter	Meaning	Default
vstrCaption	Text associated with data object.	
vblnInitialData	Initial value of the data object	False
vstrTag	Unique identifier for data control	vstrCaption
venmVisualType	Visual control with which data is rendered	enmBoolean. CheckBox

---

clsUDInteger

```
Inherits Protronics.User.clsUDType
```

This class manages an integer type, rendering it as a control whose state represents the value of the underlying data.

Visual Types

The visual types implemented for this clsUDType are as follows.

Type	Description
Label	Windows label control
IntegerTextbox	Windows textbox with integer only input
IntegerTextboxPositive	Windows textbox with positive integer only input
Custom	User supplied control

Properties

See Data Control Value Properties

Formatting properties

For the textbox visual types, sglDataWidth is the width, in characters, of the textbox and sglDataWidthMax is the maximum length of the integer, including sign, that may be entered. The property sglDataWidth defaults to 4. The property sglDataWidthMax defaults to 8. The properties sglDataHeight and sglDataHeightMax are not used.

Public Methods

New(vstrCaption)
New(vstrCaption, vintInitialData)
New(vstrCaption, vintInitialData, vstrTag, [venmVisualType])

Parameter	Meaning	Default
vstrCaption	Text associated with data object.	
vintInitialData	Initial value of the data object	0
vstrTag	Unique identifier for data control	vstrCaption
venmVisualType	Visual control with which.data is rendered	enmInteger. TextBox

---

clsUDFloat

```
Inherits Protronics.User.clsUDType
```

This class manages a floating point data type, rendering it as a control whose state represents the value of the underlying data.

Visual Types

The visual types implemented for this clsUDType are as follows.

Type	Description
Label	Windows label control
FloatTextbox	Windows textbox with float only input
FloatTextboxPositive	Windows textbox with positive float only input
Custom	User supplied control

Properties

See Data Control Value Properties

Formatting properties

For the textbox visual types, sglDataWidth is the width, in characters, of the textbox and sglDataWidthMax is the maximum length of the number, including sign and radix, that may be entered. The property sglDataWidth defaults to 4. The property sglDataWidthMax defaults to 8. The properties sglDataHeight and sglDataHeightMax are not used.

Public Methods

New(vstrCaption)
New(vstrCaption, vdblInitialData, [vstrTag], [venmVisualType])

Parameter	Meaning	Default
vstrCaption	Text associated with data object.	“”
vdblInitialData	Initial value of the data object	0
vstrTag	Unique identifier for data control	vstrCaption
venmVisualType	Visual control with which.data is rendered	enmFloat.Text Box

---

clsUDDate

```
Inherits Protronics.User.clsUDType
```

This class manages a date data type, rendering it as a control whose state represents the value of the underlying data.

Visual Types

The visual types implemented for this clsUDType are as follows:

Type	Description
DateTimePickerLongDate	Windows date time picker. Long date format
DateTimePickerLongDate	Windows date time picker. Long date format
Custom	User supplied control

Public Properties

See Data Control Value Properties

Formatting properties

For the Windows DateTimePicker visual types, sglDataWidth is the width, in characters, of the control. The other formatting properties are not used.

Public Methods

New(vstrCaption, vdatInitialValue)
New(vstrCaption, vdatInitialValue, vstrTag, [venmVisualType])

Parameter	Meaning	Default
vstrCaption	Text associated with data object.	
vdatInitialValue	Initial value of the data object	
vstrTag	Unique identifier for data control	vstrCaption
venmVisualType	Visual control with which.data is rendered	enmDate.DateTimePickerLongDate

---

clsUDString

```
Inherits Protronics.User.clsUDType
```

This class manages an string data type, rendering it as a control whose state represents the value of the underlying data.

Visual Types

The visual types implemented for the clsUDString are as follows:

Type	Description
Label	Windows label control
LabelSeparator	Separator bar with a label caption
LabelGradient	Gradient color bar with a caption
Textbox	Windows textbox control
Passwordbox	Windows textbox control with hidden data entry
Custom	User supplied control

Public Properties

See Data Control Value Properties

Formatting properties

For the visual types Textbox and Passwordbox, sglDataWidth is the width, in characters, of the textbox. The property sglDataHeight determines the height, in characters, of the textbox. The property sglDataWidthMax determines the maximum length of the string that may be entered. The property sglDataWidth defaults to 30. The property sglDataWidthMax defaults to 30. The property sglDataHeightMax is not used.

For the type Label, the properties sglDataWidth and sglDataHeight are the label width and height, in characters, respectively. By default they are sized to the initial text. The properties sglDataWidthMax and sglDataHeightMax are not used.

For the type LabelSeparator, the property sglDataWidth determines the length, in characters, of the entire separator bar including caption. It defaults to four times the length of the caption if the caption is not null. Otherwise it defaults to length 40.

