Work with Parcels More Efficiently

Create simple tools with ArcObjects

By Lang Deng, Ph.D., Pioneer Hi-Bred International

The author has created several useful parcel searching tools in Visual Basic for Applications (VBA) using ArcObjects. This article gives detailed descriptions of tools illustrated by the required code. Any ArcGIS Desktop user, with an ArcInfo, ArcEditor, or ArcView license, can use this code to easily create the same tools. The sample map document containing these tools and sample data to run them are available from the ArcUser Online *site.*

Story County, Iowa, has been using ArcInfo to create parcels since 1996. When ArcSDE was adopted in 2003, it became easier and more cost-effective for county agencies to share base data on parcels, roads, annotations, water bodies, and zoning data.

Parcel data is most frequently used. Requests for this data come from staff members working in the Assessor, Auditor, Planning and Zoning, Engineering, E911, Sheriff's, and other offices. Consequently, a user-friendly tool in ArcMap to search for a parcel by ID, acreage, value, owner, address, or other attribute will save time for many people.

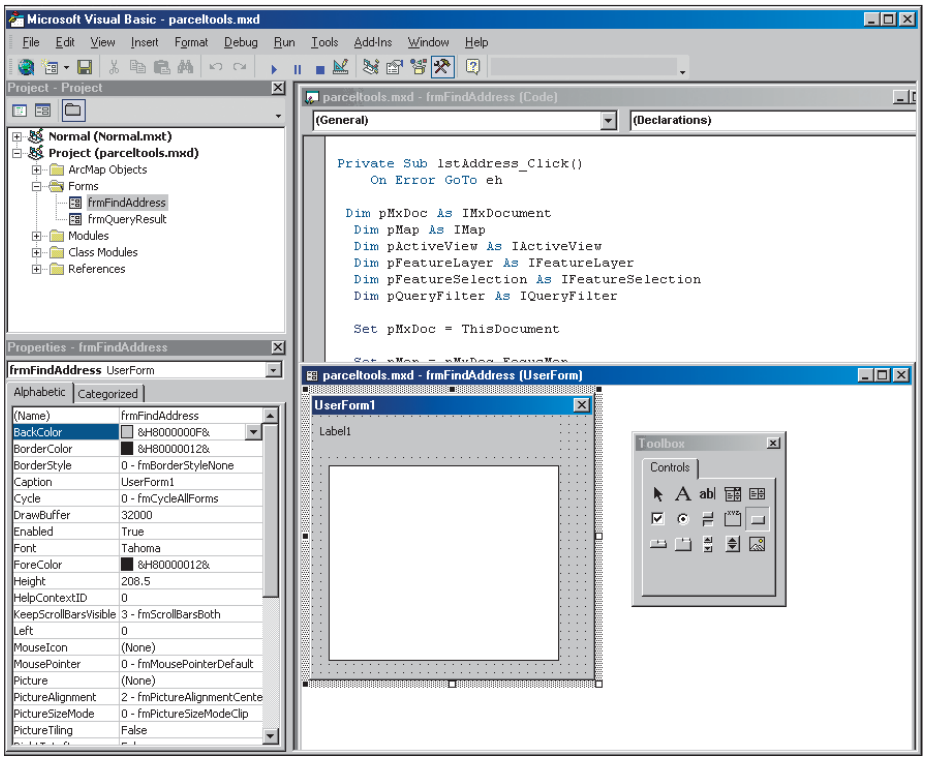
The Find by Parcel Address Tool

First, think through the whole process of finding a parcel by its address—step by step. What will need to happen? Of course, testing will be required to ensure the code executes properly but good planning saves time and requires writing less code.

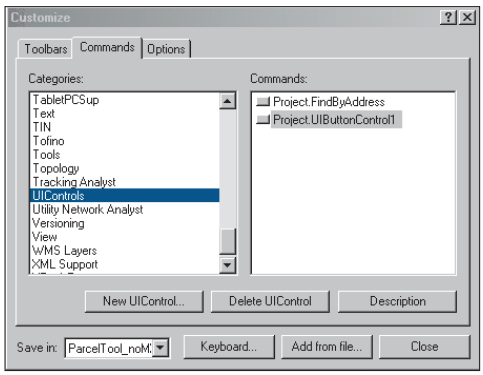
The goal is to find a parcel by address. A user may enter a partial address or a full address. A full address may have only one match and if that entry is found, the next step is to simply zoom directly to that parcel. However, a partial address may return multiple matches. All matches will need to be shown to the user so the user can select one. In this case, a list is needed to hold the returned matches.

