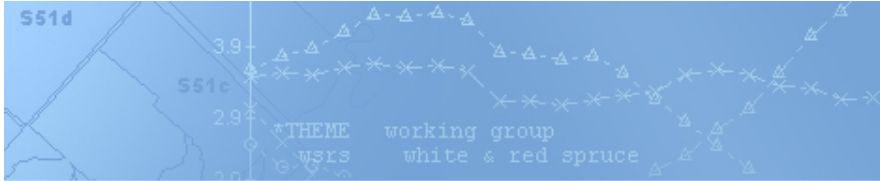


**Remsoft®**

# Yields and DLLs – Making Life Easy

Ugo Feunekes

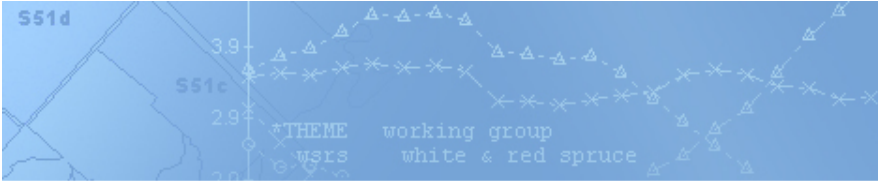
VP, Research and Development



## *Purpose*

**Address the following types of questions on DLLs**

- What are they?
- Different types
- How can they be used?
- What's currently in Woodstock?



## *What is a DLL?*

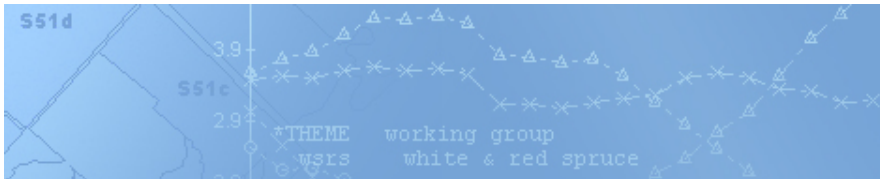
### **Dynamic Link Library:**

“A file containing executable code and data bound to a program at load time or run time.”

[www.sabc.co.za/manual/ibm/9agloss.htm](http://www.sabc.co.za/manual/ibm/9agloss.htm)

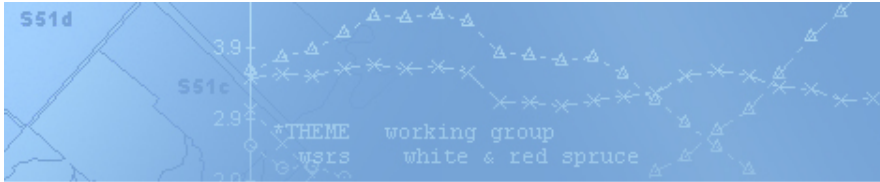
“Executable routines usually serving a specific function or set of functions to be stored separately as files with the extension .dll, and to be loaded only when called by the program that needs them.”

[www.ism01.west.asu.edu/ISM411Carey/notes/glossary.html](http://www.ism01.west.asu.edu/ISM411Carey/notes/glossary.html)



## *Woodstock and DLLs*

- Software code that connects to Woodstock
- Plug-in application
- Only used in the YIELD section
- Requires a compiler to make DLL –  
FORTRAN, PASCAL, C++, Visual Basic, etc.



