

BATCH REPORTING OF FOREST INVENTORY STATISTICS USING THE EVALIDATOR

Patrick D. Miles¹

Abstract—The EVALIDator Web application, developed in 2007, provides estimates and sampling errors of forest statistics (e.g., forest area, number of trees, tree biomass) from data stored in the Forest Inventory and Analysis database. In response to user demand, new features have been added to the EVALIDator. The most recent additions are 1) the ability to generate multiple reports in a single retrieval (batch reporting); 2) the flexibility to report change components (growth, removals, and mortality) by stand and/or tree classification values as recorded at the time of the first or second measurement; and 3) the ability to generate reports of ratio estimates that have pages, rows, and columns. Information on the data and methods used is provided along with sample output from a query that demonstrates the new batch feature.

INTRODUCTION

The EVALIDator Web-application was designed to simplify the generation of population estimates—and their associated sampling errors—from data in the Forest Inventory and Analysis database (FIADB) (USDA Forest Service 2015). The EVALIDator guides the user, via a graphical user interface (GUI), to select 1) the desired attribute estimate; 2) the area of interest; 3) the page, row, and column classifiers; and 4) additional filtering. The user can generate desired output in as few as seven mouse clicks. Depending on the type of retrieval and the size of the geographic area queried, the output will be generated in from several seconds to several minutes. Over 50,000 retrievals were completed using this method in 2014.

The EVALIDator GUI was designed to prevent users from obtaining estimates that are not possible given the underlying data set. For example, forest land volume estimates for inventories collected prior to the annual inventory design (Bechtold and Patterson 2005) may not be available because tree measurements

were often not collected on reserved and unproductive forest land. Therefore, if the user selects volume on forest land as the estimate of interest using the EVALIDator GUI, the user will be presented with only a list of those inventories where this estimate is appropriate (usually only on annual inventories completed after 1998).

There are situations in which a user might need to run hundreds or thousands of retrievals for an analysis. The siting of a mill, for example, may require a “wood basket” analysis where several estimates would be needed for several hundred possible mill locations. The analysis might also include a sensitivity analysis where the radius of the wood basket may vary from 50 to 100 miles from the prospective mill locations. A different approach is required for this type of analysis. Users cannot be expected to use the EVALIDator GUI interface thousands of times waiting from seconds to minutes after each run to save the output for future use. Thus an application programming interface (API) was developed for the EVALIDator web-application. This API can be integrated into an MS-Excel® macro-enabled spreadsheet to enable batch processing of these estimates.

¹Pat Miles, Research Forester, Forest Service, Northern Research Station Forest Inventory and Analysis, 1992 Folwell Ave. Saint Paul, MN 55108, (651) 649-5146, pmiles@fs.fed.us

Because the EVALIDator API bypasses the EVALIDator’s GUI, there is an increased likelihood that the user will produce forest statistics that are not supported by the data. For example, the user could try to produce an estimate of the volume on forest land for the 1977 inventory of Minnesota. However, the 1977 Minnesota inventory predates the annual inventory implementation, so tree measurements would not have been taken on reserved and unproductive forest land. The estimate returned would be based only on productive nonreserved forest land, thereby underreporting the desired number. It is suggested that a trial run of the EVALIDator GUI be used to verify that an estimate is supported by the data.

METHODS

The EVALIDator API requires 16 parameters (Table 1). Figure 1 shows how this information can be used to invoke the EVALIDator Web application. This example contains the URL (first line) and the parameters (subsequent lines) necessary to generate an estimate of the “Area of timberland, in acres” by “Stand-size class”, and “Ownership group – Major”, within “50” miles of latitude “45” degrees North, and “-93” degrees West.

Copying and pasting a series of URLs containing these 16 parameters would be awkward and time consuming. An MS-Excel macro was created to demonstrate how URLs and parameters could

```

http://apps.fs.fed.us/Evalidator/batcheval.jsp?
reptype=Circle
&lat=45.0
&lon=-93.0
&radius=50
&snum=Area of timberland, in acres
&sdenom=No denominator - just produce estimate.
&wc=272014,552014
&pselected=None
&rselected=Stand-size class
&cselected=Ownership group - Major
&ptime=Current
&rtime=Current
&ctime=Current
&wf=
&wnum=
&wnumdenom=

```

Figure 1. Example parameters for the EVALIDator API. Note: all this information would be on a single line and copied to the browser address line.

Table 1. EVALIDator API input parameters.

