
# PERFORMANCE ENHANCEMENTS IN TreeAge Pro 2014 R1.0

15th January 2014

Al Chrosny
Director, Software Engineering
TreeAge Software, Inc.
achrosny@treeage.com

Andrew Munzer
Director, Training and Customer Support
TreeAge Software, Inc.
amunzer@treeage.com

## INTRODUCTION

TreeAge Pro 2014 R1.0 introduced performance enhancements that enable users to open and manipulate large models more efficiently. Since 2011, TreeAge Software uses the Eclipse software platform and uses Java programming language for all development. The combination of Java and Eclipse allows TreeAge Pro to run on multiple Operating Systems with virtually the same User Interface look and feel. Prior to 2011, TreeAge Pro was available on Windows only. Since 2011 the software is able to run on Windows, Mac OS and different flavors of Linux.

The Eclipse based TreeAge Pro provided faster analyses run times vs. the old TreeAge Pro 2009 Windows product. More details on analyses performance between 2009 and 2013 version will be provided later. The Windows based User Interface was less flexible, but was very fast allowing users to display and manipulate large models. The Eclipse User interface is more sophisticated but not as efficient in manipulating large models. With version 2014 the speed of opening and editing large models was increased by over 200%. Thanks to this significant display efficiency enhancement TreeAge Pro 2014 is now able to expand and display “Cloned” sub-trees.

**BENCHMARKS**

TreeAge Pro users continue to create models that stress the limits of what TreeAge Pro is able to effectively handle. We work very closely with our customers and provide advice on how to structure their models and the size of analyses that they can expect to run. However, the ability to deal with large models is a function of many parameters:

1. Computer CPU architecture (single vs. multicore).
2. Computer CPU clock speed.
3. Computer RAM size.
4. Computer Operating Architecture (32-bit vs. 64-bit).
5. Java Virtual Machine version.
6. Amount of Java Heap Space allocated to TreeAge Pro (default size 768 MB).

Additionally Java uses two processes that can significantly affect the performance of running the same operation:

1. Just in Time (JIT) compilation – Java virtual machine detects areas of the software which are used more often and converts them to a more efficient code.
2. Garbage Collection that periodically reclaims unused memory from the heap.

Combination of the JIT-Compilation and Garbage collection can significantly vary the performance of the same operation. For example, sometimes you may experience a longer time before a new dialog is presented by TreeAge Pro. The next time the same dialog is accessed it will come up faster. This is a result of JIT-Compiler performing the optimization on the first access to the dialog.

Subsequent versions of Java Virtual Machine try to introduce better and better technologies for dealing with Garbage Collection, Compilation, Security, etc., which result in varying performance between different versions of Java Virtual Machine VM. To some extent benchmarking of Java based applications can be a moving target and your run times may vary!

We have selected four medium to very large size models from our customers. We have also used a simple 1000 node model that we used for copy and paste benchmarking.

The following is the description of the models:

| Model | Size MB | Number of Nodes (Clone copies are not included) | Notes |
|---|---|---|---|
| Nodes 1000.trex | 1.2 | 1000 | Used for Copy Paste tests. |
| Model A | 10.0 | 6,355 | Medium size customer model |
| Model B | 15.3 | 10,353 | Large size customer model |
| Model C | 24.0 | 14,108 | Large size customer model |
| Model D | 97.6 | 157,948 | Very Large customer model |

Some very large models would not be able to be opened with standard size of the heap memory allocation (default setting for TreeAge Pro 2013 and 2014 is 768 MB). In case of Model D the heap size was increased to 6000 MB.