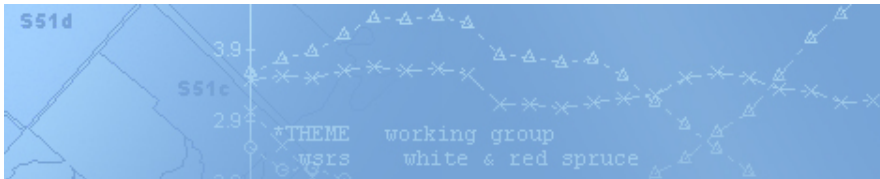
## *History of DLLs in Woodstock*

### WHEN?

- Dates back to mid 90's
- Idea came when WK was DOS; needed to go to 16 bit Windows for workgroups to implement

### WHY?

- Temple-Inland in Texas had detailed G&Y and didn't want to use static yield tables
- Inventory system used DLLs
- Objective: Data consistency - ensure that WK and Inventory were using the same values



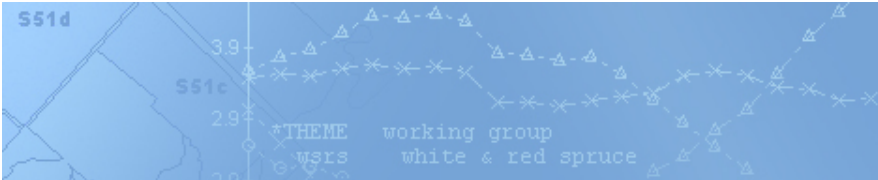
## *History of DLLs in Woodstock*

Thereafter....

- Expanded development of DLLs in Yield section
- “Fixed” and “Variant” DLLs
- Internal system DLLs - easier to implement applications through DLLs as opposed to software code

Use today across Remsoft users?

- Probably less than 10%

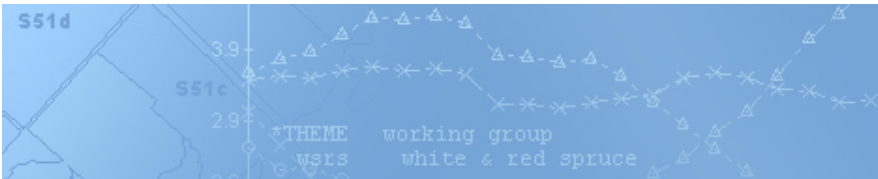


## What are the disadvantages and advantages?

### Disadvantages:

- Software development
- Easy to make big and nasty models
- Can take longer to run
- Yield coefficient visibility, potential for “black box” (\_YIELD report)
- Models with few forest strata and silvicultural regimes may not receive benefit<sup>1</sup>
- Models with complex growth models may be prohibitive<sup>1</sup>

<sup>1</sup>An innovative, dynamic yield table creation process for forest estate modeling, M.L. McLarin, T.E. Osborn, L.J. Bennett, Forestry Tasmania, Hobart, TAS

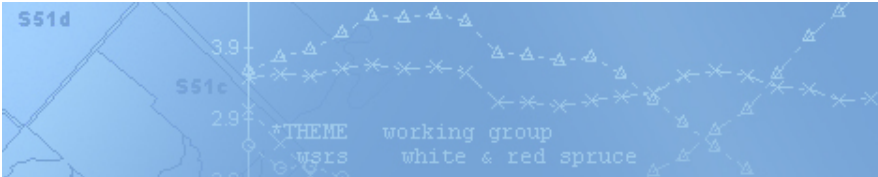


## What are the disadvantages and advantages?

### Advantages:

- Easy to change model
- Yield sections are small – 50 lines vs 25 Mb files
- Harder to make mistakes – e.g. missed yield curve
- Flexible depending on how it is developed – e.g. flip between periodic to annual model
- DLLs save time and effort – remove yield “nastiness”
- Data consistency – ensures all staff are using the latest inputs available for decision making<sup>1</sup>
- Facilitates the use of continuous variables and the creation of more yields, rather than lumping into broad strata<sup>1</sup>

<sup>1</sup>An innovative, dynamic yield table creation process for forest estate modeling, M.L. McLarin, T.E. Osborn, L.J. Bennett, Forestry Tasmania, Hobart, TAS



# How they work?

- Used during Yield set build
- WK calls the DLL to pass numbers to functions or sub routines within the DLL which returns the coefficients.