Parameter	Valid values
reptype	“State”, “Circle”
lat	0.0000 to 90.0000 (decimal degrees NAD83)
lon	-180.0000 to 180.0000 (decimal degrees NAD83)
radius	0.0 to 1000.0 (units are in miles)
snum	See values of attribute_descr variable in ref_pop_attribute FIADB table
sdenom	If not performing a ratio estimate then enter “No denominator - just produce estimate.” For ratio estimates see values of attribute_descr variable in ref_pop_attribute FIADB table.
wc	See values of eval_grp variable in pop_eval_grp FIADB table. When more than one evaluation group is selected the evaluation group numbers should be separated by a comma.
pselected	See values of label_var variable in validator_variable_library table where page_list='Y'. If pages breakdowns are not desired enter “None”
rselected	See values of label_var variable in validator_variable_library table where row_list='Y'.
cselected	See values of label_var variable in validator_variable_library table where col_list='Y'.
ptime	“Accounting”, “Previous”, “Current”, “Previous if available else current”, “Current if available else previous” Note: Always use “Current” unless generating growth, removals or mortality estimates.
rtime	“Accounting”, “Previous”, “Current”, “Previous if available else current”, “Current if available else previous” Note: Always use “Current” unless generating growth, removals or mortality estimates.
ctime	“Accounting”, “Previous”, “Current”, “Previous if available else current”, “Current if available else previous” Note: Always use Current unless generating growth, removals or mortality estimates.
wf	SQL clause filter used for non-ratio estimates.
wnum	SQL clause filter - only applied to numerator in a ratio estimate .
wnumdenom	SQL clause filter - applied to both numerator and denominator in a ratio estimate.

be submitted as a batch retrieval. An MS-Excel workbook named “BatchInternetEvaluator.xlsm” containing this macro can be downloaded from a link on the FIA Data and Tools page (<http://www.fia.fs.fed.us/tools-data/default.asp>).

When the workbook is downloaded and opened, a worksheet named “Sheet1” will appear (Fig. 2). On this worksheet, there is a button labeled “QueryEvaluatorWeb-application”. Clicking this button will result in the execution of the macro named “getMillLocationEstimate,” which will then create nine html pages (there are three mill locations listed on Sheet1 and three estimates will be created for each mill) that are each automatically opened and displayed in MS-Excel. Users can modify the code in the macro to generate additional estimates.

The MS-Excel macro “getMillLocationEstimate”, as currently written, will read information for up to 1000 mill locations¹ from the MS-Excel worksheet labeled “Sheet 1.” Mill location information is contained in the columns labeled “Latitude” and “Longitude.” The radius in miles of the circular area is contained in the column labeled “Radius.” The column labeled “EvaluationGroups” identifies the inventories to be used in this retrieval. In this example 272013 refers

to the Minnesota (Federal Information Processing Standards state code = 27) 2013 inventory and 552013 refers to the Wisconsin 2013 inventory. For each mill location, the macro will generate three population estimate reports. The first report is for the “Area of timberland, in acres” (see line 12 in appendix) by “Forest type” (see line 38 in appendix) and by “Ownership group – Major” (see line 39 in appendix). The second report is for the “Net volume of live trees (at least 5 inches d.b.h./d.r.c.), in cubic feet, on timberland” (see lines 13 and 14 in appendix). The third report is a ratio report. It reports the “Average annual net growth of live trees (at least 5 inches d.b.h./d.r.c.), in cubic feet, on timberland” divided by the “Average annual removals of live trees (at least 5 inches d.b.h./d.r.c.), in cubic feet, on timberland.” It should be noted that this is for the area that was timberland at both the time of the current and previous inventories as this provides a more realistic ratio estimates of the actual removals that have occurred on lands that remained in the timberland base. The code is executed by left-clicking on the “QueryEvaluatorWeb-application” button on Sheet1 with the computer’s mouse device. The output from this example is stored in nine html files. The filenames consist of the MillName combined with the estimate number.