The formatting properties are not implemented for the types LabelSeparator and LabelGradient.

Public Methods

New (vstrCaption, venmVisualType)

New ([vstrCaption], [vstrInitialData], [vsglDataWidth],[vsglDataWidthMax], [vstrTag], [venmVisualType])

Parameter	Meaning	Default
vstrCaption	Text associated with data object.	“”
vstrInitialData	Initial value of the data object	“”
vstrTag	Unique identifier for data control	vstrCaption
vsglDataWidth	Characters appearing in the UI	30
vsglDataWidthMax	Maximum characters allowed on input	20
venmVisualType	Visual control with which.data is rendered	enmBoolean. TextBox

---

clsUDBitmap

```
Inherits Protronics.User.clsUDType
```

This class manages an image data type, rendering it as a control whose state represents the value of the underlying data.

Visual Types

The only visual type implemented for the clsUDBitmap is currently a Windows bitmap. It also supports custom controls.

Formatting properties

The formatting properties are not implemented for clsUDBitmap.

Public Properties

bmpPicture as Bitmap

This property exposes the displayed bitmap

Public Methods

New (venmImage, vstrCaption)
New (venmImage, vstrCaption, vstrTag, [venmVisualType])

Parameter	Meaning	Default
venmImage	Predefined icon in the core pod	
vstrCaption	Text associated with data object.	“”
vstrTag	Unique identifier for data control	vstrCaption
venmVisualType	Visual control with which.data is rendered	enmImage.Bitmap

New (vintImage, vobjUserpod)
New (vintImage, vobjUserpod, vstrCaption, [vstrTag], [venmVisualType])

Parameter	Meaning	Default
vintImage	Index of image in the user pod	
vobjUserPod	User pod of type IPod	
vstrCaption	Text associated with data object.	“”
vstrTag	Unique identifier for data control	vstrCaption
venmVisualType	Visual control with which.data is rendered	enmImage.Bitmap

New (vbmpBitmap)
New (vbmpBitmap, vstrCaption, [vstrTag], [venmVisualType])

Parameter	Meaning	Default
vbmpBitmap	User supplied bitmap of type Bitmap	
vstrCaption	Text associated with data object.	“”
vstrTag	Unique identifier for data control	vstrCaption
venmVisualType	Visual control with which.data is rendered	enmImage.Bitmap

---

clsDataButton

```
Inherits Protronics.User.clsUDType
```

This class manages an abstraction of a command control (button), rendering it as a Windows button. The button raises a evtCommandClicked event itself, and causes its containers (to any level) to raise this event.

Visual Types

The only visual type implemented for the clsUDButton is currently a Windows Button. It also supports custom controls.

Public Properties

objCommandData()

The clsDataControl object associated with the button. This is passed in the event arguments when the button is clicked.

objValidatedControl

Control which is to be validated when button is clicked

objPortedButtonControl ()

This property, of type IPortedButton control is set to an object that implements the interface. It is used if venmVisualType is set to custom.

Formatting properties

For the Windows button visual type, sglDataWidth is the width, in characters, of the button. The property sglDataHeight determines the height, in characters, of the textbox. The property sglDataWidth defaults to 8. The property sglDataHeight defaults to 1. The respective maximum properties, sglDataWidthMax and sglDataHeightMax place limits on these parameters and default to 200 and 20 respectively.

Public Methods

New (vstrCaption)

New (vstrCaption, vblnAddPad, [vstrTag], [vsglDataWidth], [vsglDataHeight], [vobjCommandData], [venmVisualType])

Parameter	Meaning	Default
vstrCaption	Text associated with data object. This is displayed on the button.	
vblnAddPad	If true, space is added around the control	
vstrTag	Unique identifier for data control	vstrCaption
vsglDataWidth	Button width in characters. See Formatting Properties	8
vsglDataHeight	Button height in characters. See Formatting Properties	1
vobjCommandData	clsDataControl which is sent when the button raises a CommandClicked event.	Nothing
venmVisualType	Visual type to use	enmButton.WindowsButton

New (vstrCaption, vblnAddPad, vstrTag, vdlgHandler, [vsglDataWidth], [vsglDataHeight], [venmVisualType])

Parameter	Meaning	Default
vstrCaption	Text associated with data object. This is displayed on the button.	
vblnAddPad	If true, space is added around the control	
vdlgHandler	Delegate to be called when the button is clicked. This delegate has no arguments.	
vsglDataWidth	Button width in characters. See Formatting Properties	8
vsglDataHeight	Button height in characters. See Formatting Properties	1
venmVisualType	Visual type to use	enmButton.WindowsButton

AddClickHandler (vdlgHandler)

Adds a handler for this button. vdlgHandler takes no arguments.

clsUDTypeList (abstract sub class)

Heirs of this class represent lists of data.

