Appendix 4-K

Redistribution of Overage Land Use Acres in Model Cells

Redistribution of Overage Land Use Acres in Model Cells

Appendix 4.K contains the Visual Basic program used to redistribute dryland acres (and in rare cases groundwater acres) in excess of 160 acres. The overages resulted from county-scaled adjustments applied to 160 acre model cells, which caused some cells to exceed 160 acres. The overages were redistributed to cells for a given county and year that contained rangeland. The dryland (or in rare cases grounwater irrigated land) replaced the rangeland. The process was heirarchial, as the overage was redistributed first to cells with a small amount of rangeland, and any leftover overage was redistributed to cells with increasingly larger portions of rangeland until the overage was completely redistributed. The program was written by Rick Vollertson and was developed with Excel software.

```
--------------------------------------------------------------------------------
‘Subdistribute()
‘Created: August 1st, 2012
‘Author: Rick Vollertson, Applications Developer, Nebraska Department of
‘Natural Resources (NDNR)
‘Purpose: To distribute "overage acres" (Dryland) listed in Worksheet "OV" by
County/years.
‘Method:  Check each cell for available Rangeland acres to be converted to
Dryland acres. Field Rangeland is sorted Small to Large. Two worksheets used
OV and 1950 to thru 1959.


' Variables
Dim OV As Double
Dim AV As Double
Dim StartTime As Double
Dim StopTime As Double
Dim RunTime As Double
Dim icount As Integer
Dim Yr As String
Dim Yrs As String
Yrs = "1"
Yr = "1990"

StartTime = [now()] 'ActiveCell.FormulaR1C1 = Format(Now, "hh:mm") StartTime
= Format(Now, "h:mm")
Application.Workbooks("Acres1990.xlsm").Activate

'Sort worksheet Year by County A to Z and Rangeland Small to Largest
 Sheets(Yr).Select
 Range("A2").Select
ActiveWorkbook.Worksheets(Yr).Sort.SortFields.Clear
```

```
ActiveWorkbook.Worksheets(Yr).Sort.SortFields.Add Key:=Range("A2:A33484") _
, SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
ActiveWorkbook.Worksheets(Yr).Sort.SortFields.Add Key:=Range("F2:F33484") _
SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
   With ActiveWorkbook.Worksheets(Yr).Sort
      SetRange Range("A1:J33484")
      .Header = xlYes
      .MatchCase = False
      .Orientation = xlTopToBottom
      SortMethod = xlPinYin
      Apply
   End With
‘Preset Pointer
Sheets(Yr).Select
Range("D2").Select
Sheets("OV").Select
Range("C2").Select
OV = ActiveCell.Value
Cnty = ActiveCell.Offset(0, -1).Value
For icount = 1 To 1
   Do While IsEmpty(ActiveCell.Value) = False
      Sheets(Yr).Select
      AV = ActiveCell.Offset(0, 2).Value
      Cnty2 = ActiveCell.Offset(0, -3).Value
         Do While Cnty = Cnty2
            If OV <> 0 Then
               If (AV - OV) > 0 Then
                  ActiveCell.Offset(0, 2).Value = AV - OV
                  ActiveCell.Value = OV + ActiveCell.Value
                  OV = 0
                  Cnty2 = ActiveCell.Offset(0, -3).Value
               Else
                  ActiveCell.Value = AV + ActiveCell.Value
                  ActiveCell.Offset(0, 2) = 0
                  OV = OV - AV
                  ActiveCell.Offset(1, 0).Select
                  AV = ActiveCell.Offset(0, 2).Value
                  Cnty2 = ActiveCell.Offset(0, -3).Value
               End If
            Else
               AV = ActiveCell.Offset(0, 2).Value
               Sheets("OV").Select
               ActiveCell.Offset(1, 0).Select
               Cnty = ActiveCell.Offset(0, -1).Value
               OV = ActiveCell.Value
               Yrs = ActiveCell.Offset(0, -2).Value
               Yr = Yrs
            End If
         Loop
         Do While Cnty <> Cnty2
            If IsEmpty(ActiveCell.Offset(0, -2).Value) = True Then
               Exit Do
            End If
         Sheets(Yrs).Select
         Range("D1").Select
         ActiveCell.Offset(0, -3).Select
         Cells.Find(What:=Cnty, After:=ActiveCell, LookIn:=xlFormulas _
         , LookAt:=xlPart, SearchOrder:=xlColumns, SearchDirection:=xlNext, _
         MatchCase:=False, SearchFormat:=False).Activate
         ActiveCell.Offset(0, 3).Select
         AV = ActiveCell.Offset(0, 2).Value
         Sheets("OV").Select
         Cnty2 = ActiveCell.Offset(0, -1).Value
```

```
         OV = ActiveCell.Value
      Loop
   Loop
   If IsEmpty(ActiveCell.Offset(0, -2).Value) = True Then
      Exit For
   End If
   Yr = Yr + 1

'Sort worksheet Year by County A to Z and Rangeland Small to Largest
 Sheets(Yr).Select
 Range("A2").Select
ActiveWorkbook.Worksheets(Yr).Sort.SortFields.Clear
ActiveWorkbook.Worksheets(Yr).Sort.SortFields.Add Key:=Range("A2:A33484") _
, SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
ActiveWorkbook.Worksheets(Yr).Sort.SortFields.Add Key:=Range("F2:F33484") _
SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
   With ActiveWorkbook.Worksheets(Yr).Sort
      SetRange Range("A1:J33484")
      .Header = xlYes
      .MatchCase = False
      .Orientation = xlTopToBottom
      SortMethod = xlPinYin
      Apply
   End With

   Sheets(Yr).Select
   Range("D2").Select
   Sheets("OV").Select
   ActiveCell.Offset(0, 1).Value = OV
   ActiveCell.Offset(1, 0).Select
   OV = ActiveCell.Value
   Cnty = ActiveCell.Offset(0, -1).Value
Next icount
‘Runtime Msgbox to show start and stop and duration time.
StopTime = [now()] 'ActiveCell.FormulaR1C1 = Format(Now, "h:mm") StopTime =
Format(Now, "h:mm")
RunTime = StopTime - StartTime
MsgBox "Done" & vbNewLine & "Process Time " & Format(RunTime, "h:mm") &
vbNewLine & "Start " & Format(StartTime, "h:mm") & vbNewLine & "Stop  " &
Format(StopTime, "h:mm")
End Sub
```