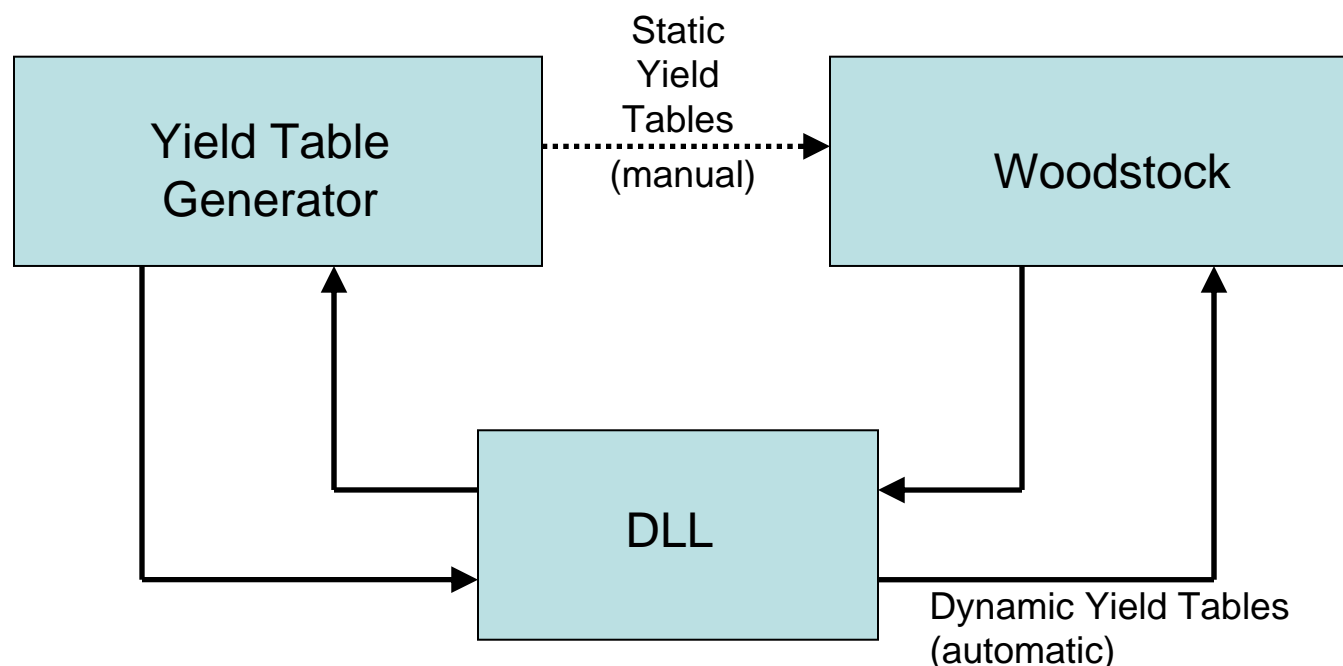
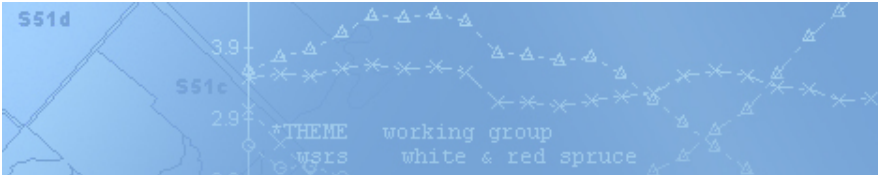