Public Properties

intSelection ()
The first selected integer of an array

aintSelection ()
The list of selected integers of an array

objPortedListControl
The IPortedListControl object held by this UDType.

clsUDStringList

```
Inherits Protronics.User.clsUDTypeList
```

This class manages a string list data type, rendering it as a control whose state represents the value of the underlying data.

Visual Types

The visual types implemented for the clsUDString are as follows:

Type	Description
ListBox	Windows list box
ListBoxMultiSelect	Windows list box with multiple selection
CheckedListBox	Windows checked list box
CheckedListBoxMultiSelect	Windows checked list box with multiple selection
Combobox	Windows combo box
CheckChoice	List of checkboxes with captions as strings
RadioChoice	List of radio buttons with captions as strings
Custom	User supplied control

Public Properties

See Data Control Value Properties

objGamut as Object

Overriden from clsDataControl. This property exposes the range of values allowed by the object as an array of String objects cast to type Object.

Formatting properties

The controls are sized by default to the initial data, unless parameters sglDataWidth or sglDataHeight are modified from the default of -1. For the ComboBox sglDataHeight is always set to 1. For Windows listbox types, sglDataWidth is the width, in characters, of the listbox contents while sglDataHeight is the number of rows displayed. The properties sglDataWidthMax and sglDataHeightMax may be used as a maximum value for width and height when these parameters are set automatically. They default to 100 characters for sglDataWidth and 20 rows for sglDataHeight.

The formatting parameters are not implemented for composite types CheckChoice and RadioChoice.

Public Methods

New(vstrCaption, vstrList, venmVisualType)

Constructor taking caption, CSV list and visual type. See following constructor for defaults.

New(vstrCaption, vstrList, [vintData], [vstrTag], [venmVisualType])

Parameter	Meaning	Default
vstrCaption	Text associated with data object.	
vstrList	CSV list of strings. These are trimmed of whitespace.	
vstrTag	Unique identifier for data control	vstrCaption
vintData	Zero – based index of selected element.	0
venmVisualType	Visual control with which.data is rendered	enmStringList.ListBox

New(vstrCaption, rastrList(), [vintData], [vstrTag], [venmVisualType])

Parameter	Meaning	Default
vstrCaption	Text associated with data object.	
rastrList()	Array of strings to be displayed	
vstrTag	Unique identifier for data control	vstrCaption
vintData	Zero – based index of selected element.	0
venmVisualType	Visual control with which.data is rendered	enmStringList.ListBox

New(vstrCaption, rastrList(), vaintData(), [vstrTag], [venmVisualType])

Parameter	Meaning	Default
vstrCaption	Text associated with data object.	
rastrList()	Array of strings to be displayed	
vstrTag	Unique identifier for data control	vstrCaption
vaintData()	Zero – based indexes of selected element.	0
venmVisualType	Visual control with which.data is rendered	enmStringList.ListBoxMulitSelect

---

clsUDTypeTable (abstract sub class)

Heirs of this class represent the ADO.Net DataTable.

---

clsUDViewSet

```
Inherits Protronics.User.clsUDTypeTable
```

This class manages an ADO.Net DataTable, displaying it as a ListView

Visual Types

The only visual type currently implemented for clsUDViewSet is the Windows ListView control.

Public Properties

objGamut as Object

Overriden from clsDataControl. This property exposes the range of values allowed by the object as an array of String objects cast to type Object.

See Data Control Value Properties

Formatting properties

If the parameters sglDataWidth and sglDataHeight are not altered from their defaults of –1, then the listview width and height default to the size of the data, with the width dependent on column header width and the height set to the number of rows. If the parameters are set, then sglDataWidth sets the column width, in characters, while sglDataHeight sets the number of rows. In either case, the parameters sglDataWidthMax and sglDataHeightMax set a maximum value on these properties. They default to 200 characters and 20 rows respectively.

The format string, provided to the constructor, allows the definition of individual column widths, in characters. The string consists of a comma separated lists of header fields followed by a colon and then the length. For example, the string

name,address:30,postcode,tel:20

defines fields name and address as length 30, and fields postcode and tel as length 20. A missing column, or a value of –1 will cause the header to be sized to the value sglDataWidth. If sglDataWidth is –1, then the header is sized to the width of the column header.

New(vstrCaption, vdtaTable, [vstrTag], [vstrFormat], [venmVisualType])

Parameter	Meaning	Default
vstrCaption	Text associated with data object.	
vdtaTable	DataTable to be displayed	
vstrTag	Unique identifier for data control	vstrCaption
vstrFormat	Format string determining column width.	“”
venmVisualType	Visual control with which.data is rendered	enmStringList.ListView

Core Classes

Core objects are pre-built objects which combine the simpler, or more general, objects described above into commonly needed user interface elements.

---

clsCoreStringBlock

```
Inherits Protronics.User.clsDataBlockList
```

This class manages a list of text boxes of a fixed length. The constructor takes a comma-separated list of field names which act both as captions and as identifiers for the fields. Field contents are set and got by using the inherited `strValue` property.

