Appendix B – Exercise Solutions (Example Code)

This appendix contains examples of code that provide solutions to each of the exercises.

ⓘ *Local variables have been used to shorten lines to avoid "word wrap" and improve clarity. Look for the definition of these variables within the examples and where/how they are used.*

Appendix B1 - Example c2ex1.pmlfrm

```
-------------------------------------------------------------------------------------------
--                                                                                       --
-- (c) Copyright  2010 to Current Year  AVEVA Solutions Limited                          --
--                                                                                       --
-- File:            c2ex1.pmlfrm                                                         --
--                                                                                       --
-- Description:                                                                          --
--   Example form solution to Exercise 1                                                 --
--                                                                                       --
--   Supplied to support AVEVA training course TM1881 - PML: Form Design                 --
--                                                                                       --
-------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------
-- Form definition                                                                       --
-------------------------------------------------------------------------------------------
setup form !!c2ex1 dialog resizeable

  -- Set the form title and the initialisation call
  !this.formTitle = |Example Form - Exercise 1|
  !this.initcall  = |!!this.init()|

  -- Set the default path for gadget layout
  path down

  -- Input frame (LHS)
  frame .inputFrame |Inputs| at x 0 anchor T+L+B

    -- Top Frame - Temperature Input
    frame  .tempInputFrame |Temperature conversion (Input)| at x 1
      text .tempInput |Temperature| call |!!this.temperatureConvert()| width 10 is REAL
      frame .tempChoiceFrame panel at xmax.tempInput ymin.tempInputFrame + 0.5
        rToggle .celsius |°C| tagwid 3 at xmax.tempInput + 3 ymin.tempInput
        rToggle .fahrenheit |°F| tagwid 3 at xmax.celsius ymin.tempInput
      exit
    exit

    -- Middle Frame - Temperature Range
    frame .tempRangeFrame |Temperature Range| at x 1 width.tempInputFrame
      text .minimum |Minimum    | call |!this.check(1)| at x 1 width 10 is REAL
      text .maximum |Maximum    | call |!this.check(2)| at x 1 width 10 is REAL
      !format = |format !!INTEGERFMT|
      text .step    |Step Size  | call |!this.check(0)| at x 1 width 10 is REAL $!format
      !bPos = |xmax.step + 1 ymin.step|
      button .fillFahrenheit linklabel |Fill with °F to °C >>|  at $!bPos wid 12
      !bPos = |xmin.fillFahrenheit ymin.fillFahrenheit - size|
      button .fillCelsius    linklabel |Fill with °C to °F >>|  at $!bPos wid 12
    exit

    -- Lower Frame - Temperature Split
    frame .stringSplitFrame |Temperture Split| anchor L+B+R at x 1 width.tempInputFrame
      text .stringInput |Input      | width 24 is STRING
      text .delimiter   |Delimiter  | width 10 is STRING
      !bPos = |xmax.delimiter + 1 ymin.delimiter|
      button .split linklabel |Split temperature >>| at $!bPos  wid 12
    exit
  exit

--(Continues on next page)
```

Fix1: U not valid

Fix2: REAL with one L

```
  -- Results Frame (RHS)
  frame .results |Results| at xmax + 1 y 0 anchor ALL

    -- Top frame - Temperature ouput
    !fSize = |width.tempInputFrame height.tempInputFrame|
    frame  .tempOutputFrame |Temperature conversion (Output)| anchor L+T+R at x 1 $!fSize
      text .tempOutput |Temperature| call || width 10 is REAL
      para .unit at xmax.tempOutput ymin.tempInput text || width 5
    exit

    -- Middle frame - Temperature results
    !fSize = |width.tempRangeFrame height.tempRangeFrame|
    frame .tempRangeOutFrame |Temperature Results| anchor L+T+R at x1 $!fSize
      list .tempList dock fill width 1 height 1
    exit

    -- Lower frame - Split temperature
    !fPos  = |x1 ymin.stringSplitFrame|
    !fSize = |width.stringSplitFrame height.stringSplitFrame|
    frame .stringSplitOutFrame |Temperature Split Result| anchor L+B+R at $!fPos $!fSize
      text .number |No. of Temp| width 10          is REAL format !!INTEGERFMT
      text .result |Result     | width.stringInput is STRING
    exit
  exit
exit
-------------------------------------------------------------------------------
-- Constructor Method - Set the callbacks on the form                        --
-------------------------------------------------------------------------------
define method .c2ex1()
  !this.tempChoiceFrame.callback = |!this.temperatureConvert()|
  !this.fillFahrenheit.callback  = |!this.fill('Fahrenheit')|
  !this.fillCelsius.callback     = |!this.fill('Celsius')|
  !this.split.callback           = |!this.split()|
endmethod
-------------------------------------------------------------------------------
-- Initialisation Method - Set the initial gadget values and run the methods --
-------------------------------------------------------------------------------
define method .init()
  !this.tempInput.val = 0
  !this.tempChoiceFrame.val = 1
  !this.minimum.val = 0
  !this.maximum.val = 100
  !this.step.val = 25
  !this.stringInput.val = |10°C/30°C/20°C/5°C|
  !this.delimiter.val = |/|
  !this.temperatureConvert()
  !this.fill(|Celsius|)
  !this.split()
endmethod
-------------------------------------------------------------------------------
-- Method .celsiusToFahrenheit(REAL)REAL - Convert the number to Fahrenheit  --
-------------------------------------------------------------------------------
define method .celsiusToFahrenheit(!celsius is REAL) is REAL
  !fahrenheit = !celsius * 1.8 + 32
  return !fahrenheit
endmethod
-------------------------------------------------------------------------------
-- Method .fahrenheitToCelsius(REAL)REAL - Convert the number to Celsius     --
-------------------------------------------------------------------------------
define method .fahrenheitToCelsius(!fahrenheit is REAL) is REAL
  !celsius = (!fahrenheit - 32 ) / 1.8
  return !celsius
endmethod
-------------------------------------------------------------------------------
-- Method .temperatureConvert() - Convert input temperature based on radio button --
-------------------------------------------------------------------------------
define method .temperatureConvert()
  if !this.tempChoiceFrame.val.eq(1) then
    !this.tempOutput.val = !this.CelsiusToFahrenheit(!this.tempInput.val)
    !this.unit.val = |°F|
  elseif !this.tempChoiceFrame.val.eq(2) then
    !this.tempOutput.val = !this.FahrenheitToCelsius(!this.tempInput.val)
    !this.unit.val = |°C|
  endif
endmethod

--(Continues on next page)
```

Fix3: Width+height at the end of

Fix4: Incorrect constructor name

A new init method applies default values to the form

Fix5: F and C the wrong way around

```
-------------------------------------------------------------------------------------
-- Method .fill(STRING) - Fill the temperature conversion list based on argument   --
-------------------------------------------------------------------------------------
define method .fill(!whichTemperature is STRING)
  !n = 0
  -- Loop through the supplied values and build array for the list
  do !temperature from !this.minimum.val to !this.maximum.val by !this.step.val
    !n = !n + 1
    !tempArray[!n][1] = !n.string()
    !tempArray[!n][2] = !temperature.string()
    !tempArray[!n][3] = |=|
    if !whichTemperature.eq(|Celsius|) then
      !fahrenheit = !this.celsiusToFahrenheit(!temperature)
      !tempArray[!n][4] = STRING(!fahrenheit,!!realFMT)
    else
      !celsius = !this.fahrenheitToCelsius(!temperature)
      !tempArray[!n][4] = STRING(!celsius,!!realFMT)
    endif
  enddo

  -- Set the list headings based on the argument
  if !whichTemperature.eq(|Celsius|) then
    !headings = |No. Celsius = Fahrenheit|
  else
    !headings = |No. Fahrenheit = Celsius|
  endif
  -- Display the collected data
  !this.tempList.setHeadings(!headings.split())
  !this.tempList.setRows(!tempArray)
endmethod
-------------------------------------------------------------------------------------
-- Method .check(REAL) - Based on the type of check, ensure the entered values are ok   --
-------------------------------------------------------------------------------------
define method .check(!flag is REAL)

  -- Check the step value, if = 0 - then make it 1
  if !flag.eq(0) then
    if !this.step.val.eq(0) then
      !this.step.val = 1
    endif

  -- For others....
  elseif !flag.eq(1).or(!flag.eq(2)) then
    -- Check that text boxes have values entered in them
    if !this.maximum.val.unset() then
      !this.maximum.val = !this.minimum.val + !this.step.val
    elseif !this.minimum.val.unset() then
      !this.minimum.val = !this.maximum.val - !this.step.val
    endif
    -- Check that the values aren't lower than absolute zero.
    if (!this.minimum.val.lt(-273)) then
      !this.minimum.val = -273
      if (!this.maximum.val.leq(!this.minimum.val)) then
        !this.maximum.val = -272
      endif
    elseif (!this.maximum.val.lt(-273)) then
      !this.minimum.val = -273
      !this.maximum.val = -272
    endif
    -- Check that the maximum value is larger than the minimum value
    if (!this.maximum.val.leq(!this.minimum.val)) then
      if !flag.eq(1) then
        !this.maximum.val = !this.minimum.val + !this.step.val
      elseif !flag.eq(2) then
        !this.minimum.val = !this.maximum.val - !this.step.val
      endif
    endif
  endif
endmethod


--(Continues on next page)
```

```
-------------------------------------------------------------------------------
-- Method .split() - Split the user entered string, convert and concatenate   --
-------------------------------------------------------------------------------
define method .split()
  -- Split the string
  !split = !this.stringInput.val.trim().split(!this.delimiter.val)
  !this.number.val = !split.size()
  !result = ||
  -- Loop through the split
  do !n index !split
    -- If it's Celsius....convert to Fahrenheit
    if !split[!n].substring(!split[!n].length()).upcase().eq(|C|) then
      !temp = !split[!n].substring(1, !split[!n].length() - 2).real()
      !result = !result & !this.celsiusToFahrenheit(!temp).string(!!INTEGERFMT) & |°F |
    else
      !temp = !split[!n].substring(1, !split[!n].length() - 2).real()
      !result = !result & !this.fahrenheitToCelsius(!temp).string(!!INTEGERFMT) & |°C |
    endif
  enddo
  -- Return the concatenated string back to the form
  !this.result.val = !result
endmethod
-------------------------------------------------------------------------------
-- End of Form definition and methods                                        --
-------------------------------------------------------------------------------
-- (c) Copyright  2010 to Current Year  AVEVA Solutions Limited              --
-------------------------------------------------------------------------------
```

New method to split the string

Appendix B2 - Example c2ex2.pmlfrm

```
-------------------------------------------------------------------------------------------
--                                                                                       --
-- (c) Copyright  2010 to Current Year  AVEVA Solutions Limited                          --
--                                                                                       --
-- File:            c2ex2.pmlfrm                                                         --
--                                                                                       --
-- Description:                                                                          --
--   Example form solution to Exercise 2                                                 --
--                                                                                       --
--   Supplied to support AVEVA training course TM1881 - PML: Form Design                 --
--                                                                                       --
-------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------
-- Form definition                                                                       --
-------------------------------------------------------------------------------------------
setup form !!c2ex2 dialog resizeable

  -- Set the form title and the initialisation call
  !this.formTitle = |Example Form - Exercise 2|
  !this.initcall  = |!!this.init()|
  -- Set the default path for gadget layout
  path down
  frame .tabset TABSET anchor all

    -- Input frame (LHS)
    frame .inputFrame |Inputs| at x 0 y 0 anchor all

      -- Top Frame - Temperature Input
      frame  .tempInputFrame |Temperature conversion (Input)| at x 1
        text .tempInput |Temperature| call |!!this.temperatureConvert()| width 10 is REAL
        frame .tempChoiceFrame panel at xmax.tempInput ymin.tempInputFrame
          rToggle .celsius |°C| tagwid 3 at xmax.tempInput + 3 ymin.tempInput
          rToggle .fahrenheit |°F| tagwid 3 at xmax.celsius ymin.tempInput
        exit
      exit

      -- Middle Frame - Temperature Range
      frame .tempRangeFrame |Temperature Range| at x 1 width.tempInputFrame
        text .minimum |Minimim    | call |!!this.check(1)| at x 1 width 10 is REAL
        text .maximum |Maximum    | call |!!this.check(2)| at x 1 width 10 is REAL

        !format = |format !!INTEGERFMT|
        text .step    |Step Size  | call |!this.check(0)| at x 1 width 10 is REAL $!format
        !bPos = |xmax.step + 1 ymin.step|

        button .fillFahrenheit linklabel |Fill with °F to °C >>|  at $!bPos wid 12
        !bPos = |xmin.fillFahrenheit ymin.fillFahrenheit - size|
        button .fillCelsius    linklabel |Fill with °C to °F >>|  at $!bPos wid 12
      exit

      -- Lower Frame - Temperature Split
      frame .stringSplitFrame |Temperture Split| anchor L+B+R at x 1 width.tempInputFrame
        text .stringInput |Input      | width 24 is STRING
        text .delimiter   |Delimiter  | width 10 is STRING
        !bPos = |xmax.delimiter + 1 ymin.delimiter|
        button .split linklabel |Split temperature >>| at $!bPos  wid 12
      exit
    exit

    -- Results Frame (RHS)
    frame .results |Results| at x0 y 0 anchor ALL

      -- Top frame - Temperature ouput
      !fSize = |width.tempInputFrame height.tempInputFrame|
      frame  .tempOutputFrame |Temperature conversion (Output)| anchor L+T+R at x 1 $!fSize
        text .tempOutput |Temperature| call || width 10 is REAL
        para .unit at xmax.tempOutput ymin.tempInput text || width 5
      exit

      -- Middle frame - Temperature results
      !fSize = |width.tempRangeFrame height.tempRangeFrame|
      frame .tempRangeOutFrame |Temperature Results| anchor L+T+R at x1 $!fSize
        list .tempList dock fill width 1 height 1
      exit
--(Continues on next page
```