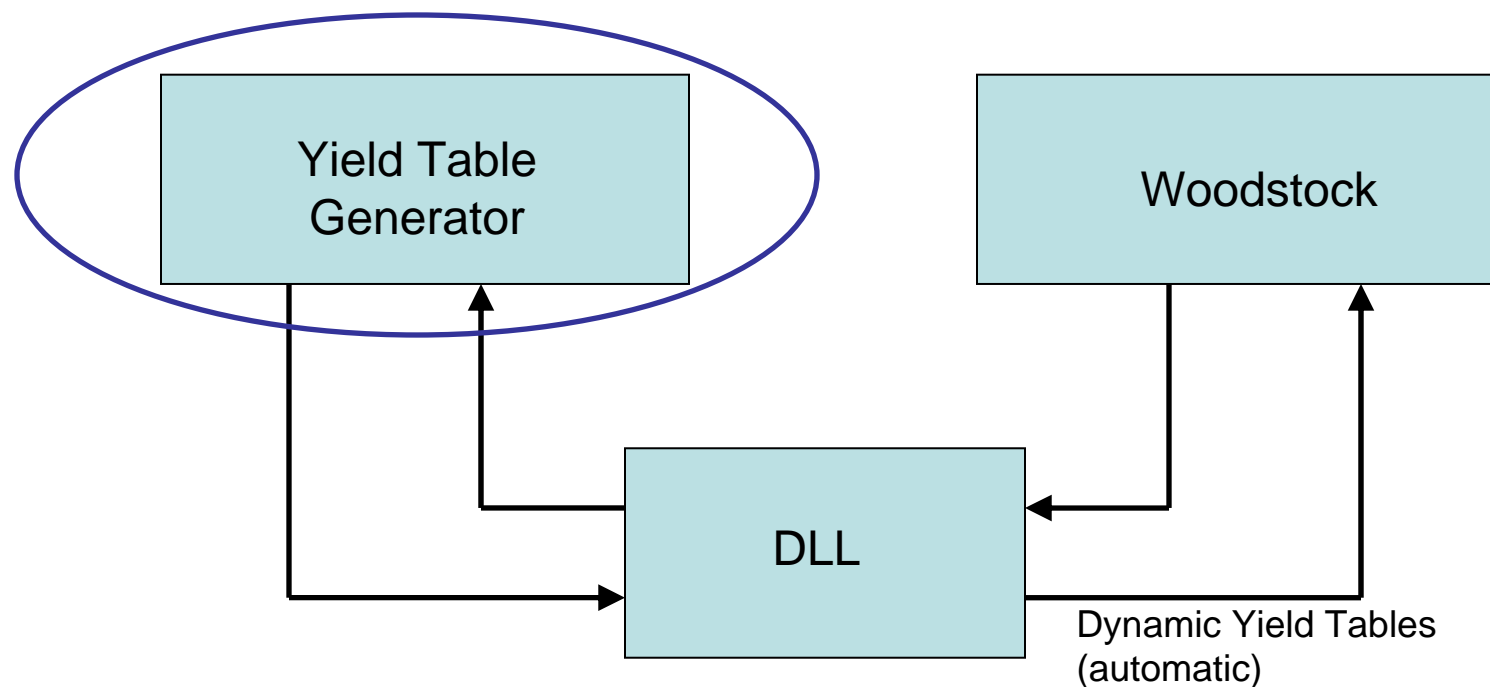