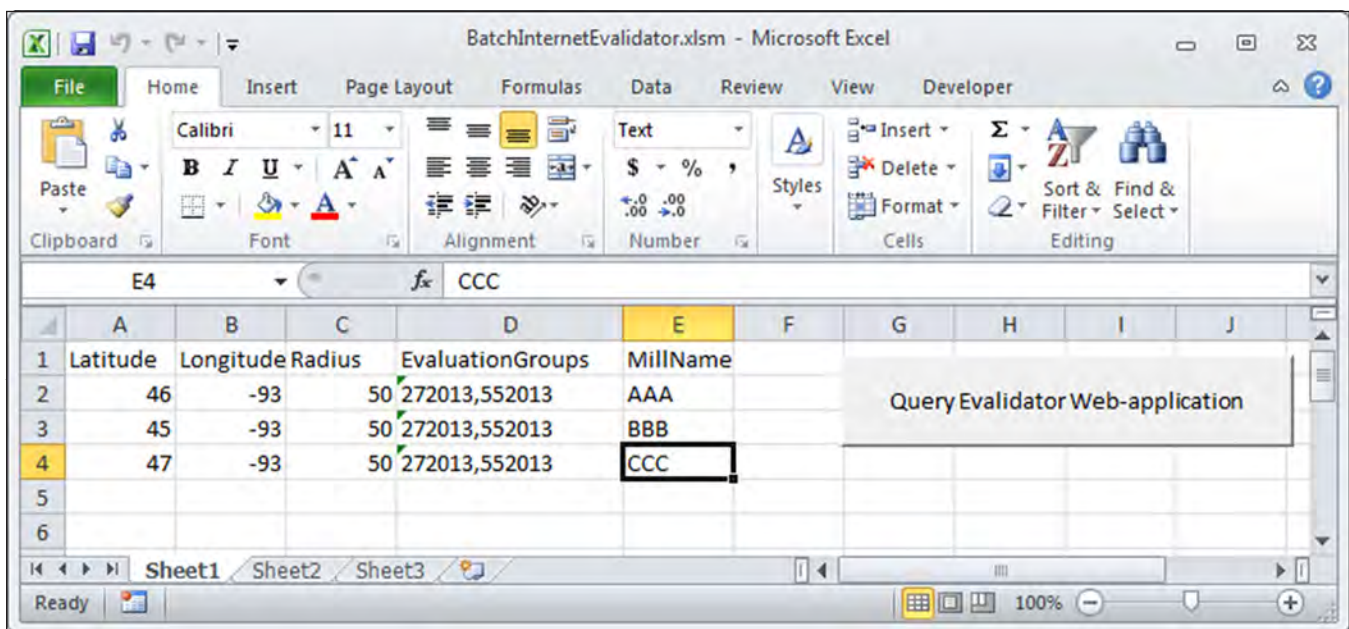


Figure 2. MS-Excel Workbook “BatchInternetEvaluator.xlsm” Sheet1 with “QueryEvaluatorWeb-application” button to initiate EVALIDator batch runs.

RESULTS

The code in Figure 1, when entered into a browser's address line, will return an html page that can be copied and pasted into an MS-Excel worksheet. The output from this query is depicted in Table 2. The total area of timberland within 50 miles of a site located at 45 degrees North and -93 degrees West is 1.0 million acres. The sampling error for this estimate, based on one standard deviation, is 4.8 percent or +/- 50,000 acres. This estimate is based on 445 FIA timberland plots.

LITERATURE CITED

- Bechtold, W.A.; Patterson, P.L., eds. 2005. The enhanced Forest Inventory and Analysis program – national sampling design and estimation procedures. Gen. Tech. Rep. SRS-80. Asheville, NC: U.S. Department of Agriculture Forest Service, Southern Research Station. 85 p.
- USDA Forest Service. 2015. The Forest Inventory and Analysis database: database description and user guide. Version 6.0.1 for Phase 2. Washington, DC: U.S. Department of Agriculture, Forest Service. 748 p. Available: <http://www.fia.fs.fed.us/library/database-documentation/>. (Accessed June 1, 2015.)

APPENDIX

Workbook BatchInternetEvaluator.xlsm macro getMillLocationEstimate is triggered by clicking on the "QueryEvaluatorWeb-application" button. The code for the macro is provided below.

```
Sub getMillLocationEstimate()
    Dim mill_lat As Double
    Dim mill_lon As Double
    Dim mill_radius As Double
    Dim evalGroups As String
    Dim mill_id As String
```

Table 2. Results from running query depicted in Figure 1.