Tabset addd

```
      -- Lower frame - Split temperature
      !fPos  = |x1 ymin.stringSplitFrame|
      !fSize = |width.stringSplitFrame height.stringSplitFrame|
      frame .stringSplitOutFrame |Temperature Split Result| anchor L+B+R at $!fPos $!fSize
        text .number |No. of Temp| width 10          is REAL format !!INTEGERFMT
        text .result |Result     | width.stringInput is STRING
      exit

    exit
  exit

  !pixmap = |pixmap width 16 height 16|
  !link   = |linklabel width 10|
  para   .paAccept at  xmin.tempInputFrame  ymax+0.5  anchor L+B $!pixmap
  button .lkAccept |Accept changes| at xmax.paAccept+1 ymin anchor L+B $!link
  button .lkDiscard |Discard changes | at xmax.tempInputFrame - size ymin anchor R+B $!link
  para   .paDiscard at  xmin.lkDiscard - 1.5 * size  ymin anchor R+B $!pixmap

  -- Form member to store the form values
  member .data is ARRAY

  -- MENU objects to be used as context menus on the list gadget
  !menu = !this.newMenu(|celsiusMenu|)
  !menu.add(|Callback|, |Switch to °F = °C|, |!this.fill('Fahrenheit')|)
  !menu = !this.newMenu(|fahrenheitMenu|)
  !menu.add(|Callback|, |Switch to °C = °F|, |!this.fill('Celsius')|)

exit
-------------------------------------------------------------------------------
-- Constructor Method - Set the callbacks on the form                        --
-------------------------------------------------------------------------------
define method .c2ex2()
  !this.results.callback =  |!this.tabCall(|
  !this.paAccept.addPixmap(!!pml.getPathName('accept.png'))
  !this.paDiscard.addPixmap(!!pml.getPathName('discard.png'))
  !this.lkAccept.callback = |!this.setData(1)|
  !this.lkDiscard.callback = |!this.setData(2)|
endmethod
-------------------------------------------------------------------------------
-- Initialisation Method - Set the initial gadget values and run the methods --
-------------------------------------------------------------------------------
define method .init()
  !this.tempInput.val = 0
  !this.tempChoiceFrame.val = 1
  !this.minimum.val = 0
  !this.maximum.val = 100
  !this.step.val = 25
  !this.stringInput.val = |10°C/30°C/20°C/5°C|
  !this.delimiter.val = |/|
  !this.setData(1)
endmethod
-------------------------------------------------------------------------------
-- Method .tabCall(GADGET, STRING) - Open callback on Results tab            --
-------------------------------------------------------------------------------
define method .tabCall(!gadget is GADGET, !type is STRING)
  -- If the results tab is shown, then run the form methods
  if !type.eq(|SHOWN|) then
    !this.temperatureConvert()
    !this.fill(|Celsius|)
    !this.split()
  endif
endmethod
-------------------------------------------------------------------------------
-- Method .celsiusToFahrenheit(REAL)REAL - Convert the number to Fahrenheit  --
-------------------------------------------------------------------------------
define method .celsiusToFahrenheit(!celsius is REAL) is REAL
  !fahrenheit = !celsius * 1.8 + 32
  return !fahrenheit
endmethod
-------------------------------------------------------------------------------
-- Method .fahrenheitToCelsius(REAL)REAL - Convert the number to Celsius     --
-------------------------------------------------------------------------------
define method .fahrenheitToCelsius(!fahrenheit is REAL) is REAL
  !celsius = (!fahrenheit - 32 ) / 1.8
  return !celsius
endmethod
--(Continues on next page)

-------------------------------------------------------------------------------
-- Method .temperatureConvert() - Convert input temperature based on radio button --
```

New menu objects

New Accept/Discard buttons

Open Callback on results tab

```
-------------------------------------------------------------------------------------------
define method .temperatureConvert()
  if !this.tempChoiceFrame.val.eq(1) then
    !this.tempOutput.val = !this.CelsiusToFahrenheit(!this.tempInput.val)
    !this.unit.val = |°F|
  elseif !this.tempChoiceFrame.val.eq(2) then
    !this.tempOutput.val = !this.FahrenheitToCelsius(!this.tempInput.val)
    !this.unit.val = |°C|
  endif
endmethod
-------------------------------------------------------------------------------------------
-- Method .fill(STRING) - Fill the temperature conversion list based on argument        --
-------------------------------------------------------------------------------------------
define method .fill(!whichTemperature is STRING)
  !n = 0
  -- Loop through the supplied values and build array for the list
  do !temperature from !this.minimum.val to !this.maximum.val by !this.step.val
    !n = !n + 1
    !tempArray[!n][1] = !n.string()
    !tempArray[!n][2] = !temperature.string()
    !tempArray[!n][3] = |=|
    if !whichTemperature.eq(|Celsius|) then
      !fahrenheit = !this.celsiusToFahrenheit(!temperature)
      !tempArray[!n][4] = STRING(!fahrenheit,!!realFMT)
    else
      !celsius = !this.fahrenheitToCelsius(!temperature)
      !tempArray[!n][4] = STRING(!celsius,!!realFMT)
    endif
  enddo
  -- Set the list headings based on the argument
  if !whichTemperature.eq(|Celsius|) then
    !headings = |No. Celsius = Fahrenheit|
      !this.tempList.setPopup(!this.celsiusMenu)
  else
    !headings = |No. Fahrenheit = Celsius|
      !this.tempList.setPopup(!this.fahrenheitMenu)
  endif
  -- Display the collected data
  !this.tempList.setHeadings(!headings.split())
  !this.tempList.setRows(!tempArray)
endmethod
-------------------------------------------------------------------------------------------
-- Method .check(REAL) - Based on the type of check, ensure the entered values are ok   --
-------------------------------------------------------------------------------------------
define method .check(!flag is REAL)

  -- Check the step value, if = 0 - then make it 1
  if !flag.eq(0) then
    if !this.step.val.eq(0) then
      !this.step.val = 1
    endif

  -- For others....
  elseif !flag.eq(1).or(!flag.eq(2)) then
    -- Check that text boxes have values entered in them
    if !this.maximum.val.unset() then
      !this.maximum.val = !this.minimum.val + !this.step.val
    elseif !this.minimum.val.unset() then
      !this.minimum.val = !this.maximum.val - !this.step.val
    endif
    -- Check that the values aren't lower than absolute zero.
    if (!this.minimum.val.lt(-273)) then
      !this.minimum.val = -273
      if (!this.maximum.val.leq(!this.minimum.val)) then
        !this.maximum.val = -272
      endif
    elseif (!this.maximum.val.lt(-273)) then
      !this.minimum.val = -273
      !this.maximum.val = -272
    endif
    -- Check that the maximum value is larger than the minimum value
    if (!this.maximum.val.leq(!this.minimum.val)) then
      if !flag.eq(1) then
--(Continues on next page)
```

```
        !this.maximum.val = !this.minimum.val + !this.step.val
      elseif !flag.eq(2) then
        !this.minimum.val = !this.maximum.val - !this.step.val
      endif
    endif
  endif
endmethod
-------------------------------------------------------------------------------------------
-- Method .split() - Split the user entered string, convert and concatenate              --
-------------------------------------------------------------------------------------------
define method .split()
  -- Split the string
  !split = !this.stringInput.val.trim().split(!this.delimiter.val)
  !this.number.val = !split.size()
  !result = ||
  -- Loop through the split
  do !n index !split
    -- If it's Celsius....convert to Fahrenheit
    if !split[!n].substring(!split[!n].length()).upcase().eq(|C|) then
      !temp = !split[!n].substring(1, !split[!n].length() - 2).real()
      !result = !result & !this.celsiusToFahrenheit(!temp).string(!!INTEGERFMT) & |°F |
    else
      !temp = !split[!n].substring(1, !split[!n].length() - 2).real()
      !result = !result & !this.fahrenheitToCelsius(!temp).string(!!INTEGERFMT) & |°C |
    endif
  enddo
  -- Return the concatenated string back to the form
  !this.result.val = !result
endmethod
-------------------------------------------------------------------------------------------
-- Method .setData(REAL) - Store or retrieve the gadget values from the storage array    --
-------------------------------------------------------------------------------------------
define method .setData(!flag is REAL)
  -- Either save the values, or bring them back (array of strings)
  if !flag.eq(1) then
    !this.data[1] = !this.tempInput.val
    !this.data[2] = !this.tempChoiceFrame.val
    !this.data[3] = !this.minimum.val
    !this.data[4] = !this.maximum.val
    !this.data[5] = !this.step.val
    !this.data[6] = !this.stringInput.val
    !this.data[7] = !this.delimiter.val
  else
    if !this.data.size().eq(7) then
      !this.tempInput.val       = !this.data[1]
      !this.tempChoiceFrame.val = !this.data[2]
      !this.minimum.val         = !this.data[3]
      !this.maximum.val         = !this.data[4]
      !this.step.val            = !this.data[5]
      !this.stringInput.val     = !this.data[6]
      !this.delimiter.val       = !this.data[7]
      !this.temperatureConvert()
      !this.fill(|Celsius|)
      !this.split()
    endif
  endif
endmethod
-------------------------------------------------------------------------------------------
-- End of Form definition and methods                                                    --
-------------------------------------------------------------------------------------------
-- (c) Copyright  2010 to Current Year  AVEVA Solutions Limited                          --
-------------------------------------------------------------------------------------------
```

Method to transfer to and from the .data member

Appendix B3 - Example c2ex3.pmlfrm

```
-------------------------------------------------------------------------------------------
--                                                                                       --
-- (c) Copyright  2010 to Current Year  AVEVA Solutions Limited                          --
--                                                                                       --
-- File:            c2ex3.pmlfrm                                                         --
--                                                                                       --
-- Description:                                                                          --
--   Example form solution to Exercise 3                                                 --
--                                                                                       --
--   Supplied to support AVEVA training course TM1881 - PML: Form Design                 --
--                                                                                       --
-------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------
-- Form definition                                                                       --
-------------------------------------------------------------------------------------------
setup form !!c2ex3 dialog docking

  -- Set the form title
  !this.formTitle = |FILE Objects - Exercise 3|

  -- Define the gadgets for the form
  textpane .txtp at x 0 ymax anchor all width 60 height 15
  para     .para at xmin ymax anchor b+l+r text || width 30
  button .load |Load| at xmax.txtp - size ymin anchor b+r pixmap width 16 height 16
  button .save |Save| at xmin - size ymin anchor b+r pixmap width 16 height 16

  -- Form member to record which file is being used
  member .file is FILE

exit
-------------------------------------------------------------------------------------------
-- Constructor Method - Set the imahes and callbacks on the form                         --
-------------------------------------------------------------------------------------------
define method .c2ex3()
  -- Apply the images to the pixmap buttons
  !this.load.addPixmap(!!pml.getPathName(|openbrowser.png|))
  !this.save.addPixmap(!!pml.getPathName(|saveview16.png|))
  -- Apply tooltips to the icon buttons for clarity
  !this.load.setTooltip(|Load text file|)
  !this.save.setTooltip(|Save text file|)
  -- Set the callback to run the standard function !!filebrowser
  !dir    = 'C:\AVEVA\Plant\Training'
  !this.load.callback = |!!fileBrowser('| & !dir & |', '', 'Load text file', $
    TRUE, '!!c2ex3.load(!!fileBrowser.file)')|
  !this.save.callback = |!!fileBrowser('| & !dir & |', '', 'Save text file', $
    FALSE, '!!c2ex3.save(!!fileBrowser.file)')|
endmethod
-------------------------------------------------------------------------------------------
-- Method .load(FILE) - Read the chosen file into the textpane                           --
-------------------------------------------------------------------------------------------
define method .load(!file is FILE)
  -- Store and read the file
  !this.file = !file
  !this.txtp.val = !this.file.readFile()
  -- Display the filename
  !this.para.val = !this.file.string()
endmethod
-------------------------------------------------------------------------------------------
-- Method .save(FILE) - Save the chosen file from the textpane                           --
-------------------------------------------------------------------------------------------
define method .save(!file is file)
  !this.file = !file
  !this.file.writefile('OVER', !this.txtp.val)
  !this.para.val = !this.file.string()
endmethod
-------------------------------------------------------------------------------------------
-- End of Form definition and methods                                                    --
-------------------------------------------------------------------------------------------
-- (c) Copyright  2010 to Current Year  AVEVA Solutions Limited                          --
-------------------------------------------------------------------------------------------
```