Public Methods

New(vstrBlockCaption, vstrFieldList)
New(vstrBlockCaption, vstrFieldList, vintWidth)
New(vstrBlockCaption, vstrFieldList, [vintWidth], [venmBorder],[vstrDelimiter])

Parameter	Meaning	Default
vstrBlockCaption	Caption appearing above panel	“”
vstrFieldList	Comma-separated list of field names which act as captions	
vintWidth	Maximum number of characters in fields	30
venmBorder	Border style	Rectangular
vstrDelimiter	Delimiter used in vstrFieldList	Comma

Usage notes:

This class is demonstrated by the example *clsSimpleForm*.

---

clsCoreCheckBlock

```
Inherits Protronics.User.clsDataBlockList
```

This class manages a list of text boxes of arbitrary length. The constructor takes a comma-separated list of field names which act both as captions and as identifiers for the fields. Field contents are set and got by using the inherited strValue property.

Public Properties

intValue As Integer

Overriden from clsDataControl. This property gets or sets the state of the checkboxes using the value as a binary coded decimal representation. The checkbox added first is the least significant bit.

Public Methods

New(vstrCaption, vstrOptionsCSV, venmBorder)

vstrOptionsCSV is a CSV list of Strings representing checkbox captions. See the table below for definitions of the other constructors.

New(vstrCaption, venmBorder)
New(vstrCaption, venmBorder, vastrOptions(), [venmCompaction])

Parameter	Meaning	Default
vstrCaption	Caption appearing above checkboxes.	
venmBorder	Border style of type enmBorderStyle	
vastrOptions	Array of Strings representing checkbox captions	`Nothing`
venmCompaction	Control for vertical spacing of checkboxes	`enmCompactionSet.Regular`

AddChoice(vstrCaption)
AddChoice(vstrCaption, vblnValue)

Adds a new checkbox to the container.

Data Control Value Properties

All properties are made available at the level of Data Control so that they may be overriden by the assembly user to allow a custom Data Control to return any type of value. This approach also allows 'late referencing' of properties of a Data Control of unknown type returned from, for example, a search. If a property which is not fully implemented is called, then the assembly will thow an exception. The following tables define which properties are implemented for different UDTypes. Blue denotes read/write properties; others are read only.

Scalar UDTypes

	UDBoolean	UDInteger	UDFloat	UDDate	UDString
blnValue	Boolean				
intValue		Integer	Integer part conversion		
dblValue		Double conversion	Double		
datValue				Date	
strValue	String conversion	String conversion	String conversion	String conversion	String
objValue	Boolean	Integer	Double	Date	String

List UDTypes

	UDStringList	UDViewSet
strValue	First selected string	
astrValue()	Array of selected strings	
colValue		
objValue	Same as aintSelection	Same as aintSelection
objGamut	String()	DataTable
intSelection	First selected index	First selected index
aintSelection()	List of selection indexes	List of selection indexes

For a List UDType, the property objValue returns the selection state of the objValue object. The tables shows the underlying type of the data, although the property is always rendered as an Object type. Property objValueUI is the value displayed in the UI, and may temporarily be different from objValue.

The property objGamut is implemented only in list UDTypes and and exposes the range of values that a UDType represents – for example a list of strings or a DataTable.

Menus

Menus may be added to any object of type clsDataControl. There are two kinds of menu; *main* and *context*, where main is the sited at the top of a form and context menus are those which pop up when the user right-clicks on a control. A main menu may be added to a form or another main menu; in the latter case, it will appear as a submenu. A context menu may be added to any kind of control except a main menu.

Menus are represented by the class clsDataMenu, which inherits from clsDataContainer.

Example code for menus may be found in the custom control clsSimpleFormMenu.

Pods

Pods hold bitmap and icon resources for Coderform applications. The Coderform dll contains a core pod and additional pods are available as separate dlls: each implements the IPod interface. The core pod is exposed by the CoderFormManager class as property objCorePod.

The DataContainer class holds a WriteOnly property which defines the current user pod. This may be set to the core pod, or to an externally created pod. External pods are provided in separate DLLs and are instantiated in the main application.

Duties and Constraints

Duties are actions that a Data Control performs on a specific occasion, or event. For example, to have the contents of a textbox selected when the user tabs into it, the assembly user may define a duty on the corresponding Data Control.

Constraints are limits on the values of a Data Control as for example, ensuring that a value represents a date in the future. If the value displayed in the UI is outside these limits, then the constraint is not satisfied. Depending on the result of a constraint – satisfied or not - one or more actions may be performed. The actions are associated with a duty.

A Data Control maintains a collection of duties. Each duty may or may not have a constraint associated with it. Thus, constraints are mapped to DataControls through duties. At certain event times (e.g. display, mouse enter) the Data Control checks to see if any duties are bound to that event time. A Duty which is bound to a validating event will prevent the closing of a form when the constraint is not satisfied, when the OK button is pressed.