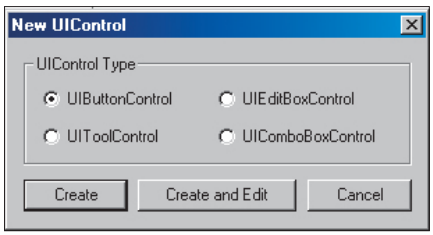
A list box on a form will do the job, so the next step is to create a form. From the ArcMap standard menu, choose Tools > Macros > Visual Basic Editor. From the Visual Basic Editor menu, choose Insert > add Userform and name the form frmFindAddress.

If the toolbox is not displayed, choose View > Toolbox. Click the list box on the toolbox to add it to the form. Name it lstAddress. Adjust the size and look of the form and the size of the list box to ensure it is wide enough to hold the data from the address and owner fields. Add a label above the list box that will display the search result that gives simple instructions about how to zoom to any of the entries in the list box.



Add a form and create a list box.





From the ArcMap interface, choose Customize > Commands and add an UIbuttonControl. Name it FindbyAddress and add it to a toolbar.

On the form, create a button and attach the code that will find the parcel by its address. From the ArcMap interface, choose Customize > Commands and add a UIbuttonControl. Name it FindByAddress and add it to a toolbar.

Right-click on the control and choose View Code to open FindByAddress_Click event in VBA editor. Listing 1 shows the first section of code that selects on the parcel layer to find a particular parcel.

To ensure the parcel layer is in the ArcMap Table of Contents, use a Boolean variable to report if the target layer, a parcel layer in this case, was found. A parcel feature class can be named anything by the user after it is added to

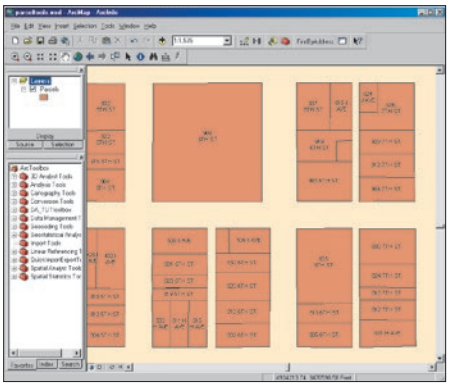
```
Private Sub FindByAddress_Click()
    On Error GoTo eh 'it's a good idea to always use error handler,
that way, you don't get complaints from users
   'who see the debug warning message.

  Dim pMxDoc As IMxDocument
  Dim pMap As IMap
  Dim pActiveView As IActiveView
  Dim pFeatureLayer As IFeatureLayer
  Dim pFeatureSelection As IFeatureSelection
  Dim pQueryFilter As IQueryFilter

  Set pMxDoc = ThisDocument

  Set pMap = pMxDoc.FocusMap
  Set pActiveView = pMap 'if the focus is Layout window, it works
just as well

  'use EnumLayer to find a parcel layer
```

Listing 1

```
  Dim LayerFound As Boolean
  LayerFound = False

        Dim pDataset As IDataset
        Dim pFeatureLayer1 As IFeatureLayer
       Dim X As Long
         Dim pMatchLayer As ILayer
        Dim LayerName1 As String
        Dim LayerName2 As String
        Dim LayerName3 As String
        Dim pDataset As IDataset
        Dim pFeatureLayer1 As IFeatureLayer

'The following are the names of parcel views inside SDE
        LayerName1 = "gis_data.gis.parcel_refgis"
        LayerName2 = "gis_data.GIS.parcels"
        LayerName3 = "gis_data.gis.parcel_view"

 For X = 0 To pMap.LayerCount - 1 'loop through all layers in TOC and
compare each layer's name with the Parcel 'Featureclass or Parcel
Views in SDE
        Set pMatchLayer = pMap.Layer(X)

        'check the source of parcels layer

        Set pFeatureLayer1 = pMatchLayer
        Set pDataset = pFeatureLayer1.FeatureClass

     If UCase(pDataset.Name) = UCase(LayerName1) Or _
         UCase(pDataset.Name) = UCase(LayerName2) Or _
         UCase(pDataset.Name) = UCase(LayerName3) Or _
         UCase(pMatchLayer.Name) = UCase("Parcels") Then
         LayerFound = True
         Exit For 'when the parcel layer is found, exit loop,
otherwise keeps on moving to the next layer
      End If
    Next

   'Add a parcel layer here from SDE, if not connected to SDE, give
option _
    'to add from any other source
 If LayerFound = False Then
    MsgBox "No Parcel Layer Found in TOC, Parcels from SDE will be
added to your ArcMap", vbInformation, "ADD Parcel Layer"
   ' Call AddDatafromSDE.AddSDELayer2ArcMap 'calls a subroutine to
add the parcels layer to TOC
    'set  pMatchLayer as the newly added layer
    Set pMatchLayer = pMap.Layer(0)
  End If
```

