

TreeAge Pro Object Interface

TreeAge Software, Inc.



TreeAge Pro Interfaces

- Graphical User Interface (GUI)
 - Normal human interface for desktop software as viewed on your computer.
- Object Interface
 - Full-featured scripting interface provides access from any program/script/macro.
 - Provides access to many of TreeAge Pro's core functions.
 - Requires the Excel Module.
 - TreeAge Pro Excel or TreeAge Pro Suite



TreeAge Pro Object Interface

- TreeAge Pro 2007 introduced the TreeAgeProLib library which provides access to TreeAge Pro objects.
 - Application, trees, nodes, etc.
- TreeAgeProLib implements a standard COM interface to create TreeAge Pro objects.
- Any program/script/macro can create TreeAge Pro objects through the COM interface.



Why Use the Object Interface?

- Automate tasks that are common, repetitive and/or time-consuming.
- Set a tree's data values from outside TreeAge Pro.
- Automatically run a set of analyses.
- Automatically export analysis results.
- Integrate TreeAge Pro functions into another system or application.



What is an Object?

- An object is a building block for programs usually representing a specific concept (i.e., a tree or node).
- An object contains properties (data) and methods (functions).
- Objects can be instantiated (created) in your scripts to perform functions and set/retrieve data.



Object Interface Documentation

- Online documentation:
<http://server.treeage.com/ObjDocs/TP/TOC/ref.php3>
 - Includes descriptions of all objects and their properties and methods.
 - Code samples.
- Object browser:
 - Included with Microsoft VB editor.
 - Shows object properties and methods.
- For more details and examples:
<http://www.treeage.com/support/TreeAgeObjects.html>



Primary TreeAge Pro Objects

- ApplicationObj
- TreeObj
- NodeObj



ApplicationObj

- Provides access to the TreeAge Pro application.
 - Similar to the GUI's application window.
- Open and save TreeAge Pro documents.
 - Trees
 - Graphs
 - Analysis objects
 - Etc.



TreeObj

- Provides access to data and functions of a TreeAge Pro tree, including...
 - Get node objects.
 - Get/set tree properties and preferences.
 - Create/update/read variables.
 - Create/update/read tables.
 - Create/update distributions.
 - Create/update bilinks (dynamic links to Excel).
 - Run analyses and get results.



NodeObj

- Read/update variable definitions.
- Navigate to other nodes.
- Get expected values.
 - Including cost/effectiveness.
- Evaluate variable definitions/formulas.



Analysis Objects

- CEOutput
 - Output from Cost-Effectiveness Analysis
- MarkovParams/MarkovOutput
 - Input to/Output from Markov Analysis
- MonteParams/MonteOutput
 - Input to/Output from Monte Carlo Simulation



Other Objects

- GraphObj
 - Graph document object
- TextReportObj
 - Text report document object
- TableObj
 - Table object (data, properties)
- Utility objects
 - CEStrategy, GlobalObj, etc.



Example:

Set Variables & Run Analyses

1. Loop through a set of input values.
2. Place input values in variable definitions within the model.
3. Run analyses on model...
 1. Calculate EV for payoff 1 and payoff 2 at root node and at each strategy.
 2. Run Monte Carlo simulation on model.
 3. Run Markov Cohort Analysis on model.
4. Place results of analyses in Excel.



Example:

Set Variables & Run Analyses

- Inputs&Analysis.xlsm worksheet contains a module with Visual Basic code
- Reference to TreeAgeProLib (TreeAge Pro 2007.1.0 Type Library)
- Global variables in module can be accessed by all subroutines/functions within the module



Example:

Set Variables & Run Analyses

- `start()`
 - Starting point for execution
- `setGlobalValues()`
 - Set constants used within the scripts
(Excel rows/columns, variable names, etc.)



Example:

Set Variables & Run Analyses

- `loopThroughData()`
 - Main processing loop for rows of data in Excel
- `assignVariable()`
 - Set variable definition at root node
- `openTree`
 - Open `ApplicationObj`
 - Open `TreeObj` for the specific model



Example:

Set Variables & Run Analyses

- `calcEV()`
 - Calculate EV for root and strategies
 - Calculate EVs above for 2 payoff sets
- `getNodeEV()`
 - Get a single EV value for the selected node
 - Used by `calcEV()` several times



Example:

Set Variables & Run Analyses

- `runSimulation()`
 - Run a Monte Carlo simulation
 - Gather mean and std dev from simulation
- `setupMonteParams()`
 - Setup options for simulation above
- `outputTextReport()`
 - Loop through Text Report from simulation and place data in Excel



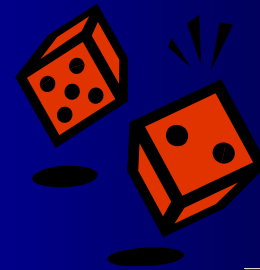
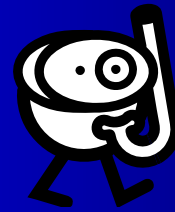
Example:

Set Variables & Run Analyses

- `runMarkovCohort()`
 - Run a Markov Cohort Analysis

Summary

- Questions?



TreeAge Pro Interactive

- The TreeAge Pro Object Interface is similar to the interface for TreeAge Pro Interactive (TPI).
- TPI has no GUI and only acts on TreeAge Pro objects via scripts.
- Used by customers to write scripts to use TreeAge Pro functions without requiring a TreeAge Pro license for distribution.
 - License fees apply to a protected package file with unlimited distribution.
- Syntax is slightly different from the TP object interface.
- TPI Documentation:
<http://server.treeage.com/ObjDocs/TPI/TOC/ref.php3>

Additional Notes

- ByRef arguments return values
 - TreeObj.getVariableProperties
 - <http://server.treeage.com/ObjDocs/TP/TreeObj/getVariableProperties.php3>
- Monte Carlo Simulation
 - Need to wait for simulation to complete before processing results.
 - MonteOutput.TimeElapsed
 - <http://server.treeage.com/ObjDocs/TP/MonteOutput/TimeElapsed.php3>
- Need to add reference to the object library to use the TreeAge Pro objects.
 - In VB Editor: Tools > References.
 - Check box for “TreeAge Pro 2007.1.0 Type Library”.
- Occasionally, a script coding issue will cause TreeAge Pro to hang.
 - Close TreeAge Pro via the Task Manager.