Appendix B4 - Example c2ex4.pmlfrm

```
-------------------------------------------------------------------------------------------
--                                                                                       --
-- (c) Copyright  2010 to Current Year  AVEVA Solutions Limited                          --
--                                                                                       --
-- File:            c2ex4.pmlfrm                                                         --
--                                                                                       --
-- Description:                                                                          --
--   Example form solution to Exercise 4                                                 --
--                                                                                       --
--   Supplied to support AVEVA training course TM1881 - PML: Form Design                 --
--                                                                                       --
-------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------
-- Form definition                                                                       --
-------------------------------------------------------------------------------------------
setup form !!c2ex4 dialog resiz

  -- Set the form title and initialisation call
  !this.formTitle     = |Equipment Checker|
  !this.initcall      = |!!this.init()|

  -- Create a MENU object for use as a context menu on the list gadget
  !this.newMenu(|popupMenu|)
  !popUpCall = '!!CE = !this.nozzleList.selection().dbref()'
  !this.popupMenu.add('CALLBACK', 'Go to Nozzle', !popUpCall)

  -- Define the gadgets at the top of the form
  button .update linklabel |Update| call |!!this.init()| width 5
  para   .title at xmax ymin anchor t+l+r text |Available Equipments below | width 34
  line   .line at xmin.update ymax anchor t+l+r horizontal width 42

  !buttPos = |xmax - size ymax anchor b+r|
  !anch1   = |anchor t+l+r |
  !anch2   = |anchor b+l+r|

  -- Define the gadgets to choice the equipments from
  combo  .equip at xmin.update ymax+0.2 |Select an Equipment| tagwidth 15 $!anch1 width 24
  list   .nozzleList  at xmin.update ymax+0.2 anchor all width 40 length 5
  button .refresh linklabel |Refresh Checks| at $!buttPos width 9.6

  -- Define the gadgets to allow the attributes to be altered
  !tag = |Equipment Attributes|
  textpane .atta |$!tag| at xmin.nozzleList ymax-0.5 $!anch2 width 42 height 3.5
  button   .updateAtta linklabel |Update Attributes| at $!buttPos width 10.3

exit
-------------------------------------------------------------------------------------------
-- Constructor Method - Set the images and callbacks on the form                         --
-------------------------------------------------------------------------------------------
define method .c2ex4()
  -- Set the titles and context menu on the list
  !title = |Nozzles/Connected?/Attached?/Aligned?/Size?|
  !this.nozzleList.setHeadings(!title.split(|/|))
  !this.nozzleList.setpopup(!this.popupMenu)
  -- Pre-fill-in some attributes to be editted
  !info[1] = 'Description - Unset'
  !info[2] = 'Function - Unset'
  !info[3] = 'Purpose - Unset'
  !this.atta.val = !info
  -- Set the gadget callbacks
  !this.equip.callback = |!this.setEquip(|
  !this.refresh.callback = |!this.checkNozzles()|
  !this.updateAtta.callback = |!this.updateAtt(2)|
endmethod
-------------------------------------------------------------------------------------------
-- Initialisation Method - Collect all equi for level and display nozzles                --
-------------------------------------------------------------------------------------------
define method .init()
  -- If at SITE level, collect for the site otherwise do it for ZONE
  if !!ce.type.eq(|SITE|) then
    !level = |SITE|
  else
    !level = |ZONE|
  endif
--(Continues on next page)
```

```
  -- Collect the equipment using PML 1 style collection syntax
  VAR !coll COLL ALL EQUIP FOR $!level
  handle ANY
    !!alert.error(|Make sure you are at an EQUI, ZONE or SITE element|)
  elsehandle NONE
    -- If no equipment are found, and we're at a ZONE try collecting for SITE
    if !coll.size().eq(0) and !level.eq(|ZONE|) then
      !!alert.message(|No equipment found in current ZONE, so whole SITE will be searched|)
      VAR !coll COLL ALL EQUIP FOR SITE
      !level = |SITE|
    endif
    -- Update the title paragraph gadget
    !this.title.val = |Available Equipments below | & !level
    -- Evaluate the fullnames of the equis and display the information
    VAR !name EVAL FLNN FOR ALL FROM !coll
    !this.equip.dtext = !name
    !this.equip.rtext = !coll
    -- Collect the nozzles
    !this.collectNozzles()
  endhandle
endmethod
-------------------------------------------------------------------------------------------
-- Method .collectNozzles() - For the selected EQUI, collect all the nozzles             --
-------------------------------------------------------------------------------------------
define method .collectNozzles()
  -- Get the selected EQUI element
  !equipRef = !this.equip.selection().dbref()
  if !equipRef.unset() then
    -- If the element is unset, then clear the nozzle list
    !this.nozzleList.clear()
  else
    -- Perform a PML2 style collection for the nozzles
    !nozzColl = object COLLECTION()
    !nozzColl.type('NOZZ')
    !nozzColl.scope(!equipRef)
    !results = !nozzColl.results()
    !this.nozzleList.dtext = !results.evaluate(object block ('!results[!evalIndex].flnn'))
    !this.nozzleList.rtext = !results.evaluate(object block ('!results[!evalIndex].string()'))
    -- Check the nozzles and update the results
    !this.checkNozzles()
    !this.updateAtt(1)
  endif
endmethod
-------------------------------------------------------------------------------------------
-- Method .setEquip(GADGET, STRING) - Open callback on combo gadget to check user entry --
-------------------------------------------------------------------------------------------
define method .setEquip(!gad is gadget, !event is STRING)
  -- If the user has typed into the gadget
  if !event.eq('VALIDATE') then
    !userInput = !this.equip.displayText()
    -- Loop through available equipments and choice the nearest match
    do !n index !this.equip.dtext
      !chrs = !userInput.length()
      !test = !this.equip.dtext[!n].upcase().substring(1, !chrs)
      if !userInput.upcase().eq(!test) then
        !this.equip.val = !n
        break
      endif
    enddo
  endif
  -- Recollect the nozzles on the newly selected equipment
  !this.collectNozzles()
endmethod
-------------------------------------------------------------------------------------------
-- Method .checkNozzles() - Check the collected nozzles and display the results          --
-------------------------------------------------------------------------------------------
define method .checkNozzles()
  if !this.nozzleList.rtext.set() then
    -- Loop through the collected nozzles
    do !n index !this.nozzleList.rtext
      !nozz = !this.nozzleList.rtext[!n].dbref()
      -- set the default result
      do !m from 2 to 4
        !result[!m] = |N/A|
      enddo
      -- Check 1: is the connection valid
--(Continues on next page)

      if !nozz.cref.unset().or(!nozz.cref.badref()) then
        !result[1] = |Check|
```

```
      else
        !result[1] = |OK|
        !end = |t|
        if !nozz.cref.href.eq(!nozz) then
          !end = |h|
        endif
        -- Check 2: is the nozzle and the same position as the attached
        if !nozz.pos.wrt( /* ).eq(!nozz.cref.attribute(!end & |pos|).wrt( /* )) then
          !result[2] = |OK|
        endif
        -- Check 3: is the nozzle and the same direction as the attached
        if !nozz.pdir[1].wrt( /* ).eq(!nozz.cref.attribute(!end & |dir|).wrt( /* )) then
          !result[3] = |OK|
        endif
        -- Check 4: is the nozzle and the same size as the attached
        if !nozz.cpar[1].eq(!nozz.cref.attribute(!end & |bore|).real()) then
          !result[4] = |OK|
        endif
      endif
      -- Record the check results for this nozzle
      !nozzleInfo[!n][1] = !nozz.flnn
      !nozzleInfo[!n].appendArray(!result)
    enddo
    -- Apply the results back to the list gadget
    !rtext = !this.nozzleList.rtext
    !this.nozzleList.setRows(!nozzleInfo)
    !this.nozzleList.rtext = !rtext
  endif
endmethod
-------------------------------------------------------------------------------------------
-- Method .updateAtt(REAL) - Check the collected nozzles and display the results        --
-------------------------------------------------------------------------------------------
define method .updateAtt(!flag is REAL)
  -- Get the selected piece of equipment and the available attributes
  !equip = !this.equip.selection().dbref()
  !availAtts = !equip.attributes()
  -- Update the textpane title
  !this.atta.tag = !equip.flnn & ' Attributes'
  !rows = !this.atta.val
  -- Loop through the rows of the textpane to extract and use the information
  !out = ARRAY()
  do !n index !rows
    -- Split the line on a "-"
    !split = !rows[!n].split('-')
    !attrib = !split[1].trim()
    -- Evaluate the available attributes so they are the same string length as the entered
    !block = object block ('!availAtts[!evalIndex].upcase().substring(1, !attrib.length())')
    !avail = !availAtts.evaluate(!block)
    -- If the attribute can be found, then use it
    if !avail.findfirst(!attrib.upcase()).set() then
      if !flag.eq(1) then
        -- If the flag = 1 then update the textpane with the current attribute value
        !value = !equip.attribute(!availAtts[!avail.findfirst(!attrib.upcase())])
        !out.append(!attrib & | - | & !value)
      elseif !flag.eq(2) then
        -- If the flag = 2 then assign the value to the attribute
        !val = !split[2].trim()
        !equip.attribute(!availAtts[!avail.findfirst(!attrib.upcase())]).assign(!val)
        !out.append(!attrib & | - | & !val)
      endif
    endif
  enddo
  -- Display the updated attribute list
  !this.atta.val = !out
endmethod
-------------------------------------------------------------------------------------------
-- End of Form definition and methods                                                    --
-------------------------------------------------------------------------------------------
-- (c) Copyright  2010 to Current Year  AVEVA Solutions Limited                          --
-------------------------------------------------------------------------------------------
```