Listing 2

First, think through the whole process of finding a parcel by its address—step by step. What will need to happen?



Using the sample map document with FindByAddress tool added to a button.

ArcMap so the name of the parcel layer can't be controlled. However, because it is added to the Table of Contents from ArcSDE or as a shapefile, the layer's source name can be checked to determine if it is a parcel layer.

At Story County, many views have been created based on the Parcels feature class in ArcSDE. Any of these views, in addition to the feature class called Parcels, can be added to ArcMap and renamed. To allow the find parcel tool to work on any parcel view from ArcSDE, get all the names of parcel views in ArcSDE and check to see if any of them was added to the Table of Contents. The source name of a layer in ArcMap is the name of the view or feature class in ArcSDE. To get that name, use IDataset interface. It has a name property that returns the layer's name in ArcSDE. The code in Listing 2 tests for a parcel layer.

The code in Listing 3 passes the parcel layer to the form, verifies that the query field (e.g., the ADDRESS field) exists, prompts the user for a search string, and searches for an address. Adding an asterisk (*) before and after the search string will find any address that contains part of the search string. For example, if the search string is 456, it will return not only "456 4th St" (a match) but also "1456 3rd St", "4561 Washington Ave", and "32 456th St". The more specific the search string, the fewer results will be returned and the quicker the parcel will be located. However, with just part of a street name, this search method will guarantee that the parcel will be found. The part of the address used will affect the number of returns. In this example, typing "900" will find