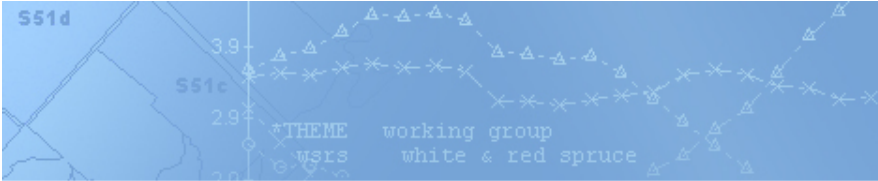
Diagram: An innovative, dynamic yield table creation process for forest estate modeling, M.L. McLarin, T.E. Osborn, L.J. Bennett, Forestry Tasmania, Hobart, TAS



## What do DLLs link to?

- Equations
- Growth and yield models
- Look-up tables?





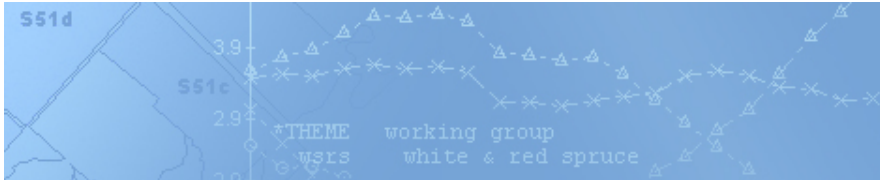
## Example?

What numbers must be sent?

- Depends on DLL!!

Example

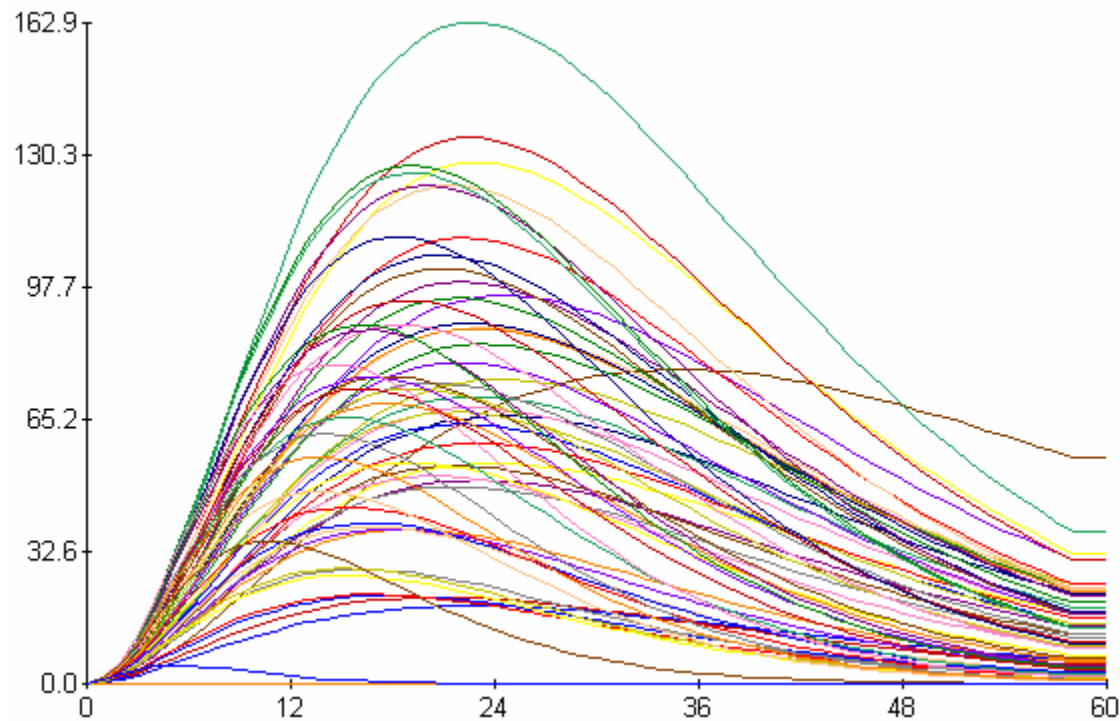
- 2 Parameter regression function
- Can use:
  - Yield masks OR
  - Thematic indexes

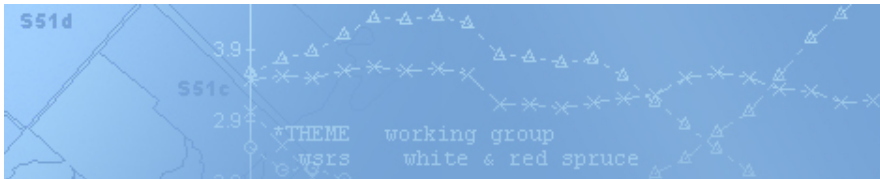


# Example?

\*YC ? ? ? ? ? ? ? ? ? ?