Appendix B5 - Example !!traExampleVolumeView

```
-------------------------------------------------------------------------------------------
--                                                                                       --
-- (c) Copyright  2010 to Current Year  AVEVA Solutions Limited                          --
--                                                                                       --
-- File:            traExampleVolumeView.pmlfrm                                          --
--                                                                                       --
-- Description:                                                                          --
--   Form to demonstrate an VOLUME view gadget                                           --
--                                                                                       --
--   Supplied to support AVEVA training course TM1881 - PML: Form Design                 --
--                                                                                       --
-------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------
-- Form definition                                                                       --
-------------------------------------------------------------------------------------------
setup form !!traExampleVolumeView

  -- Set the form title
  !this.formTitle = |Volume View|

  -- Set the callbacks on the form
  !this.initcall  = |!!this.init()|
  !this.firstShownCall  = |!!this.firstShown()|
  !this.killingCall     = |!!this.close()|

  -- Define a context menu for the volume view gadget
  !this.newMenu(|pop1|)
  !this.pop1.add('CALLBACK', 'Limits CE', '!!this.limitsCE()')
  -- Set the standard icon size
  !pix = |pixmap wid 16 hei 16|

  -- Define a prompt paragraph gadget.  This will provide feedback to the user
  para .prompt text |Navigate : | width 46 lines 1

  -- Define the four buttons down the L.H.S of the form
  button .but1 at xmin ymax call |!!this.limitsCE()| $!pix
  button .but2 at xmin ymax call |!!this.walkDrawlist()| $!pix
  button .but3 at xmin ymax call |!!this.setDrawlist(1, FALSE)| $!pix
  button .but4 at xmin ymax call |!!this.setDrawlist(2, FALSE)| $!pix

  -- Define the volume view element
  view .volumeView at xmax.but1 ymax.prompt volume

    -- Set the size and the initial viewing direction
    width 45 height 15
    limits auto
    isometric 3

    -- Set the required "INMODES". These are required if the user is to pick from the view
    inmode create _navi       type |DES_NAVIGATE|    $* Navigate
    inmode create _pick       type |DES_PICK|        $* Standard Element Pick
    inmode create _pAny       type |DES_PICK_ANY|    $* Pick Any (Element, Ppoint, Pline)
    inmode create _pLine      type |DES_PICK_PLINE|  $* Plines
    inmode create _point      type |DES_PICK_POINT|  $* Ppoints
    inmode create _screen     type |DES_3D_LINE|     $* Screen Position
    inmode create _pickxgeom  type |DES_PICK_XGEOM|  $* Laser Model rays
    inmode create _pickDetail type |DES_PICK_DETAIL| $* Default (Navigation)
    inmode create _default    type |DES_NAVIGATE|    $* Default (Navigation)

  exit

  -- Define the footer
  line .line at xmin.but1 ymax+0.5 anchor b+l+r horizontal width 47.5
  para .footer at xcen - 0.5 * size ymax anchor b text |AVEVA Training| width 11

  -- A form member to remember the drawlist associated with the view
  member .drawlist is REAL

exit

--(Continues on next page)
```

```
-------------------------------------------------------------------------------------------
-- Constructor Method - Setup the form when it is loaded                                 --
-------------------------------------------------------------------------------------------
define method .traExampleVolumeView()

  -- Apply the icons to the four buttons
  !this.but1.addPixmap(!!pml.getPathName(|autoce16.png|))
  !this.but2.addPixmap(!!pml.getPathName(|ng_zoomtodrawlist.png|))
  !this.but3.addPixmap(!!pml.getPathName(|addce16.png|))
  !this.but4.addPixmap(!!pml.getPathName(|removece16.png|))

  -- As icons are being used, set tooltips to help users
  !this.but1.setTooltip(|Limits CE|)
  !this.but2.setTooltip(|Walk to Drawlist|)
  !this.but3.setTooltip(|Add to Drawlist|)
  !this.but4.setTooltip(|Remove from Drawlist|)

  -- Apply some default settings to the view element
  !this.volumeView.borders = FALSE
  !this.volumeView.background = |darkslate|
  !this.volumeView.shaded = TRUE
  !this.volumeView.projection = |PARALLEL|
  !this.volumeView.radius = 100
  !this.volumeView.range = 500.0
  !this.volumeView.eyemode = FALSE
  !this.volumeView.step = 25
  !this.volumeView.setpopup(!this.pop1)

  -- Set Defaults for various INMODEs
  !this.volumeView.navi.cursor     = 'Pointer'
  !this.volumeView.navi.prompt     = 'Navigate : '
  !this.volumeView.pick.cursor     = 'Pointer'
  !this.volumeView.pick.prompt     = 'Pick Element : '
  !this.volumeView.pAny.cursor     = 'Pointer'
  !this.volumeView.pAny.prompt     = 'Pick Any : '
  !this.volumeView.pLine.cursor    = 'Pointer'
  !this.volumeView.pLine.prompt    = 'Pick Pline : '
  !this.volumeView.point.cursor    = 'Pick'
  !this.volumeView.point.prompt    = 'Point Ppoint : '
  !this.volumeView.screen.cursor   = 'Crosshair'
  !this.volumeView.screen.prompt   = 'Pick 3D Position : '
  !this.volumeView.default.cursor  = 'Pointer'
  !this.volumeView.default.prompt  = 'Navigate : '

  -- Create local drawlist within the global object
  !this.drawlist = !!gphDrawlists.createDrawlist()
endmethod
-------------------------------------------------------------------------------------------
-- First Shown Method - Register the view gadget and attach the drawlist                 --
-------------------------------------------------------------------------------------------
define method .firstShown()
  -- Add 3D view to view system
  !!gphViews.add(!this.volumeView)
  -- Add local drawlist add to 3D view
  !!gphDrawlists.attachView(!this.drawlist, !this.volumeView)
endmethod
-------------------------------------------------------------------------------------------
-- Initialisation Method - Update the drawlist and active the view                       --
-------------------------------------------------------------------------------------------
define method .init()
  -- Update the drawlist
  !this.setDrawlist(1, TRUE)
  -- Active the volume view
  !this.volumeView.active = TRUE
  -- Turn on holes drawn
  !!gphDrawlists.drawlists[!this.drawlist].holes(TRUE)
endmethod
-------------------------------------------------------------------------------------------
-- Killing Call Method - Detach the drawlist and then delete it                          --
-------------------------------------------------------------------------------------------
define method .close()
  !!gphDrawlists.detachView(!this.volumeView)
  !!gphDrawlists.deleteDrawlist(!this.drawlist)
endmethod

--(Continues on next page)
```

```
-------------------------------------------------------------------------------------------
-- Method .walkDrawlist() - Set the view limits based on the drawlist                    --
-------------------------------------------------------------------------------------------
define method .walkDrawlist()
  -- Get the drawlist associated with the view
  !drawlist = !!gphDrawlists.drawlist(!this.drawlist)
  -- Derive a volume object from the members of the drawlist
  !volume = object volume(!drawlist.members())
  -- Derive a limits array in the expected format
  !limits[1]    = !volume.from.east
  !limits[2]    = !volume.to.east
  !limits[3]    = !volume.from.north
  !limits[4]    = !volume.to.north
  !limits[5]    = !volume.from.up
  !limits[6]    = !volume.to.up
  -- Apply the limits to the volume view
  !this.volumeView.limits = !limits
  handle ANY
    -- Incase the drawlist is empty or has no volume
  endhandle
endmethod
-------------------------------------------------------------------------------------------
-- Method .limitsCE() - Set the view limits based on the current element                 --
-------------------------------------------------------------------------------------------
define method .limitsCE()
  -- Set the Views limits based on the chosen element
  !!gphViews.limits(!this.volumeView, !!CE)
endmethod
-------------------------------------------------------------------------------------------
-- Method .setDrawlist(REAL, BOOLEAN) - Update the drawlist                              --
-------------------------------------------------------------------------------------------
define method .setDrawlist(!flag is REAL, !reset is BOOLEAN)
  -- Get the drawlist associated with the view
  !drawlist = !!gphDrawlists.drawlist(!this.drawlist)
  -- Clear the drawlist is a reset is requested
  if !reset then
    !drawlist.removeall()
  endif
  -- Add/Remove CE
  if !flag.eq(1) then
    !drawlist.add(!!CE)
  elseif !flag.eq(2) then
    !drawlist.remove(!!CE)
  endif
  -- Update the limits of the view based on the drawlist
  !this.walkDrawlist()
endmethod
-------------------------------------------------------------------------------------------
-- End of Form definition and methods                                                    --
-------------------------------------------------------------------------------------------
-- (c) Copyright  2010 to Current Year  AVEVA Solutions Limited                          --
-------------------------------------------------------------------------------------------
```