If a duty has a binding to the validation event, then it is termed a *Validating Duty*; otherwise it is called a *Non-validating Duty*. Typically, validating duties are used for form validation while non validating duties manage the UI state.

Duty Binding Times - enmDutyBinding

Parameter	Meaning
Always = -1	Always run duty
None = 0	No specified binding
Creation = 1	On form creation
Display = 2	On form display
DataChanged = 4	As user is typing
DataEntered = 8	When user enters the control
DataValidating = 16	When user leaves the control
ReturnPressed = 32	When user clicks the Enter button
Validation = 64	When user clicks a form button or menu

Examples of Duties

These examples are coded in the constraints section of the demo program.

1. To disable objDataControl1 on program startup

```
objDataControl1.objAddDuty(enmDutyBinding.Creation,
                           enmControlState.NotEnabled)
```

2. To highlight objDataControl1 when the user tabs to it

```
objDataControl1.objAddDuty(enmDutyBinding.DataEntered,
                           enmControlState.Selected)
```

3. To mark field when user clicks away or on the OK button if the field does not contain a valid UK postcode.

```
objIsPostcode = New
                clsConstraintString(enmCONSTRAINT_STRING.POSTCODE_UK
                )


objDataControl1.objAddDuty(objIsPostCode,
                           enmDutyBinding.DataValidating Or
                           enmDutyBinding.Validation,
                           enmControlState.MarkAlert)
```

4. To show a message when the user clicks on a button whose tag is set to "Apply" if the field does not contain a valid UK postcode

```
objDataControl1.objAddDuty(objIsPostcode,
                           enmDutyBinding.Validation,
                           enmActionType.MessageAlert).strHotTag=
                           "Apply"

mobjApply.objValidatedControl = objDataControl1
```

5. To enable objDataControl2 whenever objDataControl1 has a value of 4

```
objConstraint = New clsConstraintNumeric(4, 4)

objDataControl1.objAddDuty(objConstraint
                           enmDutyBinding.Display Or
                           enmDutyBinding.DataValidating,
                           enmControlState.Enabled,objDataControl2)
```

6. To enable a set of controls when objDataControl1 has a value of 4

```
objDuty = objDataControl1.objAddDuty(objConstraint,
                                     enmDutyBinding.Display Or
                                     enmDutyBinding.DataChanged,
                                     enmControlState.Enabled,
                                     objDataControl2)

objDuty.AddAction(objDataControl3, enmControlState.Enabled)
```

7. To constrain a date to be on a weekend

```
objConstraint = New clsConstraintDate(enmCONSTRAINT_DATE.WEEKDAY, ,
                                      True)
```

8. To constrain a date to be in the future

```
objConstraint = New clsConstraintDate(Now)
```

Duty classes

Duties are managed by a number of classes whose specification is given below.

---

clsDuty

This class represents a duty.

Public Properties

intBinding As Integer

Logical AND of the enumeration value defining when the duty is to be performed.

strHotTag As String

This tag, if set, is compared with the tag of a clsDataButton or clsDataMenu which causes validation of an object holding this duty. If it is set, the validation occurs only if the tags match, regardless of case.

objConstraint As clsConstraint

Constraint object associated with this duty.

strHotTag As String

If not empty, then a validation duty will be run only if the event is raised by a Data Control with this tag.

Public Methods

New(venmBinding, vobjConstraint)

This takes a binding – the enumeration defining when the duty is to be performed, and a constraint.

AddAction(vobjAction)

Adds an action to the duty, taking a clsDutyAction object as parameter.

AddAction(vobjDataControl, venmActionType, [vstrMessage], [venmActOn])

Adds an action to the duty. The target of the action is vobjDataControl, while venmActionType is the type of action to perform.

---

clsDutyAction

This class represents an action.

Public Properties

strMessage As String

Message associated with constraint, as for example a dialog message reporting failure.

enmActOnState As enmActOn

Enumeration defining whether the action takes place when the constraint passes or fails.

enmAction As enmControlState

Enumeration defining the state change of a control.

objTarget As clsDataControl

Data Control to which to apply action.

Public Methods

New(vobjTarget,venmAction, vstreMessage, [venmActOn])

Parameter	Meaning	Default
vobjTarget	Data Control on which to to perform action	
venmAction	Type `enmActionType`. Action to perform.	
vstrMessage	Message associated with the action. Applies only to some actions.	
venmActOn	Enumeration of type enmActOn which detemines whether action takes place on constraint pass or fail.	`enmActOn.ActionDependent`

Constraint classes

Constraints allow subsets of data to be defined as valid or invalid by applying a function to the data. The function may be predefined by CoderForm or defined by the application writer.