Numerator type Area of timberland, in acres
 Statecd/EVALID(s):
 Wisconsin 552014
 Minnesota 272014
 Page variable=None (based on values from the Current inventory).
 Row variable=Stand-size class (based on values from the Current inventory).
 Column variable=Ownership group - Major (based on values from the Current inventory).
 Circle retrieval centered at 45.0 degrees north and -93.0 degrees west with a radius of 50 miles.
 Filtering clause(s):

Estimate:

Stand-size class	Ownership group - Major		
	Total	Public	Private
Total	1,040,428	90,646	949,782
Large diameter	566,608	39,853	526,755
Medium diameter	290,017	18,116	271,901
Small diameter	168,573	32,676	135,897
Nonstocked	15,230	-	15,230

Sampling error percent:

Stand-size class	Ownership group - Major		
	Total	Public	Private
Total	4.8	16.99	5.02
Large diameter	6.51	24.82	6.76
Medium diameter	8.96	34.37	9.29
Small diameter	12.29	27.5	13.69
Nonstocked	31.95	-	31.95

Number of non-zero plots in estimate:

Stand-size class	Ownership group - Major		
	Total	Public	Private
Total	445	39	409
Large diameter	252	20	232
Medium diameter	150	10	140
Small diameter	76	15	62
Nonstocked	11	-	11

```

Dim sURL As String
Dim sResult As String
Dim urlStr As String
Dim numerator(3) As String
Dim denominator(3) As String

numerator(1) = "Area of timberland, in acres"

numerator(2) = "Net volume of live trees (at least 5
inches d.b.h./d.r.c.), in cubic feet, on timberland"

numerator(3) = "Average annual net growth of live
trees (at least 5 inches d.b.h./d.r.c.), in cubic feet, on
timberland"

denominator(1) = "No denominator - just produce
estimate."

denominator(2) = "No denominator - just produce
estimate."

denominator(3) = "Average annual removals of live
trees (at least 5 inches d.b.h./d.r.c.), in cubic feet, on
timberland"

For i = 2 To 1001 'maximum number of mills is set
to 1000

    For j = 1 To 3

        mill_lat = ThisWorkbook.Sheets("sheet1").
Range("a" + LTrim(Str(i)))

        mill_lon = ThisWorkbook.Sheets("sheet1").
Range("b" + LTrim(Str(i)))

        mill_radius = ThisWorkbook.Sheets("sheet1").
Range("c" + LTrim(Str(i)))

        evalGroups = ThisWorkbook.Sheets("sheet1").
Range("d" + LTrim(Str(i)))

        mill_id = ThisWorkbook.Sheets("sheet1").
Range("e" + LTrim(Str(i)))

        If mill_id <> "" Then

```

```

urlStr = "http://apps.fs.fed.us/Evalidator/
batcheval.jsp?"

urlStr = urlStr + "reptype=Circle"

urlStr = urlStr + "&lat=" + Str(mill_lat)

urlStr = urlStr + "&lon=" + Str(mill_lon)

urlStr = urlStr + "&radius=" + Str(mill_radius)

urlStr = urlStr + "&snum=" + numerator(j)

urlStr = urlStr + "&sdenom=" + denominator(j)

urlStr = urlStr + "&wc=" + evalGroups

urlStr = urlStr + "&pselected=None"

urlStr = urlStr + "&rselected=Forest type"

urlStr = urlStr + "&cselected=Ownership group -
Major"

urlStr = urlStr + "&ptime=Current"

urlStr = urlStr + "&rtime=Current"

urlStr = urlStr + "&ctime=Current"

urlStr = urlStr + "&wf="

urlStr = urlStr + "&wnum="

urlStr = urlStr + "&wnumdenom="

sResult = GetHTTPResult(urlStr)

    Call OpenTextFile(mill_id + "_" + Format(Str(j),
"00"), sResult)

    End If

    Next j

Next i

End Sub

Function GetHTTPResult(sURL As String) As String

Dim XMLHTTP As Variant, sResult As String

```

```

Set XMLHTTP = CreateObject("WinHttp.
WinHttpRequest.5.1")

XMLHTTP.SetTimeouts "300000", "300000",
"300000", "300000" 'timeout= 300 seconds

XMLHTTP.Open "GET", sURL, False

XMLHTTP.Send

'Debug.Print "Status: " & XMLHTTP.Status & " - "
& XMLHTTP.StatusText

sResult = XMLHTTP.ResponseText

'Debug.Print "Length of response: " & Len(sResult)

Set XMLHTTP = Nothing

GetHTTPResult = sResult

End Function

Sub OpenTextFile(mill_id As String, sResult As
String)

Dim File_Path As String, folder_path As String

File_Path = Application.ActiveWorkbook.Path + "\" +
mill_id + ".html"

Open File_Path For Output As #1

Write #1, sResult

Close #1

ChDir _

Application.ActiveWorkbook.Path

Workbooks.Open Filename:= _

Application.ActiveWorkbook.Path + "\" + mill_id
+ ".html"

End Sub

```