Appendix B6 - Example c2ex5.pmlfrm

```
-------------------------------------------------------------------------------------------
--                                                                                       --
-- (c) Copyright  2010 to Current Year  AVEVA Solutions Limited                          --
--                                                                                       --
-- File:            c2ex5.pmlfrm                                                         --
--                                                                                       --
-- Description:                                                                          --
--   Example form solution to Exercise 5                                                 --
--                                                                                       --
--   Supplied to support AVEVA training course TM1881 - PML: Form Design                 --
--                                                                                       --
-------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------
-- Form definition                                                                       --
-------------------------------------------------------------------------------------------
setup form !!c2ex5 dialog docking

  -- Set the form title and callbacks
  !this.formTitle     = |Equipment Checker|
  !this.initcall      = |!!this.init()|
  !this.firstShownCall  = |!!this.firstShown()|
  !this.killingCall     = |!!this.close()|
  !this.quitcall      = |!!this.clear()|

  -- Create a MENU object for use as a context menu on the list gadget
  !this.newMenu(|popupMenu|)
  !this.popupMenu.add('CALLBACK', 'Go to Nozzle', '!!CE = !this.nozzleList.selection().dbref()')
  -- Define the prompt and view gadgets
  para .prompt text |Navigate : | width 46 lines 1
  frame .view panel at x 0.5 ymax.prompt anchor all wid 1 hei 1
    view .volumeView at x 0.5 ymax.prompt VOLUME
      width 40 height 28
      border off
      shading on
      isometric 3
    exit
  exit

  -- Define the gadgets at the top of the form
  button .update linklabel |Update| at xmax ymin.prompt anchor t+r call |!!this.init()| width 5
  para   .title at xmax ymin anchor t+r text |Available Equipments below | width 34
  line   .line at xmin.update ymax anchor t+r horizontal width 42

  !buttPos = |xmax - size ymax anchor b+r|
  !anch1   = |anchor t+l+r |
  !anch2   = |anchor b+l+r|
  -- Define the gadgets to choice the equipments from
  combo  .equip at xmin.update ymax+0.2 |Select an Equipment| tagwidth 15 anchor t+r width 24
  list   .nozzleList at xmin.update ymax+0.2 anchor t+r width 40 length 5
  button .refresh linklabel |Refresh Checks| at $!buttPos anchor t+r width 9.6

  -- Define the gadgets to allow the attributes to be altered
  !tag = |Equipment Attributes|
  textpane .atta |$!tag| at xmin.nozzleList ymax-0.5 anchor t+r width 42 height 3.5
  button   .updateAtta linklabel |Update Attributes| at $!buttPos anchor t+r width 10.3
  para     .taskTitle at xmin.update ymax.atta + 0.5 anchor t+r text |Available Tasks|
  line     .taskLine at xmin.update ymax.taskTitle anchor t+r horiz width 42

  !anch = |anchor t+r|
  para     .clipPix at xmin + 1 ymax.taskLine + 0.2 anchor t+r pixmap wid 16 hei 16
  button   .clipBut    linklabel at xmax + 1 ymin |Add Connected and Enable Clipbox...| $!anch
  para     .tagNozzPix at xmin.clipPix ymax + 0.2 anchor t+r pixmap wid 16 hei 16
  button   .tagNozzBut linklabel at xmax + 1 ymin |Tag Selected Nozzle| $!anch
  para     .tagNozzlesPix at xmin.clipPix  ymax + 0.2 anchor t+r pixmap wid 16 hei 16
  button   .tagNozzlesBut linklabel at xmax + 1 ymin |Tag All Nozzles| $!anch
  para     .hlNozzPix at xmin.clipPix ymax + 0.2 anchor t+r pixmap wid 16 hei 16
  button   .hlNozzBut linklabel at xmax + 1 ymin |Highlight Nozzles| $!anch

  frame .limitslide foldup |Edit Clip Volume| at xmin.atta ymax.hlNozzPix + 0.5 $!anch width 41
    slider .slide anchor l+r range 0 +2000 step 100 val 0 width 40 height 2.5
    text   .minslide at xmin.limitslide+0.2  ymin.limitslide+2.5 width 5 is REAL
    text   .maxslide at xmax.limitslide-size ymin.limitslide+2.5 width 5 is REAL
    toggle .incl |Update view limits?| at xmax.minslide + 5 ymin.limitslide+2.5
  exit

  !fPos = |xmin.atta ymax.limitslide+0.1|
  frame .colour foldup |Edit Highlight Colour| at $!fPos $!anch width .limitslide
    para .para1       at xmin.colour + 0.5 ymin.colour + 1.2 text |Current equipment  |
```

```
    button .butt1 | | at xmax.para1 - 3 ymin.para1  pixmap wid 25 hei 15
    para .para2       at xmin.colour + 0.5 ymax.butt1        text |Connected nozzles  |
    button .butt2 |  | at xmax.para2 - 3 ymax.butt1 pixmap wid 25 hei 15
    para .para3       at xmin.colour + 0.5 ymax.butt2        text |Unconnected nozzles|
    button .butt3 |  | at xmax.para3 - 3 ymax.butt2 pixmap wid 25 hei 15
    para .para4       at xmin.colour + 0.5 ymax.butt3        text |Check nozzles      |
    button .butt4 |  | at xmax.para3 - 3 ymax.butt3 pixmap wid 25 hei 15

    frame .colourpick panel at xmax.butt1  + 2.5 ymin.colour + 0.5 width 10
      !val = 0
      do !I from 1 to 8
        do !J from 0 to 4
          !val = !val + 1
          !no = !I & |000| & !J
          !x = (!I * 2) - 1
          !y = (!J * 0.6) + 0.75
          button .butt$!no | | pixmap at x$!x ymin.colourpick + $!y backg $!val width 10 aspect 1
          !this.butt$!<no>.callback = |!this.colourpick($!val)|
        enddo
      enddo
    exit
  exit
  member .clipbox is GPHCLIPBOX
  member .drawlist is REAL
  member .limits is ARRAY
  member .col    is REAL
  member .tagged is ARRAY
  member .highlight is ARRAY
exit
-------------------------------------------------------------------------------
-- Constructor Method - Set the images and callbacks on the form              --
-------------------------------------------------------------------------------
define method .c2ex5()
  -- Set the titles and context menu on the list
  !title = |Nozzles/Connected?/Attached?/Aligned?/Size?|
  !this.nozzleList.setHeadings(!title.split(|/|))
  !this.nozzleList.setpopup(!this.popupMenu)
  -- Pre-fill-in some attributes to be editted
  !info[1] = 'Description - Unset'
  !info[2] = 'Function - Unset'
  !info[3] = 'Purpose - Unset'
  !this.atta.val = !info
  -- Set the gadget callbacks
  !this.equip.callback = |!this.setEquip()|
  !this.nozzleList.callback = |!this.check()|
  !this.refresh.callback = |!this.checkNozzles()|
  !this.updateAtta.callback = |!this.updateAtt(2)|

  !this.slide.callback = |!this.slide()|
  !this.minslide.callback = |!this.updateRange()|
  !this.maxslide.callback = |!this.updateRange()|

  !this.butt1.callback = |!this.colourpick('A' )|
  !this.butt2.callback = |!this.colourpick('B' )|
  !this.butt3.callback = |!this.colourpick('C' )|
  !this.butt4.callback = |!this.colourpick('D' )|

  !this.clipBut     .callback = |!this.enableClip()|
  !this.tagNozzBut  .callback = |!this.tag(!this.tagNozzBut)|
  !this.tagNozzlesBut.callback = |!this.tag(!this.tagNozzlesBut)|
  !this.hlNozzBut    .callback = |!this.highlight()|

  !this.volumeView.borders = FALSE
  !this.volumeView.background = |darkslate|
  !this.volumeView.shaded = TRUE
  !this.volumeView.projection = |PARALLEL|
  !this.volumeView.radius = 100
  !this.volumeView.range = 500.0
  !this.volumeView.eyemode = FALSE
  !this.volumeView.step = 25

  !this.minslide.val = 0
  !this.maxslide.val = 2000

  !this.drawlist = !!gphDrawlists.createDrawList()
  !this.clipPix.addPixmap(!!pml.getPathName('clipce16.png'))
  !this.tagNozzPix.AddPixmap(!!PML.GetPathName('draft_lab_01.png'))
  !this.tagNozzlesPix.AddPixmap(!!PML.GetPathName('draft_lab_01.png'))
  !this.hlNozzPix.AddPixmap(!!PML.GetPathName('nozzle-16.png'))
```

```
  !this.colourpick.visible = FALSE
  !this.limitslide.visible  = FALSE
  !this.limitslide.expanded = FALSE
  !this.colour.visible  = FALSE
  !this.colour.expanded = FALSE

  !this.clipbox = object GPHCLIPBOX()
  !this.clipbox.view = !this.volumeView
  !this.clipbox.capOn()

  !this.butt1.background = 1
  !this.butt2.background = 20
  !this.butt3.background = 18
  !this.butt4.background = 23

endmethod
-------------------------------------------------------------------------------------------
-- First Shown Method - Collect all equi for level and display nozzles                --
-------------------------------------------------------------------------------------------
define method .firstShown()
  -- Add 3D view to view system
  !!gphViews.add(!this.volumeView)
  -- Add local drawlist add to 3D view
  !!gphDrawlists.attachView(!this.drawlist, !this.volumeView)
endmethod
-------------------------------------------------------------------------------------------
-- Killing Call Method - Collect all equi for level and display nozzles               --
-------------------------------------------------------------------------------------------
define method .close()
  !!gphDrawlists.detachView(!this.volumeView)
  !!gphDrawlists.deleteDrawlist(!this.drawlist)
endmethod
-------------------------------------------------------------------------------------------
-- Initialisation Method - Collect all equi for level and display nozzles             --
-------------------------------------------------------------------------------------------
define method .init()
  -- Activate the view for use
  !this.volumeView.active = TRUE
  -- If at SITE level, collect for the site otherwise do it for ZONE
  if !!ce.type.eq(|SITE|) then
    !level = |SITE|
  else
    !level = |ZONE|
  endif
  -- Collect the equipment using PML 1 style collection syntax
  VAR !coll COLL ALL EQUIP FOR $!level
  handle ANY
    !!alert.error(|Make sure you are at an EQUI, ZONE or SITE element|)
  elsehandle NONE
    -- If no equipment are found, and we're at a ZONE try collecting for SITE
    if !coll.size().eq(0) and !level.eq(|ZONE|) then
      !!alert.message(|No equipment found in current ZONE, so whole SITE will be searched|)
      VAR !coll COLL ALL EQUIP FOR SITE
      !level = |SITE|
    endif
    -- Update the title paragraph gadget
    !this.title.val = |Available Equipments below | & !level
    -- Evaluate the fullnames of the equis and display the information
    VAR !name EVAL FLNN FOR ALL FROM !coll
    !this.equip.dtext = !name
    !this.equip.rtext = !coll
    -- Collect the nozzles
    !this.collectNozzles()
  endhandle
endmethod



-------------------------------------------------------------------------------------------
-- Method .collectNozzles() - For the selected EQUI, collect all the nozzles          --
-------------------------------------------------------------------------------------------
define method .collectNozzles()
  -- Get the selected EQUI element
  !equipRef = !this.equip.selection().dbref()
  if !equipRef.unset() then
    -- If the element is unset, then clear the nozzle list
    !this.nozzleList.clear()
  else
    -- Perform a PML2 style collection for the nozzles
    !nozzColl = object COLLECTION()
```

```
    !nozzColl.type('NOZZ')
    !nozzColl.scope(!equipRef)
    !results = !nozzColl.results()
    !this.nozzleList.dtext = !results.evaluate(object block ('!results[!evalIndex].flnn'))
    !this.nozzleList.rtext = !results.evaluate(object block ('!results[!evalIndex].string()'))
    -- Check the nozzles and update the results
    !this.checkNozzles()
    !this.clear()
    !this.updateAtt(1)
    !this.setDrawlist()
    !this.enableClip()
    !this.highlight()
    !this.tagged.clear()
    do !n index !this.nozzleList.dtext
      !this.tagged[!n] = FALSE
    enddo
  endif
endmethod
-------------------------------------------------------------------------------------------
-- Method .setEquip(GADGET, STRING) - Open callback on combo gadget to check user entry --
-------------------------------------------------------------------------------------------
define method .setEquip(!gad is gadget, !event is STRING)
  -- If the user has typed into the gadget
  if !event.eq('VALIDATE') then
    !userInput = !this.equip.displayText()
    -- Loop through available equipments and choice the nearest match
    do !n index !this.equip.dtext
      !chrs = !userInput.length()
      !test = !this.equip.dtext[!n].upcase().substring(1, !chrs)
      if !userInput.upcase().eq(!test) then
        !this.equip.val = !n
        break
      endif
    enddo
  endif
  -- Recollect the nozzles on the newly selected equipment
  !this.collectNozzles()
endmethod
-------------------------------------------------------------------------------------------
-- Method .checkNozzles() - Check the collected nozzles and display the results         --
-------------------------------------------------------------------------------------------
define method .checkNozzles()
  if !this.nozzleList.rtext.set() then
    -- Loop through the collected nozzles
    do !n index !this.nozzleList.rtext
      !nozz = !this.nozzleList.rtext[!n].dbref()
      -- set the default result
      do !m from 2 to 4
        !result[!m] = |N/A|
      enddo
      -- Check 1: is the connection valid
      if !nozz.cref.unset().or(!nozz.cref.badref()) then
        !result[1] = |Check|
      else
        !result[1] = |OK|
        !end = |t|
        if !nozz.cref.mref.eq(!nozz) then
          !end = |h|
        endif
        -- Check 2: is the nozzle and the same position as the attached
        if !nozz.pos.wrt( /* ).eq(!nozz.cref.attribute(!end & |pos|).wrt( /* )) then
          !result[2] = |OK|
        endif

        -- Check 3: is the nozzle and the same direction as the attached
        if !nozz.pdir[1].wrt( /* ).eq(!nozz.cref.attribute(!end & |dir|).wrt( /* )) then
          !result[3] = |OK|
        endif
        -- Check 4: is the nozzle and the same size as the attached
        if !nozz.cpar[1].eq(!nozz.cref.attribute(!end & |bore|).real()) then
          !result[4] = |OK|
        endif
      endif
      -- Record the check results for this nozzle
      !nozzleInfo[!n][1] = !nozz.flnn
      !nozzleInfo[!n].appendArray(!result)
    enddo
    !this.highlight = !nozzleInfo
    -- Apply the results back to the list gadget
    !rtext = !this.nozzleList.rtext
```

```
    !this.nozzleList.setRows(!nozzleInfo)
    !this.nozzleList.rtext = !rtext
  endif
endmethod

-------------------------------------------------------------------------------------------
-- Method .updateAtt(REAL) - Check the collected nozzles and display the results        --
-------------------------------------------------------------------------------------------

define method .updateAtt(!flag is REAL)
  -- Get the selected piece of equipment and the available attributes
  !equip = !this.equip.selection().dbref()
  !availAtts = !equip.attributes()
  -- Update the textpane title
  !this.atta.tag = !equip.flnn & ' Attributes'
  !rows = !this.atta.val
  -- Loop through the rows of the textpane to extract and use the information
  !out = ARRAY()
  do !n index !rows
    -- Split the line on a "-"
    !split = !rows[!n].split('-')
    !attrib = !split[1].trim()
    -- Evaluate the available attributes so they are the same string length as the entered
    !block = object block ('!availAtts[!evalIndex].upcase().substring(1, !attrib.length())')
    !avail = !availAtts.evaluate(!block)
    -- If the attribute can be found, then use it
    if !avail.findfirst(!attrib.upcase()).set() then
      if !flag.eq(1) then
        -- If the flag = 1 then update the textpane with the current attribute value
        !value = !equip.attribute(!availAtts[!avail.findfirst(!attrib.upcase())])
        !out.append(!attrib & | - | & !value)
      elseif !flag.eq(2) then
        -- If the flag = 2 then assign the value to the attribute
        !val = !split[2].trim()
        !equip.attribute(!availAtts[!avail.findfirst(!attrib.upcase())]).assign(!val)
        !out.append(!attrib & | - | & !val)
      endif
    endif
  enddo
  -- Display the updated attribute list
  !this.atta.val = !out
Endmethod

-------------------------------------------------------------------------------------------
-- Method .setDrawlist(REAL) - Add the select equipment to the drawlist                  --
-------------------------------------------------------------------------------------------

define method .setDrawlist()
  -- Get the selected equipment and drawlist associated with the view
  !equip = !this.equip.selection().dbref()
  !drawlist = !!gphDrawlists.drawlist(!this.drawlist)
  !drawlist.holes(TRUE)
  -- Clear the draw, add the equi and set the limits
  !drawlist.removeall()
  !drawlist.add(!equip)
  !!gphViews.limits(!!this.volumeView, !equip)
Endmethod

-------------------------------------------------------------------------------------------
-- Method .setDrawlist(REAL) - Add the select equipment to the drawlist                  --
-------------------------------------------------------------------------------------------

define method .colourpick(!colour is ANY)
  -- Based on the colour button pressed, set the colour
  !col = !colour.string()

  if !col.eq('A') then
    !this.col = 1
    !this.colourpick.visible = TRUE
  elseif !col.eq('B') then
    !this.col = 2
    !this.colourpick.visible = TRUE
  elseif !col.eq('C') then
    !this.col = 3
    !this.colourpick.visible = TRUE
  elseif !col.eq('D') then
    !this.col = 4
    !this.colourpick.visible = TRUE
  else
    if !this.col.eq(1) then
```

```
      !this.butt1.background = !col.real()
    elseif !this.col.eq(2) then
      !this.butt2.background = !col.real()
    elseif !this.col.eq(3) then
      !this.butt3.background = !col.real()
    elseif !this.col.eq(4) then
     !this.butt4.background = !col.real()
    endif
    !this.colourpick.visible = FALSE
    -- Rehighlight with the new colours
    !this.highlight()
  endif
endmethod
-------------------------------------------------------------------------------------------
-- Method .highlight() - Highlight the equipment and nozzles                              --
-------------------------------------------------------------------------------------------
define method .highlight()
  -- Get the selected equipment
  !equip = !this.equip.selection().dbref()
  -- If highlighting is required
  if !this.hlNozzBut.val then
    -- Highlight the equipment
    !!gphDrawlists.drawlists[!this.drawlist].highlight(!equip,!this.butt1.background)
    -- Update the icons
    !this.hlNozzPix.AddPixmap(!!PML.GetPathName('nozzle-16.png'))
    !this.hlNozzBut.tag = |Unhightlight Nozzles|
    -- Show then hidden frame
    !this.colour.visible = TRUE
    -- Loop through any displayed nozzles
    if !this.nozzleList.rtext.set() then
      -- Highlight the nozzles based on the check results
      !nozz = !this.highlight
      do !I index !nozz
        !nozzle = !this.nozzleList.rtext[!I].dbref()
        if !nozz[!I][2].eq(|OK|) then
          !col = !this.butt2.background
        elseif !nozz[!I][2].eq(|Check|) then
          !col = !this.butt3.background
        endif
        if !nozz[!I][3].eq(|Check|) or !nozz[!I][4].eq(|Check|) or !nozz[!I][5].eq(|Check|) then
          !col = !this.butt4.background
        endif
        -- Highlight the nozzle
        !!gphDrawlists.drawlists[!this.drawlist].highlight(!nozzle,!col)
      enddo
    endif
  else
    -- Reset the icons
    !this.hlNozzPix.AddPixmap(!!PML.GetPathName('nozzle-16.png'))
    !this.hlNozzBut.tag = |Hightlight Nozzles|
    -- Hide the frame
    !this.colour.visible = FALSE
    !this.colour.expanded = FALSE
    -- Unhighlight everything
    !!gphDrawlists.drawlists[!this.drawlist].unhighlight(!equip)
    if !this.nozzleList.rtext.set() then
      do !i index !this.nozzleList.rtext
        !!gphDrawlists.drawlists[!this.drawlist].unhighlight(!this.nozzleList.rtext[!i].dbref())
      enddo
    endif
  endif
endmethod

-------------------------------------------------------------------------------------------
-- Method .slide() - Increase the size of the clip box based on the slider                --
-------------------------------------------------------------------------------------------
define method .slide()
  -- Get the value of the slider
  !value = !this.slide.val
  -- Update the limits if required
  if !this.incl.val then
    !limits = !this.limits
    !modlimit[1] = !limits[1] - !value
    !modlimit[2] = !limits[2] + !value
    !modlimit[3] = !limits[3] - !value
    !modlimit[4] = !limits[4] + !value
    !modlimit[5] = !limits[5] - !value
    !modlimit[6] = !limits[6] + !value
    !this.volumeView.limits = !modlimit
  endif
```

```
  -- Update the clip box and refresh the view
  !this.volumeView.clipBoxXlen = !this.clipbox.box.xlength + !value
  !this.volumeView.clipBoxYlen = !this.clipbox.box.ylength + !value
  !this.volumeView.clipBoxZlen = !this.clipbox.box.zlength + !value
  !this.volumeView.refresh()
endmethod
-------------------------------------------------------------------------------------------
-- Method .enableClip() - Apply the clipbox to the view                                   --
-------------------------------------------------------------------------------------------
define method .enableClip()
  -- Get the drawlist and find the connected elements
  !drawlist = !!gphDrawlists.drawlist(!this.drawlist)
  !connectNozz = ARRAY()
  do !n index !this.nozzleList.rtext
    !nozz = !this.nozzleList.rtext[!n]
    if !nozz.dbref().cref.unset().not().and(!nozz.dbref().cref.badref().not()) then
      !connectNozz.append(!nozz.dbref().cref)
    endif
  enddo
  -- Get the volume of the equipment.  Set the clipbox to the size of the volume
  !volume = object volume(!this.equip.selection().dbref())
  !this.clipbox.box = !volume.box()
  -- If the clip is turned on
  if !this.clipBut.val then
    -- Update the icons
    !this.clipBut.tag = |Remove Connected and Disable Clipbox...|
    !this.clipPix.addPixmap(!!pml.getPathName(|nozzle-16.png|))
    !this.limitslide.visible = TRUE
    -- Loop through connected and add them to the drawlist
    do !n index !connectNozz
      !drawlist.add(!connectNozz[!n])
    enddo
    -- Update the clip box and apply it
    !this.limits = !this.volumeView.limits
    !this.clipbox.active = FALSE
    !this.clipbox.set()
    !this.clipbox.active = TRUE
    !this.volumeView.clipping = TRUE
    !this.slide()
  else
    -- Update the icons
    !this.clipBut.tag = |Add Connected and Enable Clipbox...|
    !this.clipPix.addPixmap(!!pml.getPathName(|nozzle-16.png|))
    !this.limitslide.visible = FALSE
    !this.limitslide.expanded = FALSE
    -- Remove the connected elements from the drawlist
    do !n index !connectNozz
      !drawlist.remove(!connectNozz[!n])
    enddo
    !this.clipbox.active = FALSE
    !this.volumeView.clipping = FALSE
  endif
endmethod

-------------------------------------------------------------------------------------------
-- Method .tag(GADGET) - Apply the clipbox to the view                                    --
-------------------------------------------------------------------------------------------
define method .tag(!gad is GADGET)
  -- If the gadget is a single tag - only tag the selected
  !check1 = !gad.tag.upcase().eq(|TAG SELECTED NOZZLE|)
  !check2 = !gad.tag.upcase().eq(|UNTAG SELECTED NOZZLE|)
  if !check1.or(!check2) then
    !start = !this.nozzleList.val
    !finish = !this.nozzleList.val
  else
    !start = 1
    !finish = !this.nozzleList.dtext.size()
  endif
  -- If tagging, loop through the nozzles and tag
  if !gad.val then
    do !n from !start to !finish
      if !this.tagged[!n].eq(FALSE) then
        !nozzname = !this.nozzleList.rtext[!n].dbref().flnn
        !nozzpos  = !this.nozzleList.rtext[!n].dbref().pos
        !num = 1500 + !n
        AID TEXT NUMBER $!num '$!nozzname' AT $!nozzpos
        !this.tagged[!n] = TRUE
      endif
    enddo
  else
```

```
    -- else, loop through and untag
    do !n from !start to !finish
      !num = 1500 + !n
      AID CLEAR text $!num
      handle ANY
      endhandle
      !this.tagged[!n] = FALSE
    enddo
  endif
  -- Check if all nozzles have been tagged, if so - update the icons
  !testTagged = !this.tagged
  !testTagged.unique()
  if !testtagged.size().eq(1).and(!testtagged[1].eq(TRUE)) then
    !this.tagNozzlesBut.val = TRUE
    !this.tagNozzlesBut.tag = |Untag All Nozzles|
    !this.tagNozzlesPix.addPixmap(!!pml.getPathName(|draft_lab_06.png|))
  else
    !this.tagNozzlesBut.val = FALSE
    !this.tagNozzlesBut.tag = |Tag All Nozzles|
    !this.tagNozzlesPix.addPixmap(!!pml.getPathName(|draft_lab_01.png|))
  endif
  -- Re-check
  !this.check()
endmethod
-------------------------------------------------------------------------------------------
-- Method .check() - Apply the clipbox to the view                                       --
-------------------------------------------------------------------------------------------
define method .check()
  -- If the selected is tagged, update the icons and tag
  !check = !this.tagged[!this.nozzleList.val]
  if !check then
    !this.tagNozzBut.val = TRUE
    !this.tagNozzBut.tag = |Untag Selected Nozzle|
    !this.tagNozzPix.addPixmap(!!pml.getPathName(|draft_lab_06.png|))
  else
    !this.tagNozzBut.val = FALSE
    !this.tagNozzBut.tag = |Tag Selected Nozzle|
    !this.tagNozzPix.addPixmap(!!pml.getPathName(|draft_lab_01.png|))
  endif
endmethod
-------------------------------------------------------------------------------------------
-- Method .updateRange() - Update the slider range based on user entered values         --
-------------------------------------------------------------------------------------------
define method .updateRange()
  !range[1] = !this.minslide.val
  !range[2] = !this.maxslide.val
  !range[3] = 100
  !this.slide.range = !range
  !this.slide.refresh()
endmethod

-------------------------------------------------------------------------------------------
-- Method .clear() - Tidy up and reset the form                                         --
-------------------------------------------------------------------------------------------
define method .clear()
  do !n index !this.nozzleList.rtext
    !num = 1500 + !n
    AID CLEAR text $!num
    handle ANY
    endhandle
  enddo
  !this.tagNozzBut.val = FALSE
  !this.tagNozzBut.tag = |Tag Selected Nozzle|
  !this.tagNozzPix.addPixmap(!!pml.getPathName(|draft_lab_06.png|))
  !this.tagNozzlesBut.val = FALSE
  !this.tagNozzlesBut.tag = |Tag All Nozzles|
  !this.tagNozzlesPix.addPixmap(!!pml.getPathName(|draft_lab_01.png|))
endmethod
-------------------------------------------------------------------------------------------
-- End of Form definition and methods                                                   --
-------------------------------------------------------------------------------------------
-- (c) Copyright  2010 to Current Year  AVEVA Solutions Limited                         --
-------------------------------------------------------------------------------------------
```