The clsConstraint has children of Integer, Numeric, String and Custom constraints. These are added to the clsDataControl object, along with a defined binding which determines when the constraint is applied, an action which determines what the constraint does, and a target clsDataControl. These provide validation feedback respectively: as the user types in the field; when the user leaves the field; when the user presses OK.

If the paramater vstrMessage is left blank, then the default message is used.

---

clsConstraint (base class)

This class represents a general constraint.

Public Methods

New()

Parameterless constructor. This is typically used implicitly by child classes.

New(vobjValue)

Constraint succeeds if the source Data Control has the given value.

New(vstrMessage, vblnDelegateValidate)

This constructor allows the definition of a delegate function to perform validation on the data with which the constraint is associated. The delegate function has the interface as follows:

```
Public Delegate Function blnValidate(ByVal vobjValue As Object) As Boolean
```

It should return true if the data is valid.

---

clsConstraintNumeric

```
Inherits Protronics.User.clsConstraint
```

This class represents a constraint on a numeric value.

Public Methods

New(vintValue)

This constructor defines an allowed range of a single integer

New(vintMin, vintMax, [vstrMessage], [vblnInvert])

This constructor allows the definition of an integer range.

Parameter	Meaning	Default
vintMin	Minimum of valid range.	
vintMax	Maximum of valid range.	
vstrMessage	Message displayed when validation fails.	Standard message
vblnInvert	If True, the constraint to use the inverse of the defined valid set.	False

New(vdblMin, vdblMax)
New(vdblMin, vdblMax, vstrMessage, [vblnInvert])

This constructor allows the definition of a range of real numbers.

Parameter	Meaning	Default
vdblMin	Minimum of valid range.	
vdblMax	Maximum of valid range.	
vstrMessage	Message displayed when validation fails.	Standard message
vblnInvert	If True, the constraint to use the inverse of the defined valid set.	False

New(venmConstraintType, [vstrMessage])

This constructor allows the definition of a range of real numbers using a predefined constraint.

Parameter	Meaning	Default
venmConstraintType	Predefined constraint.	
vstrMessage	Message displayed when validation fails.	Standard message

The predefined constraint may be either enmCONSTRAINT_NUMERIC. POSITIVE_INTEGER or enmCONSTRAINT_NUMERIC.POSITIVE_REAL.

New(vstrMessage, vblnDelegateValidate)

This constructor allows the definition of a delegate function to perform validation on the data with which the constraint is associated.

Parameter	Meaning	Default
vstrMessage	Message displayed when validation fails.	
vblnDelegateValidate	Delegate validation function.	

The interface of the delegate function is as follows:

```
Public Delegate Function blnValidateNumeric(ByVal strValue As String)
As Boolean
```

The delegate takes a string, rather than an integer as argument, to allow more detail in any failure message when non numeric data is supplied. It should return true if the data is valid.

---

clsConstraintDate

This class represents a constraint on a date value.

Public Methods

New (venmConstraintType)
New (venmConstraintType, vstrMessage, [vblnInvert])

These constructors create a predefined constraint that sets a range of values for the date it is associated with.

Parameter	Meaning	Default
venmConstraintType	Predefined constraint	
vstrMessage	Message displayed when validation fails	Standard message
vblnInvert	Generates inverse set for constraint	False

One predefined constraint is implement. The constraint WEEKDAY ensures the day selected is Monday to Friday.

New(vdatStartDate, vdatEndDate)
New(vdatStartDate, vdatEndDate, vstrMessage, [vblnInvert])

These constructors create a constraint that sets a range of values for the date it is associated with.

Parameter	Meaning	Default
vdatStartDate	Minimum allowed length of string	
vdatEndDate	Maximum allowed length of string	
vstrMessage	Message displayed when validation fails	Standard message
vblnInvert	If True, the constraint to use the inverse of the defined valid set.	False

---

clsConstraintString

```
Inherits Protronics.User.clsConstraint
```

This class represents a constraint on a string value.

Public Methods

New(vintMinLength, [vintMaxLength], [vstrMessage], [vblnInvert])

This constructor creates a constraint that sets a range of lengths for the string it is associated with.

Parameter	Meaning	Default
vintMinLength	Minimum allowed length of string	
vintMaxLength	Maximum allowed length of string	
vstrMessage	Message displayed when validation fails	
vblnInvert	If True, the constraint to use the inverse of the defined valid set.	False

New(venmConstraintType, [vstrMessage])

This constructor allows a predefined string constraint to perform validation on the data with which the constraint is associated.

Parameter	Meaning	Default
venmConstraintType	Predefined constraint function	
vstrMessage	Message displayed when validation fails	Depends on constraint

The following predefined constraint types are available.