Continued on page 34

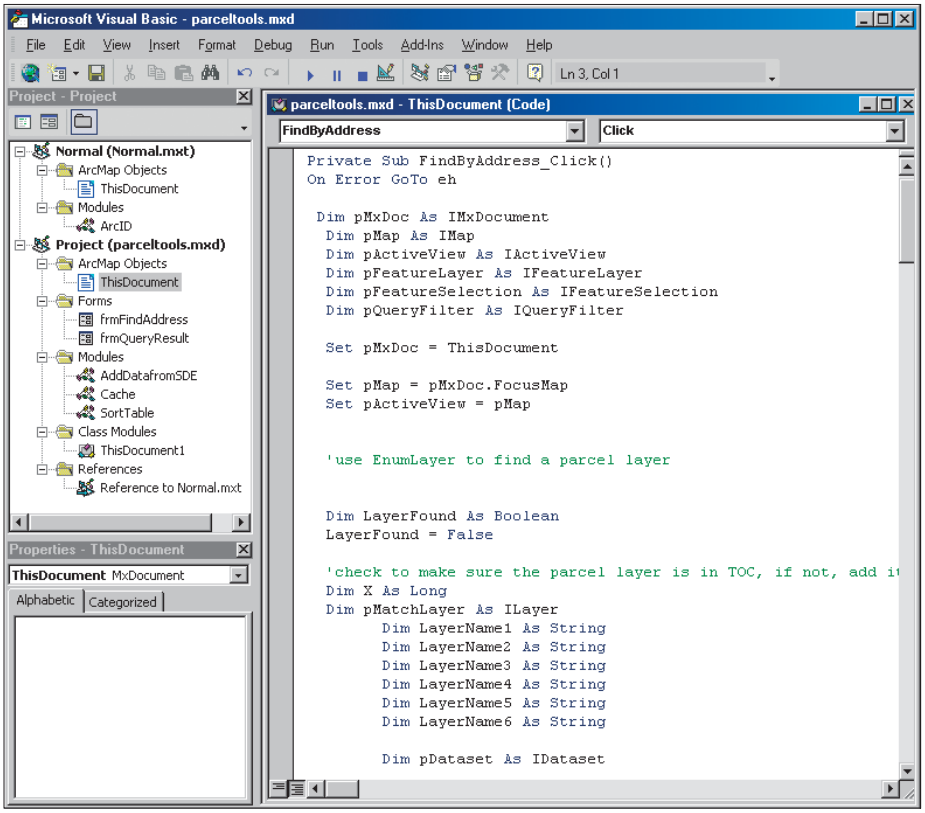
Work with Parcels More Efficiently

Continued from page 33

32 matches while typing in "6th St" will return 442 results. Typing "900 6th St"will generate only one result and will zoom the user to that parcel.

Listing 4 shows the code that performs the selection and zooms to one of the selected features. If more than one feature is found, it adds all the selected features to the list box. VBA has no sort function but one has been added to this tool. The SortTable function shown in Listing 5 was written by Brad Posthumus and Kirk Kuykendall and shared through the ArcGIS Desktop Discussion Conference on the ESRI Support site (support.esri.com). See *ArcUser Online* for links to these postings.

When more than one parcel is located, a click event in the list box will be needed to zoom to a specific record in the list box. The code shown in Listing 6 lets the user click an item in the lstAddress list box to go to a desired parcel. Clicking on a record in the list box will cause the corresponding parcel to be immediately displayed in ArcMap.



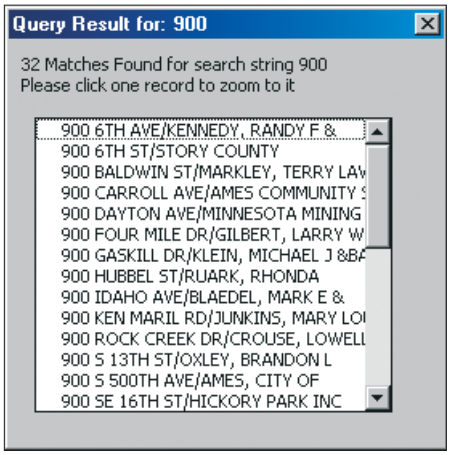
Code in VBA for the FindByAddress tool.

Creating Other Parcel Tools

After looking at the FindByAddress parcel tool, a FindbyParcel ID tool is a lot easier to create because it searches for a unique ID and returns only one match each time. A list box is not needed, and the tool highlights and zooms

```
  Set pFeatureLayer = pMatchLayer
    Set pFeatureSelection = pFeatureLayer 'QI, query interface to set the FeatureSelection object
Set pWorkFeatureLayer = pMatchLayer 'pass the parcel layer so the forms can use it, pWorkFeatureLayer
is a module level variable, otherwise, forms cannot see (use) it

   'Check to see if the query field exists, if not, let user know
   If pFeatureLayer.FeatureClass.FindField("ADDRESS") = -1 Then
    MsgBox "The Address field is not found!"
    Exit Sub
    End If

          'Create the query filter
       Set pQueryFilter = New QueryFilter
       Dim StrName As String

     'give some instruction on how to insert search string, actually they may enter anything and get a
result
StrName = InputBox("Please Enter An Address or Part of it" & vbNewLine & _
         "If you enter 900 or Lincoln, then all addresses include 900 or Lincoln will be returned", _
        "FIND ADDRESS", "900 6th st")
     If StrName = "" Then 'if nothing entered or user canceled it, give a warning
          MsgBox "Nothing Entered, or you clicked Cancel", vbInformation, "Try Again"
            Exit Sub
       End If

       pQueryFilter.WhereClause = "ADDRESS LIKE '" & "*" & StrName & "*'"
```