Appendix B7 - Example c2ex6.pmlfrm

```
-------------------------------------------------------------------------------------------
--                                                                                       --
-- (c) Copyright  2010 to Current Year  AVEVA Solutions Limited                          --
--                                                                                       --
-- File:            c2ex6.pmlfrm                                                        --
--                                                                                       --
-- Description:                                                                          --
--   Example form solution to Exercise 6                                                 --
--                                                                                       --
--   Supplied to support AVEVA training course TM1881 - PML: Form Design                 --
--                                                                                       --
-------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------
-- Form definition                                                                       --
-------------------------------------------------------------------------------------------
setup form !!c2ex6 dialog resiz

  -- Set the form title and initialisation call
  !this.formTitle     = |Equipment Checker|
  !this.initcall      = |!!this.init()|
  !this.quitcall      = |!!this.tidy()|

  -- Create a MENU object for use as a context menu on the list gadget
  !this.newMenu(|popupMenu|)
  !popUpCall = '!!CE = !this.nozzleList.selection().dbref()'
  !this.popupMenu.add('CALLBACK', 'Go to Nozzle', !popUpCall)

  -- Define the gadgets at the top of the form
  button .update linklabel |Update| call |!!this.init()| width 5
  para   .title at xmax ymin anchor t+l+r text |Available Equipments below | width 34
  line   .line at xmin.update ymax anchor t+l+r horizontal width 42

  !buttPos = |xmax - size ymax anchor b+r|
  !anch1   = |anchor t+l+r |
  !anch2   = |anchor b+l+r|

  -- Define the gadgets to choice the equipments from
  button .pick  at xmin.update ymax+0.2 pixmap width 32 height 32
  combo  .equip at xmax+0.2 ymin |Select an Equipment| tagwidth 14 $!anch1 width 22
  list   .nozzleList  at xmin.update ymax+0.5 anchor all width 40 length 5
  button .refresh linklabel |Refresh Checks| at $!buttPos width 9.6

  -- Define the gadgets to allow the attributes to be altered
  !tag = |Equipment Attributes|
  textpane .atta |$!tag| at xmin.nozzleList ymax-0.5 $!anch2 width 42 height 3.5
  button   .updateAtta linklabel |Update Attributes| at $!buttPos width 10.3

exit

-------------------------------------------------------------------------------------------
-- Constructor Method - Set the images and callbacks on the form                         --
-------------------------------------------------------------------------------------------
define method .c2ex6()
  -- Set the titles and context menu on the list
  !title = |Nozzles/Connected?/Attached?/Aligned?/Size?|
  !this.nozzleList.setHeadings(!title.split(|/|))
  !this.nozzleList.setpopup(!this.popupMenu)
  -- Pre-fill-in some attributes to be editted
  !info[1] = 'Description - Unset'
  !info[2] = 'Function - Unset'
  !info[3] = 'Purpose - Unset'
  !this.atta.val = !info
  -- Set the gadget callbacks
  !this.pick.callback  = |!this.pick()|
  !this.equip.callback = |!this.setEquip(|
  !this.refresh.callback = |!this.checkNozzles()|
  !this.updateAtta.callback = |!this.updateAtt(2)|
  -- Set the icon and tooltip of the pixmap button
  !this.pick.addPixmap(!!pml.getPathName(|onepick.png|))
  !this.pick.setTooltip(|Identify Equipment from 3D View|)
endmethod

--(Continues on next page)
```

```
-------------------------------------------------------------------------------------------
-- Initialisation Method - Collect all equi the current element and display nozzles     --
-------------------------------------------------------------------------------------------
define method .init()
  !this.collectEquipment(!!ce)
endmethod
-------------------------------------------------------------------------------------------
-- Quit Call Method - Tidy any activate pick events                                      --
-------------------------------------------------------------------------------------------
define method .tidy()
  -- Clear the any existing pick
  !!edgCntrl.remove(|Identify Equipment|)
endmethod
-------------------------------------------------------------------------------------------
-- Method .collectEquipment(DBREF) - Collect all equi for level and display nozzles     --
-------------------------------------------------------------------------------------------
define method .collectEquipment(!item is DBREF)
  -- Loop up to find a SITE or ZONE
  !allowableTypes = |ZONE SITE WORL|
  do
    break if !allowableTypes.split().findFirst(!item.type).set()
    !item = !item.owner
  enddo
  -- If we have found a SITE or ZONE, continue
  if !allowableTypes.split().findFirst(!item.type).neq(3) then
    -- Collect the equipment using PML 1 style collection syntax
    VAR !coll COLL ALL EQUIP FOR $!item
    -- If no equipment are found, offer the chance to collect for the owner
    if !coll.unset() then
      !message = |No equipment found for | & !item.type & |, |
      !message = !message & |do you wish to search the | & !item.owner.type & |?|
      if !!alert.confirm(!message).eq(|YES|) then
        !this.collectEquipment(!item.owner)
        return
      endif
    endif
    -- Update the title paragraph gadget
    !this.title.val = |Available Equipments below | & !item.type & | | & !item.flnn
    -- Evaluate the fullnames of the equis and display the information
    VAR !name EVAL FLNN FOR ALL FROM !coll
    !this.equip.dtext = !name
    !this.equip.rtext = !coll
    -- Collect the nozzles
    !this.collectNozzles()
  endif
endmethod
-------------------------------------------------------------------------------------------
-- Method .collectNozzles() - For the selected EQUI, collect all the nozzles            --
-------------------------------------------------------------------------------------------
define method .collectNozzles()
  -- Get the selected EQUI element
  !equipRef = !this.equip.selection().dbref()
  if !equipRef.unset() then
    -- If the element is unset, then clear the nozzle list
    !this.nozzleList.clear()
  else
    -- Perform a PML2 style collection for the nozzles
    !nozzColl = object COLLECTION()
    !nozzColl.type('NOZZ')
    !nozzColl.scope(!equipRef)
    !results = !nozzColl.results()
    !this.nozzleList.dtext = !results.evaluate(object block ('!results[!evalIndex].flnn'))
    !this.nozzleList.rtext = !results.evaluate(object block ('!results[!evalIndex].string()'))
    -- Check the nozzles and update the results
    !this.checkNozzles()
    !this.updateAtt(1)
  endif
endmethod
-------------------------------------------------------------------------------------------
-- Method .setEquip(GADGET, STRING) - Open callback on combo gadget to check user entry --
-------------------------------------------------------------------------------------------
define method .setEquip(!gad is gadget, !event is STRING)
  -- If the user has typed into the gadget
  if !event.eq('VALIDATE') then
    !userInput = !this.equip.displayText()

    --(Continues on next page)
```

```
    -- Loop through available equipments and choice the nearest match
    do !n index !this.equip.dtext
      !chrs = !userInput.length()
      !test = !this.equip.dtext[!n].upcase().substring(1, !chrs)
      if !userInput.upcase().eq(!test) then
        !this.equip.val = !n
        break
      endif
    enddo
  endif
  -- Recollect the nozzles on the newly selected equipment
  !this.collectNozzles()
endmethod
-------------------------------------------------------------------------------------------
-- Method .checkNozzles() - Check the collected nozzles and display the results          --
-------------------------------------------------------------------------------------------
define method .checkNozzles()
  if !this.nozzleList.rtext.set() then
    -- Loop through the collected nozzles
    do !n index !this.nozzleList.rtext
      !nozz = !this.nozzleList.rtext[!n].dbref()
      -- set the default result
      do !m from 2 to 4
        !result[!m] = |N/A|
      enddo
      -- Check 1: is the connection valid
      if !nozz.cref.unset().or(!nozz.cref.badref()) then
        !result[1] = |Check|
      else
        !result[1] = |OK|
        !end = |t|
        if !nozz.cref.href.eq(!nozz) then
          !end = |h|
        endif
        -- Check 2: is the nozzle and the same position as the attached
        if !nozz.pos.wrt( /* ).eq(!nozz.cref.attribute(!end & |pos|).wrt( /* )) then
          !result[2] = |OK|
        endif
        -- Check 3: is the nozzle and the same direction as the attached
        if !nozz.pdir[1].wrt( /* ).eq(!nozz.cref.attribute(!end & |dir|).wrt( /* )) then
          !result[3] = |OK|
        endif
        -- Check 4: is the nozzle and the same size as the attached
        if !nozz.cpar[1].eq(!nozz.cref.attribute(!end & |bore|).real()) then
          !result[4] = |OK|
        endif
      endif
      -- Record the check results for this nozzle
      !nozzleInfo[!n][1] = !nozz.flnn
      !nozzleInfo[!n].appendArray(!result)
    enddo
    -- Apply the results back to the list gadget
    !rtext = !this.nozzleList.rtext
    !this.nozzleList.setRows(!nozzleInfo)
    !this.nozzleList.rtext = !rtext
  endif
endmethod
-------------------------------------------------------------------------------------------
-- Method .updateAtt(REAL) - Check the collected nozzles and display the results         --
-------------------------------------------------------------------------------------------
define method .updateAtt(!flag is REAL)
  -- Get the selected piece of equipment and the available attributes
  !equip = !this.equip.selection().dbref()
  !availAtts = !equip.attributes()
  -- Update the textpane title
  !this.atta.tag = !equip.flnn & ' Attributes'
  !rows = !this.atta.val
  -- Loop through the rows of the textpane to extract and use the information
  !out = ARRAY()
  do !n index !rows
    -- Split the line on a "-"
    !split = !rows[!n].split('-')
    !attrib = !split[1].trim()
    -- Evaluate the available attributes so they are the same string length as the entered
    !block = object block ('!availAtts[!evalIndex].upcase().substring(1, !attrib.length())')
    !avail = !availAtts.evaluate(!block)
--(Continues on next page)

    -- If the attribute can be found, then use it
    if !avail.findfirst(!attrib.upcase()).set() then
```

```
      if !flag.eq(1) then
        -- If the flag = 1 then update the textpane with the current attribute value
        !value = !equip.attribute(!availAtts[!avail.findfirst(!attrib.upcase())])
        !out.append(!attrib & | - | & !value)
      elseif !flag.eq(2) then
        -- If the flag = 2 then assign the value to the attribute
        !val = !split[2].trim()
        !equip.attribute(!availAtts[!avail.findfirst(!attrib.upcase())]).assign(!val)
        !out.append(!attrib & | - | & !val)
      endif
    endif
  enddo
  -- Display the updated attribute list
  !this.atta.val = !out
endmethod
-------------------------------------------------------------------------------------------
-- Method .pick() - Setup and activate the pick event                                    --
-------------------------------------------------------------------------------------------
define method .pick()
  -- Clear the any existing pick
  !!edgCntrl.remove(|Identify Equipment|)
  -- Declare the EDGPACKET object
  !packet = object EDGPACKET()
  !packet.elementPick(|Identify Equipment or Nozzle <Esc> to finish|)
  !packet.description = |Identify Equipment|
  !packet.action = |!!c2ex6.pickProcess(!this.return[1].item)|
  -- Add the packet to the system and activate the pick
  !!edgCntrl.add(!packet)
endmethod
-------------------------------------------------------------------------------------------
-- Method .pickProcess(DBREF) - Process the identified equipment                         --
-------------------------------------------------------------------------------------------
define method .pickProcess(!item is DBREF)
  !allowableTypes = |EQUI NOZZ WORL|
  -- Loop up to find an EQUI/NOZZ (or the world)
  do
    break if !allowableTypes.split().findFirst(!item.type).set()
    !item = !item.owner
  enddo
  -- If we have found an EQUI, proceed...
  if !item.type.eq(|EQUI|).or(!item.type.eq(|NOZZ|)) then
    -- Find the item in the list
    if !item.type.eq(|EQUI|) then
      !val = !this.equip.rtext.findFirst(!item.string())
    else
      !expression = object EXPRESSION(|REFNO OF EQUI|)
      !equi = !expression.evaluate(!item)
      !val = !this.equip.rtext.findFirst(!equi.string())
    endif
    if !val.set() then
      -- If the equipment can be found in the list, pick it
      !this.equip.val = !val
      !this.collectNozzles()
      -- If a nozzle has been picked, ensure it is selected
      if !item.type.eq(|NOZZ|) then
        !val = !this.nozzleList.rtext.findFirst(!item.string())
        !this.nozzleList.val = !val
      endif
    else
      -- If not, offer the chance to recollect
      !message = |Identified equipment not in list.  Do you wish to re-collect?|
      if !!alert.confirm(!message).eq(|YES|) then
        -- Recollect for the owner of the identified equipment
        !this.collectEquipment(!item.owner)
      endif
    endif
  endif
endmethod
-------------------------------------------------------------------------------------------
-- End of Form definition and methods                                                    --
-------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------
```