TELEPHONE_INTL	International telephone number
TELEPHONE_US	US telephone number
EMAIL_WEAK	Email address – weak checking
EMAIL_STRONG	Email address – strong checking
POSTCODE_UK	UK Postcode
ALPHA	a to z and A to Z
NUMERIC	A number
ALPHANUMERIC	An alphanumeric character
ZIPCODE	US Zipcode, 5 and 9 digit
USERNAME_WEAK	User name – 6 to 16 letters
PASSWORD_WEAK	Password – 6 to 16 non-blanks
URL	URL

New(vstrMessage, vstrPattern)

This constructor allows the definition of a regular expression to perform validation on the data with which the constraint is associated.

Parameter	Meaning	Default
vstrMessage	Message displayed when validation fails	
vstrPattern	Regular expression. If the expression matches the value on which the constraint is defined, then it is validated.	

New(vblnDelegateValidate, [vstrMessage])

This constructor allows the definition of a delegate function to perform validation on the data with which the constraint is associated.

Parameter	Meaning	Default
vblnDelegateValidate	Delegate validation function.	
vstrMessage	Message displayed when validation fails.	Standard message

The interface of the delegate function is as follows:

```
Public Delegate Function blnValidateString(ByVal strValue As String)
As Boolean
```

It should return true if the data is valid.

---

clsConstraintStringList

```
Inherits Protronics.User.clsConstraint
```

This class represents a constraint on a string list UDType.

Public Methods

New(vintValue)
New(vintLowerValue, vintUpperValue)

This constructor requires the stringlist to have the defined selection value set to the value or within the limits provided.

New(venmConstraintType, vstrMessage)

This constructor allows a predefined string constraint to perform validation on the data with which the constraint is associated.

Parameter	Meaning	Default
venmConstraintType	Predefined constraint function	
vstrMessage	Message displayed when validation fails	Depends on constraint.

New(vintMinLength, vintMaxLength, vstrMessage, [vblnInvert])

This constructor creates a constraint that sets limits on the number of strings selected.

Parameter	Meaning	Default
vintMinLength	Minimum number of strings selected.	
vintMaxLength	Maximum number of strings selected.	
vstrMessage	Message displayed when validation fails	
vblnInvert	If True, the constraint to use the inverse of the defined valid set.	False

New(vblnDelegateValidate, vstrMessage)

This constructor allows the definition of a delegate function to perform validation on the data with which the constraint is associated.

Parameter	Meaning	Default
vblnDelegateValidate	Delegate validation function.	
vstrMessage	Message displayed when validation fails.	

The interface of the delegate function is as follows:

```
Public Delegate Function blnValidateStringList(ByVal aintValue() As
Integer) As Boolean
```

It should return true if the data is valid.

Visual Classes

Every Data Control keeps a reference to a clsVisual object. The Visual classes handle format and positioning functions for the clsDataControl hierarchy. The assembly user typically interacts with the data classes, rather than the visual classes. However, for making changes to the UI format, it is necessary to access some properties of visual classes.

Object Model

clsVisual (abstract base class)

clsVisualContainer (abstract class)

clsVisualBlock
clsVisualLeaves

---

clsVisual

The class clsVisual provides a virtual friend function for generating a control, including any contained controls. It also provides virtual functions for updating the GUI state and selecting the control. It holds public properties for the defining the whitespace (or padding) added to the control boundaries. Only the formatting properties are exposed.

Public Properties

intLeftPad ()
intRightPad ()
intTopPad ()
intBottomPad ()

These format properties determine the padding added to the external boundary of the Windows control. Compare with the margin properties of the class clsVisualContainer.

blnCR ()
This property, when true, causes the control to start a new packing line when it is packed in a block. If the packing direction is left to right or right to left, then the new line is placed below the last line. If the packing direction is top to bottom or bottom to top, then the new line is placed to the right of the last line

clsVisualContainer

```
Inherits clsVisual
```

The class clsVisual provides additional formatting properties that are specific to containers.

Public Properties

intLeftMargin ()
intRightMargin ()
intTopMargin ()
intBottomMargin ()

The foregoing format properties determine the margin in pixels added to a container. This is the distance between the bounding box of the contained controls, and the edge of their container.

intGridWidth ()
intGridHeight ()

These properties are the number of pixels added to the left and top offsets respectively of a control when grid packing is used, that is, when intListLength is set.

intListLength ()

This property is the maximum length of a packing line for controls added to a clsDataBlock.

IPortedControl Interface

This interface defines a standard behaviour on a class which generates a Windows control. It, and its heirs, define the events, properties and methods that a class must implement in order to visualise data for a clsUDType object.

Figure 2. IPortedControl Interface Type Hierarchy

IPortedControl

- IPortedBitmapControl
- IPortedBooleanControl
- IPortedButtonControl
- IPortedDateControl
- IPortedFloatControl
- IPortedIntegerControl
- IPortedStringControl
- IPortedStringListControl
- IPortedTableControl

The IPortedControl interface is implemented internally by the abstract class **clsPortedControl** and heirs of clsPortedControl implement one of the subtypes of IPortedControl.