Softyld DLL(4) CalcYield.dll calcvol(a,b, AGE, 5)





## Example?

```
; Landscape
*THEME 1 YIELD STRATA

0401  _INDEX (a=0.026040341,b=2.3766438)
0411  _INDEX (a=0.026040341,b=2.3766438)
0421  _INDEX (a=0.020279287,b=2.1981665)
0431  _INDEX (a=0.027736791,b=2.3114654)
0441  _INDEX (a=0.021033433,b=2.3970681)
0402  _INDEX (a=0.028652387,b=2.312138)
0412  _INDEX (a=0.028652387,b=2.312138)
0422  _INDEX (a=0.020279287,b=2.1981665)
0432  _INDEX (a=0.027736791,b=2.3114654)
0442  _INDEX (a=0.021033433,b=2.3970681)
```

```

library CalcYield;
uses
    SysUtils,
    Classes;

{the function Up is the way that exponentiation must be done in Pascal}
{ Up(4,3) is the same as saying 4^3 i.e. 4 cubed }

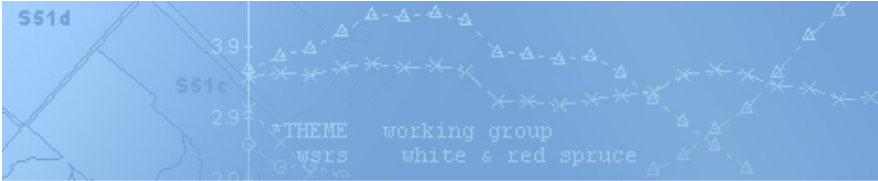
Function Up(x,y:single):single;
Begin
    Try
        Up:= exp(ln(x)*Y);
    Except
        Up:= 0;
    end;
end;

Procedure calcvol(A,B,Age,width:single;var Result:Single); Export;
begin
    Age := Age*width;
    if Age <= 0 then
        Result := 0
    else
        Result := A * up(Age,b)* exp(-a*age);
end;

exports
    calcvol;

begin
end.

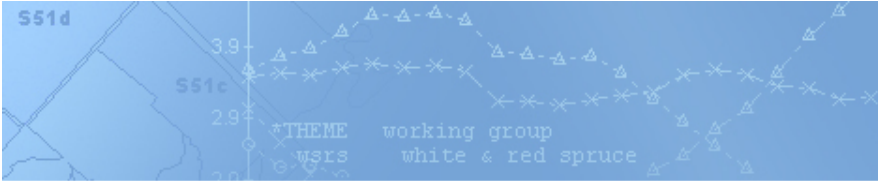
```



## *How hard they to develop?*

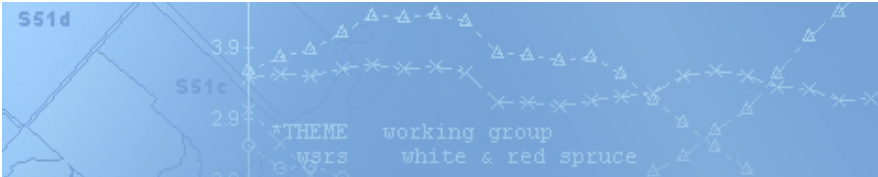
### **Depends!!**

- Complexity of growth and yield
- Complexity of how Woodstock must deal with it.
- Development time varies – 30 minutes to ????