Appendix B8 - Example traExampleExplorer.pmlfrm

```
-------------------------------------------------------------------------------------------
```

```
--                                                                                          --
-- (c) Copyright  2010 to Current Year  AVEVA Solutions Limited                             --
--                                                                                          --
-- File:            traExampleExplorer.pmlfrm                                               --
--                                                                                          --
-- Description:                                                                             --
--   Form to demonstrate the use of a NETEXPLORER PML.NET Control inside a CONTAINER        --
--                                                                                          --
--   Supplied to support AVEVA training course TM1881 - PML: Form Design                    --
--                                                                                          --
------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
-- Form definition                                                                        --
------------------------------------------------------------------------------------------
setup form !!traExampleExplorer dialog docking

  -- Load the required .dll for the explorer .NET gadget
  import 'ExplorerAddin'
  handle ANY
    -- Handle all errors, incase its already been loaded
  endhandle
  using namespace 'Aveva.Core.Presentation'

  -- Set the form title
  !this.formTitle = 'Example .NET Explorer'

  -- Create a popup menu object
  !this.newMenu(|explorerPopup|)
  !this.explorerPopup.add( 'CALLBACK', 'popup', '!!this.popup()' )

  para     .pmlTitle at x 0.2      ymax text |Standard PML gadgets|
  line     .pmlLine  at xmin - 0.1 ymax-0.5 anchor l+t+r horizontal wid 35 hei 0.7

    -- Define the functionality at the top of the form
    toggle .track            |Track CE|       at xmin + 2 ymax  anchor t+l
    button .select linklabel |Find CE >>|     at xmax ymin anchor t+r  width 8
    button .reset linklabel  |Reset to CE >>| at xmax ymin anchor t+r  width 10

  para     .netTitle at xmin.pmlTitle ymax  text |.NET Container|
  line     .netLine  at xmin.pmlLine ymax-0.5 anchor l+t+r horizontal wid 35 hei 0.7

    -- Define the container that will hold the NETEXPLORER
    !size = |wid 33 hei 15|
    container .netFrame NOBOX PMLNETCONTROL '' at xmin+1 ymax anchor all $!size

  line .line at xmin.netLine ymax+0.5 anchor b+l+r horizontal width 35
  para .footer at xcen - 0.5 * size ymax anchor b text |AVEVA Training| width 11

  -- Store the explorer control as a form member for future reference
  member .explorer  is PMLEXPLORERCONTROL
  member .element   is DBREF
exit
-- End of form definition
--
------------------------------------------------------------------------------------------
-- Constructor Method - Setup the form when it is loaded                                  --
------------------------------------------------------------------------------------------
define method .traExampleExplorer()

  -- Apply the callbacks to the PML gadgets
  !this.select.callback = |!!this.explorer.selectElement(!!ce.name)|
  !this.track.callback  = |!!this.explorer.setFollowCe(!this.track.val)|
  !this.reset.callback  = |!!this.explorer.resetRoot(!!ce.name)|

  -- Specify the namespace before using the .NET gadget
  using namespace 'Aveva.Core.Presentation'

  -- Create an instance of the control
  !this.explorer = object PMLEXPLORERCONTROL()
  !this.netFrame.control = !this.explorer.handle()
  -- Register for the OnPopup event
  !this.explorer.addeventhandler('OnPopup', !this, 'rightClickExplorer')
  --(Continues on next page)

  -- Use methods on Explorer C# control
  !this.explorer.initialise('/*','')
endmethod
-- End of Constructor Method
--
------------------------------------------------------------------------------------------
```

```
-- Method .rightClickExplorer(ARRAY) - Show context menu on explorer                    --
-------------------------------------------------------------------------------------------
define method .rightClickExplorer(!data is ARRAY)
  !this.netFrame.popup = !this.explorerPopup
  !this.netFrame.showPopup(!data[0], !data[1])
  !this.element = !data[2].dbref()
endmethod
-- End of Method .rightClickExplorer(ARRAY)
--
-------------------------------------------------------------------------------------------
-- Method .popup() - Query the name of the selected element                             --
-------------------------------------------------------------------------------------------
define method .popup()
  q var !this.element.name
endmethod
-- End of Method .popup()
--
-------------------------------------------------------------------------------------------
-- End of Form definition and methods                                                   --
-------------------------------------------------------------------------------------------
-- (c) Copyright  2010 to Current Year  AVEVA Solutions Limited                         --
-------------------------------------------------------------------------------------------
```

Appendix B9 - Example traExampleGridControl.pmlfrm

```
-------------------------------------------------------------------------------------------
--                                                                                       --
-- (c) Copyright  2010 to Current Year  AVEVA Solutions Limited                          --
--                                                                                       --
-- File:            traExampleGridControl.pmlfrm                                         --
--                                                                                       --
-- Description:                                                                          --
--   Form to demonstrate the use of a Grid Control PML.NET Control inside a CONTAINER    --
--                                                                                       --
--   Supplied to support AVEVA training course TM1881 - PML: Form Design                 --
--                                                                                       --
-------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------
-- Form definition                                                                       --
-------------------------------------------------------------------------------------------
setup form !!traExampleGridControl dialog docking

  -- Load the required .dll for the explorer .NET gadget
  import 'GridControl'
  handle ANY
    -- Handle all errors, incase its already been loaded
  endhandle
  using namespace 'Aveva.Core.Presentation'

  -- Set the form title and the initialisation call
  !this.formTitle = |Example .NET Grid Control|
  !this.initcall  = |!this.init()|

  -- Create a popup menu object
  !this.newMenu(|gridPopup|)
  !this.gridPopup.add('CALLBACK', 'Add to 3D View', '!this.addToThreeDView()')
  !this.gridPopup.add('CALLBACK', 'Save to Excel...', '!this.saveToExcel()')
  !this.gridPopup.add('CALLBACK', 'Import from Excel...', '!this.loadFromExcel()')

  -- Spefify the container that will hold the grid control .NET gadget
  container .gridFrame NOBOX PMLNETCONTROL 'grid' anchor all width 60 height 20

  line .line at xmin ymax+0.5 anchor b+l+r horizontal width 60
  para .footer at xcen - 0.5 * size ymax anchor b text |AVEVA Training| width 11

  member .grid     is NETGRIDCONTROL
  member .elements is ARRAY

exit
-- End of form definition
--
-------------------------------------------------------------------------------------------
-- Constructor Method - Setup the form when it is loaded                                 --
-------------------------------------------------------------------------------------------
define method .traExampleGridControl()

  -- Specify the namespace before using the .NET gadget
  using namespace 'Aveva.Core.Presentation'

  -- Create an instance of the control
  !this.grid = object NetGridControl()
  !this.gridFrame.control = !this.grid.handle()

  -- Apply the event handlers
  !this.grid.addeventhandler('OnPopup', !this, 'rightClickGrid')
  !this.grid.addeventhandler('AfterSelectChange', !this, 'afterSelectChange')
  !this.grid.addeventhandler('BeforeCellUpdate', !this, 'beforeCellUpdate')
endmethod
-- End of Constructor Method
--
-------------------------------------------------------------------------------------------
-- Initialisation Method - Setup the Grid Control on the form                            --
-------------------------------------------------------------------------------------------
define method .init()

  -- Specify the namespace before using the .NET gadget
  using namespace  'Aveva.Core.Presentation'
  -- Create headings
  !headings = |Name Type Owner Description|
  --(Continues on next page)
```

```
  -- Create model items for population of grid
  var !data collect all EQUIP
  var !data append collect all PIPES

  -- Bind data to grid
  !nds = object NETDATASOURCE('Grid Control Example', !headings.split(), !data)
  !this.grid.bindToDataSource(!nds)

  -- Set grid parameters
  !this.grid.columnExcelFilter(TRUE)
  !this.grid.setNameColumnImage()
  !this.grid.outlookGroupStyle(TRUE)
  !this.grid.fixedHeaders(FALSE)
  !this.grid.fixedRows(FALSE)
  !this.grid.columnSummaries(TRUE)
  !this.grid.autoFitColumns()
  --!this.grid.editableGrid(TRUE)
  --!this.grid.setEditableColumn(4,TRUE)
  --!this.grid.setColumnColor(4, 'white')
endmethod
-- End of Initialisation Method
--
-------------------------------------------------------------------------------------------
-- Method .rightClickGrid(ARRAY) - Show the context menu on the control                  --
-------------------------------------------------------------------------------------------
define method .rightClickGrid(!data is ARRAY)
  !this.gridFrame.popup = !this.gridPopup
  !this.gridFrame.showPopup(!data[0], !data[1])
  !this.elements = !data[2]
endmethod
-- End of Initialisation Method
--
-------------------------------------------------------------------------------------------
-- Method .addToThreeDView() - Adds the selected elements to the drawlist               --
-------------------------------------------------------------------------------------------
define method .addToThreeDView()
  do !element values !this.elements
    add $!element
  enddo
endmethod
-- End of Method .addToThreeDView()
--
-------------------------------------------------------------------------------------------
-- Method .saveToExcel() - Save the grid to excel via the file browser                  --
-------------------------------------------------------------------------------------------
define method .saveToExcel()

   -- Ensure the correct .dll is loaded
   import 'PMLFileBrowser'
   handle ANY
   endhandle

   -- Specify the namespace before using the .NET gadget
   using namespace 'Aveva.Core.Presentation'

   -- Show the browser form and based on the file, save the grid
   !browser = object PMLFILEBROWSER('SAVE')
   !browser.show('C:\','Example.xls','Save to Excel',false, 'Excel Files|*.xls|,1)
   !this.grid.saveGridToExcel(!browser.file())
endmethod
-- End of Method .saveToExcel()
--
-------------------------------------------------------------------------------------------
-- Method .loadFromExcel() - Load the grid from excel via the file browser              --
-------------------------------------------------------------------------------------------
define method .loadFromExcel()

   -- Ensure the correct .dll is loaded
   import 'PMLFileBrowser'
   handle ANY
   endhandle

   -- Specify the namespace before using the .NET gadget
   using namespace 'Aveva.Core.Presentation'

   --(Continues on next page)
```

```
   -- Show the browser form and based on the file, fill the grid
   !browser = object PMLFILEBROWSER('OPEN')
   !browser.show('C:\','','Load Grid from Excel',true, 'Excel Documents|*.xls',1)
   !this.grid.clearGrid()
   !nds = object NETDATASOURCE('Grid Control Example', !browser.file())
   !this.grid.BindToDataSource(!nds)
endmethod
-- End of Method .loadFromExcel()
--
-------------------------------------------------------------------------------------------
-- Method ..afterSelectChange(ARRAY) - Empty method, ready to be called by the control  --
-------------------------------------------------------------------------------------------
define method .afterSelectChange(!data is ARRAY)

endmethod
-- End of Method .loadFromExcel()
--
-------------------------------------------------------------------------------------------
-- Method ..beforeCellUpdate(ARRAY) - Perform a DB update with the supplied data        --
-------------------------------------------------------------------------------------------
define method .beforeCellUpdate(!data is ARRAY)
  !this.grid.doDabaconCellUpdate(!data)
endmethod
-- End of Method .beforeCellUpdate(ARRAY)
--
-------------------------------------------------------------------------------------------
-- End of Form definition and methods                                                    --
-------------------------------------------------------------------------------------------
-- (c) Copyright  2010 to Current Year  AVEVA Solutions Limited                          --
-------------------------------------------------------------------------------------------
```