IPortedControl

Properties

Formatting properties

sglDataWidth	As Single
sglDataWidthMax	As Single
sglDataHeight	As Single
sglDataHeightMax	As Single

The formatting properties control the dimensions for the control. The implementation of the control determines their meaning.

strCaption As String

Any caption associated with the control

ctlWindowsControl As Control

The Windows control associated with this control. It should be created when the control is instantiated.

objValue As Object

The data value associated with the control

Methods

ctlGenerateControl

This method should complete any rendering of the data value to the Windows control.

Events

evtDataEntered

This event should fire when the user selects a data field to begin an edit.

evtDataChanged

This event should fire when the data is changed in real time – for example as each letter is added to the string in a text box.

evtDataValidating

This event should fire when the data input is complete – for example when the user tabs away from a text box

evtDataChosen

This event should fire when the user explicitly chooses the data – for example when an item on a list is double-clicked

evtReturnPressed

This event should fire when the user presses the Return (or Enter) key.

Plugin Controls

The user may create custom classes for visualisation of data either by implementing an IPortedControl type or by inheriting from clsPortedControl. The former method would be necessary if the custom class inherits from

another class, typically Windows.Forms.Control. An example of a custom control, also used internally by CoderForm is clsSeparator (used for the display of a title bar). This is demonstrated as a custom control in **SimplePluginControl.vb** as follows:

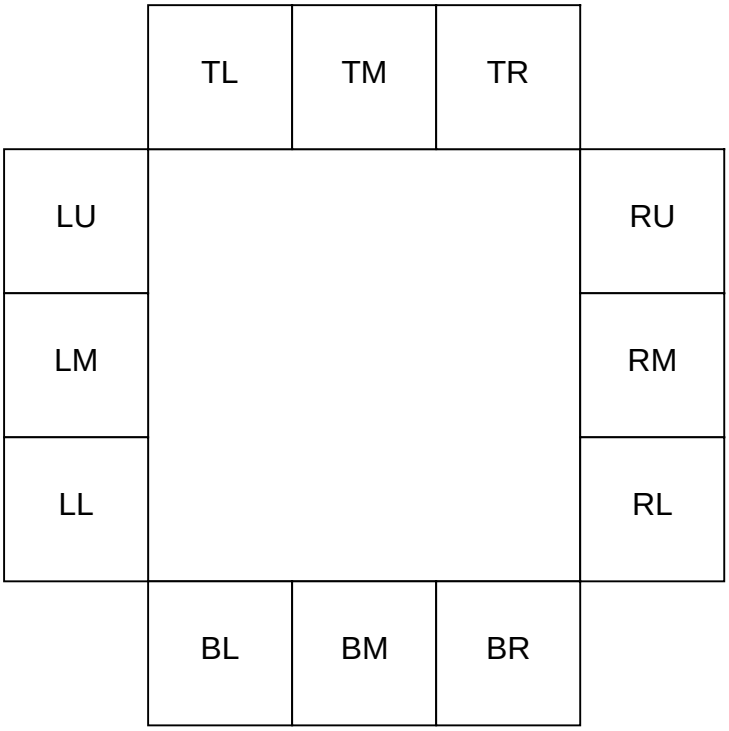
Figure 3. Example code for demonstrating plugin

```
' Example code for demonstrating plugin

ctlPlugin = New clsSeparatorControl("Title Bar", 400)

objPluginControl = New clsUDString("", enmString.Custom)

objPluginControl.objPortedControl = ctlPlugin
```

Note that the plugin control is created using the visualisation type enmString.Custom to prevent an internal control being created. The clsUDType property objPortedControl is then set to the externally created IPortedControl type.

Appendix 1.

Enumeration enmPosition



TopLeft TopMiddle TopRight

LeftUpper
LeftMiddle
LeftLower

RightUpper
RightMiddle
RightLower

BottomLeft BottomMiddle BottomRight

Note that the parameter *vblnInside* to the clsDataControl method *Locate*, if True, will move the located control inside the fixed control.

Appendix 2. Documentation note

Optional parameters and Defaults

VB.Net supports optional parameters whereas C# does not. For this reason, methods, in particular constructors, are overloaded wherever possible to provide a simple interface. So as to provide the benefit to VB.Net clients, parameters are also made optional where appropriate and provided that they are not required to distinguish between overloaded methods. In the documentation, where methods are shown on successive lines the defaults described apply to all the listed constructors.

Type Self Documenting

Types in the source code and this documentation are denoted by a Hungarian prefix, notwithstanding Microsoft's recommendation for .Net. Arrays are named by prepending "a" to the type prefix.

Prefix	Type
bln	Boolean
sgl	Single
dbl	Double
int	Integer
str	String
dtm	DateTime
dta	DataTable
enm	Enumeration
col	Collection
cls	Class
obj	Object
ctl	Control or clsVisual