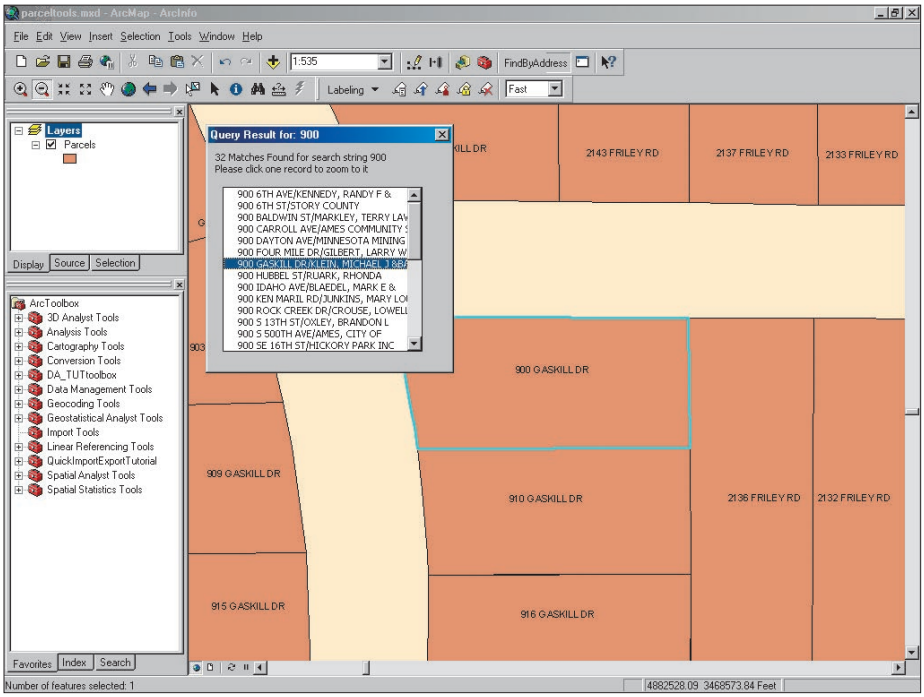
Listing 3



If more than one address is found, the list box holds all the addresses so the user can choose one to zoom to.

directly to the parcel it finds. Other tools, such as FindByOwner and FindByValue, can be created in a manner similar to the FindByAddress tool—just change the field name the search will be run against.

Continued on page 36



When the user clicks on one of the returned addresses, that parcel is selected and the map display zooms to that parcel.

```
'Flag the original selection
        pActiveView.PartialRefresh esriViewGeoSelection, Nothing, Nothing
  'Perform the selection
       pFeatureSelection.SelectFeatures pQueryFilter, esriSelectionResultNew, False


    If pFeatureSelection.SelectionSet.Count < 1 Then
        MsgBox "No Match Found", vbInformation, "Please try another address"
        Exit Sub
    End If

'zoom to the selected feature(s), we want to zoom to a single feature, not all features selected
        Dim pFeatureClass As IFeatureClass
        Set pFeatureClass = pFeatureLayer.FeatureClass

        Dim pFCursor As IFeatureCursor  ' need a featurecursor to get to each individual feature
        Set pFCursor = pFeatureClass.Search(pQueryFilter, True)

        Dim pFeature As IFeature
        Set pFeature = pFCursor.NextFeature

'if there is only one match, then we can zoom to that feature directly
        If pFeatureSelection.SelectionSet.Count = 1 Then
            Dim pEnvelope As IEnvelope
            Set pEnvelope = pFeature.Extent 'set the  pEnvelope to the extent of the selected feature
            pEnvelope.Expand 2.5, 2.5, True 'Expand the extent of the feature like to zoom out a little
bit
            pActiveView.Extent = pEnvelope
            pActiveView.Refresh
         Exit Sub
        End If

        'If there is more than one record found, need to add all of the selected to _
        'a list box, but sort them before adding to a list since VBA has no built in sorting method
```

Listing 4

Work with Parcels More Efficiently

Continued from page 35

Conclusion

The FindByAddress parcel tool, together with other similar parcel tools such as FindByOwner and FindByParcelID, has been used by Story County department heads and staff. The county Assessor's Office found the FindByAddress tool particularly useful during the annual property value review session. For more information on these tools, contact

Lang Deng
IM Leader
Pioneer Hi-Bred International
Tel.: 515-334-6390
E-mail: lang.deng@pioneer.com

About the Author

Dr. Lang Deng, currently a GIS specialist/IM leader at Pioneer Hi-Bred International, was the GIS coordinator at Story County, Iowa.