## *Fixed versus Variant DLLs?*

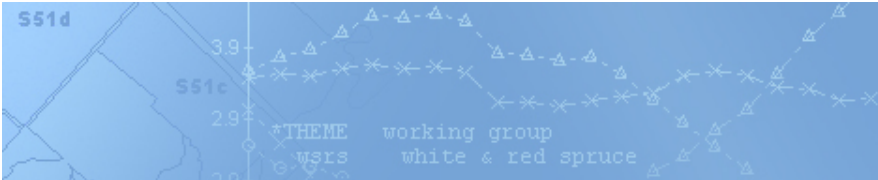
- History – started with Fixed and added Variant version in Woodstock version 3.2
- Fixed DLL
  - pass numbers and return “ONE” number
- Variant DLL
  - pass a mix of numbers and yield curves and it returns a yield curve
  - Variant – less calls, more flexible but more complicated to write DLL
- Previous example – could have been done either way
- Remsoft uses variants internally
- “MAI” example



## *Remsoft internal DLLs?*

- PNV
- \_SUM
- \_TIMES
- \_MULTIPLY
- \_DIVIDE
- \_REGENDELAY
- \_MAI
- \_PEAKMAI
- \_CAI
- \_YTP
- \_ENDPOINT

NOTE: How they are used depends on how they were programmed



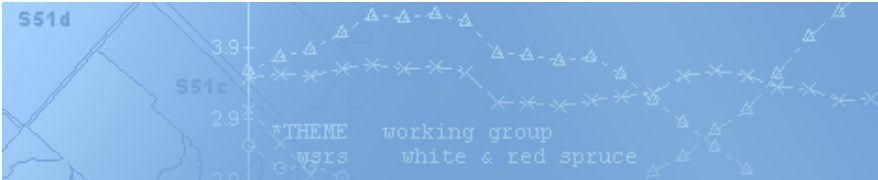
## *\_ENDPOINT example?*

- Required “end point” operability in Actions but wanted more than 2 conditions

**\*ACTION** Clearcut Y

**\*OPERABLE** Clearcut

??? PieceSize 12 TotVol 50



## \_ENDPOINT example?

- Required “end point” operability in Actions but wanted more than 2 conditions
- Initially...\_ENDPOINT programmed to translate starting and ending point as a 0 (not operable) 1 (operable) yield curve

\*Y ???

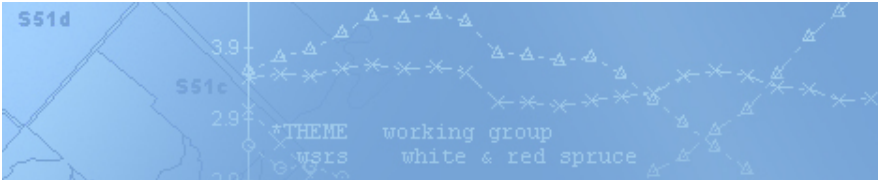
Optocut 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0

\***ACTION** Clearcut Y

\***OPERABLE** Clearcut

??? Optocut = 1 AND.....

SO...needs increased!!



## \_ENDPOINT example?

- Wanted to use curve elsewhere - e.g. trigger \_DEATH
- Modified DLL to output the following yield conditions:
  - “0” is used to trigger clearcut operability
  - # of years “before” operable are recorded as -1, -2,...
  - # of years “after” operable are recorded as +1, +2,...

\*Y ???

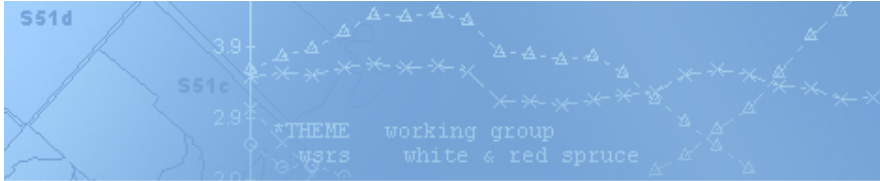
Optocut 1 -5 -4 -3 -2 -1 0 0 0 0 0 0 0 +1 +2 +3....

- Can now reference death as one year after operable...

\***ACTION** \_DEATH Y

\***OPERABLE** \_DEATH

??? Optocut = +1



## Summary

- Yields are an important but challenging part of the model
- DLLs – will they work? Depends on growth and yield.
- But once they are done, they are:
  - Powerful
  - Flexible
  - Easy to use