Appendix B10 - Example c2ex9.pmlfrm

```
-------------------------------------------------------------------------------------------
--                                                                                       --
-- (c) Copyright  2009 to Current Year  AVEVA Solutions Limited                          --
--                                                                                       --
-- File:            c2ex9.pmlfrm                                                         --
--                                                                                       --
-- Description:                                                                          --
--   Example form solution to Exercise 9                                                 --
--                                                                                       --
--   Supplied to support AVEVA training course TM1881 - PML: Form Design                 --
--                                                                                       --
-------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------
-- Form definition                                                                       --
-------------------------------------------------------------------------------------------
setup form !!c2ex9 dialog docking

  -- Load the required .dll for the explorer .NET gadget
  import 'GridControl'
  handle ANY
  endhandle
  import 'ExplorerAddin'
  handle ANY
  endhandle
  using namespace 'Aveva.Core.Presentation'

  -- Set the title and initialisation call for the form
  !this.formTitle       = |Equipment Viewer|
  !this.iconTitle       = |VIEWER|
  !this.callback        = |!this.init()|
  !this.firstShownCall  = |!this.firstShown()|
  !this.killingCall     = |!this.close()|
  !this.formRevision    = |1.0|

  -- Create a popup menu object
  !this.newMenu(|apopup|)
   !this.apopup.add('CALLBACK', 'Navigate to Element', '!ce = !this.selection')
  !this.newMenu(|gridPopup|)
    !this.gridPopup.add('CALLBACK', 'Navigate To Nozzle.', '!this.showNozzle()')

  -- Set up the container gadgets - Notice the use of variables
  container .explorerFrame NOBOX PMLNETCONTROL anchor t+1  width 35 height 10
  !pos  = |at xmin.explorerFrame ymax.explorerFrame|
  !size = |width 35 height 10|
  container .attributeFrame NOBOX PMLNETCONTROL $!pos anchor t+1+b $!size

  frame .viewFrame panel at xmax + 0.5 ymin.explorerFrame anchor ALL
    view .volumeView  VOLUME
      width 50 height 20
      border off
      shading on
      isometric 3
    exit
  exit

  -- Specify the user-defined form members
  member .attributeGrid   is NETGRIDCONTROL
  member .designExplorer  is PMLEXPLORERCONTROL
  member .selection       is DBREF
  member .drawlist        is REAL
exit
-------------------------------------------------------------------------------------------
-- Constructor Method                                                                    --
-------------------------------------------------------------------------------------------
define method .c2ex9()
  -- Specify Namespace for .Net
  using namespace 'Aveva.Core.Presentation'

  -- Setup the .Net Controls
  !this.designExplorer = object PMLEXPLORERCONTROL()
  !this.attributeGrid = object NETGRIDCONTROL()
  !this.explorerFrame.control = !this.designExplorer.handle()
  !this.attributeFrame.control = !this.attributeGrid.handle()

--(Continues on next page)
```

```
  -- add PML event handlers to .Net control
  !this.designExplorer.addeventhandler('OnPopup', !this, 'rightClickExplorer')
  !this.designExplorer.addeventhandler('OnSelectionChanged',!this, 'updateData')
  !this.designExplorer.initialise('/EQUIP','')
  !this.attributeGrid.addeventhandler('OnPopup', !this, 'rightClickGrid')

  -- Create a local drawlist within the global object
  !this.drawlist = !!gphDrawlists.createDrawList()
endmethod
-------------------------------------------------------------------------------------------
-- FirstShown Method                                                                     --
-------------------------------------------------------------------------------------------
define method .firstShown()
  -- Add 3D view to view system (this shows the form)
  !!gphViews.add(!this.volumeView)

  -- add local drawlist to 3D view
  !!gphDrawlists.attachView(!this.drawlist, !this.volumeView)
  !!gphDrawlists.drawlists[!this.drawlist].holes(TRUE)
endmethod
-------------------------------------------------------------------------------------------
-- Initialisation Method                                                                 --
-------------------------------------------------------------------------------------------
define method .init()
  -- Setup the Grid Control Gadget
  !this.attributeGrid.columnExcelFilter(FALSE)      $* Turn off the filters
  !this.attributeGrid.outlookGroupStyle(FALSE)      $* Turn off outlook grouping
  !this.attributeGrid.fixedHeaders(FALSE)           $* Turn off fixed headers
  !this.attributeGrid.fixedRows(FALSE)              $* Turn off fixed rows
  !this.attributeGrid.columnSummaries(FALSE)        $* Turn off summaries
  !this.attributeGrid.setLabelVisibility(TRUE)      $* Show Label
  !this.attributeGrid.clearGrid()                   $* Clear any existing data
endmethod
-------------------------------------------------------------------------------------------
-- Close Method                                                                          --
-------------------------------------------------------------------------------------------
define method .close()
  -- On unloading the form, remove drawlist from view and delete it
  !!gphDrawlists.detachView(!this.volumeView)
  !!gphDrawlists.deleteDrawlist(!this.drawlist)
endmethod
-------------------------------------------------------------------------------------------
-- Method .updateSystemData(ARRAY) - runs when the explorer is clicked                   --
-------------------------------------------------------------------------------------------
define method .updateData(!data is ARRAY)

  -- Establish the chosen element in the explorer
  !this.selection = !data[0].dbref()

  -- loop to find an EQUI.
  do
    break if !this.selection.type.eq(|EQUI|).or(!this.selection.type.eq(|WORL|))
    !this.selection = !this.selection.owner
  enddo

  -- If an EQUI is found, then fill in the information
  if !this.selection.type.eq(|EQUI|) then
    !this.fillAttributes()
  endif

endmethod
-------------------------------------------------------------------------------------------
-- Method .rightClickExplorer(ARRAY) - runs when the explorer is right clicked           --
-------------------------------------------------------------------------------------------
define method .rightClickExplorer(!data is ARRAY)
  -- Apply the popup to the container
  !this.explorerFrame.popup = !this.apopup

  -- Show the popup menu
  !this.explorerFrame.showPopup(!data[0], !data[1])
endmethod
-------------------------------------------------------------------------------------------
-- Method .rightClickGrid(ARRAY) - runs when the grid is right clicked                   --
-------------------------------------------------------------------------------------------
define method .rightClickGrid(!data is ARRAY)

 --(Continues on next page)
```

```
  -- Apply the popup to the container
  !this.attributeFrame.popup = !this.GridPopup

  -- Show the popup menu
  !this.attributeFrame.showPopup(!data[0], !data[1])
endmethod
-------------------------------------------------------------------------------------------
-- Method .showNozzle() - runs when the user selected from the popup                     --
-------------------------------------------------------------------------------------------
define method .showNozzle()
  !!ce = !this.attributegrid.getselectedrows()[1][1].dbref()
endmethod
-------------------------------------------------------------------------------------------
-- Method .fillAttributes() - fills the grid will information                            --
-------------------------------------------------------------------------------------------
define method .fillAttributes()

  -- Specify Namespace for .Net
  using namespace  'Aveva.Core.Presentation'

  -- Set the Views limits based on the chosen element
  !!gphViews.limits(!this.volumeView, !this.selection)

  -- Update the drawlist by removing the previous and adding the chosen
  !drawlist = !!gphDrawlists.drawlist(!this.drawlist)
  !drawlist.removeall()
  !drawlist.add(!this.selection)

  -- Active the volume view
  !this.volumeView.active = TRUE

  -- Fill in the Grid Control Gadget
  -- Collect all the NOZZs.  PML1 syntax used as array of strings needed
  var !data collect all NOZZ for $!this.selection

  -- Attributes required for each nozzle to be filled in
  !attributes = |NAME CPAR[1] CREF|

  -- The titles of the columns for the attributes
  !titles = |Nozzle Size Connection|

  -- Required information passed to a NetDataSource object to compile the info
  !nds = object NETDATASOURCE('Grid', !attributes.split(), !titles.split(), !data)
  !this.attributeGrid.bindToDataSource(!nds)

  -- Set the name column to hold icons of DB elements and autofit the columns
  !this.attributeGrid.setNameColumnImage()
  !this.attributeGrid.autoFitColumns()

  -- Alter the information in the third column (connected)
  -- Define the icons for use in the column
  !error = !!pml.getPathName('ad_t_019.png')
  !ok    = !!pml.getPathName('ad_t_018.png')

  -- Collect the current values in column three and loop through them
  !connected = !this.attributeGrid.GetColumn(3)
  do !n index !connected
    !testValue = !this.attributeGrid.getCell(!n, 3)
    !element = !this.attributeGrid.getcell(!n, 1).dbref()
    if !testValue.eq(|Nulref|) then                    $* if found Null element
      !this.attributeGrid.setCellImage(!n, 3, !error)  $* apply the missing icon
      !this.attributeGrid.setCellValue(!n, 3, |Check!|) $* change the cell value
      !this.attributeGrid.setCellColor(!n, 3, |white|)
      !!gphDrawlists.drawlists[!this.drawlist].highlight(!element, 2)
    else
      !this.attributeGrid.setCellImage(!n, 3, !ok)     $* apply the attached icon
      !!gphDrawlists.drawlists[!this.drawlist].highlight(!element, 5)
    endif
  enddo
endmethod
-------------------------------------------------------------------------------------------
-- End of Form definition and methods                                                    --
-------------------------------------------------------------------------------------------
-- (c) Copyright  2010 to Current Year  AVEVA Solutions Limited                          --
-------------------------------------------------------------------------------------------
```

Appendix B11 - Example apppml.pmlobj

PML addin file

```
name:          PMLTraining
title:         PML Training
showOnMenu:    FALSE
object:        appPML
```

apppml.obj

```
define object appPML
endobject

define method .modifyMenus()
 !this.utilitiesMenu()
endmethod

define method .modifyForm()
endmethod

define method .utilitiesMenu()
  !menu = object APPMENU('SYSUTIL')
  !menu.add('SEPARATOR')
  !menu.add('FORM', |Calculator|, |c2ex1|)
  !menu.add('MENU', |Equipment Checker|, 'EquipCheck')
  !menu.add('SEPARATOR')
  !!appMenuCntrl.addMenu(!menu, 'EQUI')

  !menu= object APPMENU('EquipCheck')
  !menu.add('FORM', |Exercise 4...|, |c2ex4|)
  !menu.add('FORM', |Exercise 5...|, |c2ex5|)
  !!appMenuCntrl.addMenu(!menu, 'EQUI')
endmethod
```