Acknowledgments

Special thanks to the Assessor's Office, Story County, Iowa, for providing the data and permission to publish the code used in creating these tools. All the parcel information used in this article is publicly accessible at www.storyassessor.org/pmc/ Contact Tammy Gardner at 515-382-7320 or tammy.gardner@storycounty.com with any additional questions. Thanks also to Brad Posthumus and Kirk Kuykendall for the SortTable function.

```
        frmFindAddress.Caption = "Query Result for " & StrName 'change the form caption and to
   'show the result from the selection with a label

        frmFindAddress.Label1.Caption = pFeatureSelection.SelectionSet.Count & _
        " Matches Found for " & StrName & vbNewLine & "Please click one record to select and zoom to it"

        Do Until pFeature Is Nothing 'loop through all features returned and add the needed information
to listbox
          frmFindAddress.lstAddress.AddItem pFeature.Value(pFeature.Fields.FindField("ADDRESS")) & _
            "/" & pFeature.Value(pFeature.Fields.FindField("Deed_holde"))

'Add owner (Deed_holde) to the list to help users easily figure out which parcel they need to select

          Set pFeature = pFCursor.NextFeature
        Loop

        Call SortTable.SortListBox(frmFindAddress.lstAddress, 0)     'call the function to sort the
listbox
        frmFindAddress.Show   'show the form and list box after being sorted

 Exit Sub
eh:
  MsgBox "Find Parcel Error- " & Err.Description
The code in Listing  sorts  Put the following in a module

Option Explicit
'pWorkFeatureLayer pass the parcel layer from ThisDocument to forms
Public pWorkFeatureLayer As IFeatureLayer


The following SortTable function is from Mr. Brad Posthumus (http://forums.esri.com/Thread.asp?c=93&f=99
3&t=138741#402818) and Mr.
Kirk Kuykendall (http://forums.esri.com/Thread.asp?c=93&f=992&t=56843&mc=2#msgid142518) at ESRI Forum.
Special thanks to them.

Public Sub SortListBox(lstListbox As ListBox, nSortColumn As Integer)

' sort the listbox
Dim i As Long, j As Long, nColumn As Long, v As Variant

For i = 0 To UBound(lstListbox.List) - 1
    For j = i + 1 To UBound(lstListbox.List)
        If lstListbox.List(i, nSortColumn) > lstListbox.List(j, nSortColumn) Then
            For nColumn = 0 To lstListbox.ColumnCount - 1
                v = lstListbox.List(i, nColumn)
                lstListbox.List(i, nColumn) = lstListbox.List(j, nColumn)
                lstListbox.List(j, nColumn) = v
            Next nColumn
        End If
    Next j
Next i

End Sub
```

Listing 5

```
Private Sub lstAddress_Click()
    On Error GoTo eh

 Dim pMxDoc As IMxDocument
  Dim pMap As IMap
  Dim pActiveView As IActiveView
  Dim pFeatureLayer As IFeatureLayer
  Dim pFeatureSelection As IFeatureSelection
  Dim pQueryFilter As IQueryFilter

  Set pMxDoc = ThisDocument

  Set pMap = pMxDoc.FocusMap
  pMap.ClearSelection
  Set pActiveView = pMap

    Set pFeatureSelection = pWorkFeatureLayer 'passed from the findAddress_click

     Dim pFeatureClass As IFeatureClass
    Set pFeatureClass = pWorkFeatureLayer.FeatureClass
    'MsgBox pFeatureClass.AliasName

    Set pQueryFilter = New QueryFilter

   Dim ListString As String
   ListString = frmFindAddress.lstAddress.Text

    pQueryFilter.WhereClause = "ADDRESS = '" & _
    Left(ListString, InStr(ListString, "/") - 1) & "'" 'we just need the address part before "/", the
owner name after "/" needs to be excluded

     pActiveView.PartialRefresh esriViewGeoSelection, Nothing, Nothing
  'Perform the selection
       pFeatureSelection.SelectFeatures pQueryFilter, esriSelectionResultNew, False
  'Flag the new selection

    Dim pFCursor As IFeatureCursor
        Set pFCursor = pFeatureClass.Search(pQueryFilter, True)

        Dim pFeature As IFeature
        Set pFeature = pFCursor.NextFeature
        Dim pEnvelope As IEnvelope
        Set pEnvelope = pFeature.Extent

        pEnvelope.Expand 2.5, 2.5, True
        pActiveView.Extent = pEnvelope
        pActiveView.Refresh

 Exit Sub
eh:
  MsgBox "Find Owner Error- " & Err.Description

End Sub
```

Listing 6