Computer used for the tests had the following specifications:

| | |
|---|---|
| Processor | Intel i7-3615QM @ 2.29 GHz |
| Memory RAM | 8 GB |
| Operating System | Windows 7 Professional N 64-bit |
| Java Virtual Machine version | 1.6.0_43 |
| Heap size allocation see test notes | default 768 MB (Adjusted to 6000MB for Model D) |

Each model was opened 3 times and average time is used for comparison between TreeAge Pro 2014 and TreeAge Pro 2013. For large and very large models, TreeAge Pro needed to be restarted before attempting to open the large model again. Closing the model does not always reclaim heap memory and this may lead to unusually long opening times. To avoid throwing this additional variability TreeAge Pro was restarted prior to opening the large models (as indicated in the notes).

| Model | Performance Increase (ratio 2013 / 2014) | 2013 R2.1 Average [seconds] | 2014 R1.0 Average [seconds] | Number of Nodes | Notes |
|---|---|---|---|---|---|
| Nodes 1000.trex | **1.4** | 4.1 | 2.8 | 1,000 | |
| Model A | **1.7** | 16.9 | 9.9 | 6,355 | |
| Model B | **2.0** | 38.9 | 19.6 | 10,353 | See **Note 1** |
| Model C | **3.7** | 66.7 | 18.0 | 14,108 | Restarting TreeAge |
| Model D | **5.2** | 151.8 | 28.9 | 157,948 | Restarting TreeAge. Heap memory 6000 MB |

**Note 1** - *In tests of Model B TreeAge Pro was not restarted prior to opening the model. As a result there has been more variability between different tests due to the fact that Garbage collection sometimes would happen during the model opening. That is why Model C average is somewhat smaller than model B due to the fact that Garbage collection had greater effect on model B tests*.

Following table shows the individual test results for model B in TreeAge Pro 2013 and 2014. Times are indicated in seconds:

| TreeAge Pro Version | Test 1 [s] | Test 2 [s] | Test 3 [s] | Average [seconds] | Performance Increase |
|---|---|---|---|---|---|
| **2014** | 16.5 | 26.6* | 15.8 | **19.6** | **2.0** |
| **2013** | 38.2 | 40.1* | 38.4 | **38.9** | **-** |

** Garbage collection was triggered during the second opening of model B in both TreeAge Pro 2013 and 2014.*

It was necessary to adjust memory heap size to 6000 MB to open model D in both TreeAge Pro 2013 and TreeAge Pro 2014. However, in the 2013 version the ability to open and edit was not practical since it took several minutes to manipulate the model. This highlights one of the important aspects of modeling, the user desire to create very detailed and complex models has to be balanced against available tools and computing resources. TreeAge Pro offers an intuitive graphical modeling platform,

but it is also easy to create models that stretch the limits of the performance of the tool and the computer.

**NOTE ON ANALYSES PERFORMANCE**

The performance of actual analyses such as PSA, microsimulation, etc. was significantly improved when TreeAge Pro was re-implemented in Java and Eclipse. The difference in performance is more pronounced for larger more complicated models.

Here are two an examples comparing of simulations performed in TreeAge Pro 2013 and 2009 on the same Windows 64 bit computer (6 cores 4GB memory and 3.4GHz clock speed).

**Model:** **Example13-MicrosimulationPSA.trex (From Healthcare training examples).**

**Analysis:** **Monte Carlo 2 dimensional (Sampling + Trials)**

2nd-order parameter samples: 1000

1st-order parameter trials: 1000

(Total of 1,000,000 iterations)

Multithreading enabled.

The following is a comparison between TreeAge Pro 2013 and TreeAge Pro 2009.

| TreeAge Pro Version | Run 1 [s] | Run 2 [s] | Average [seconds] | Performance Increase |
|---|---|---|---|---|
| **2013** | 27.58 | 27.92 | **27.75** | **2.4** |
| **2009** | 65.94 | 69.96 | **67.95** | **-** |

The same computer was used with an even more computationally demanding model that compares Markov and Discrete Event Simulation modeling techniques within single decision tree.

**Model:** **MarkovDES ComparisonTwoStrategiesHalfCycleStudy3.trex**

**Analysis:** **Monte Carlo 2 dimensional (Sampling + Trials)**

2nd-order parameter samples: 200

1st-order parameter trials: 20,000

(Total of 4,000,000 iterations)

Multithreading enabled.

The following is a comparison between TreeAge Pro 2013 and TreeAge Pro 2009.

| TreeAge Pro Version | Run 1 [s] | Performance Increase |
|---|---|---|
| **2013** | **293.0** | **48.3** |
| **2009** | **14160.0** | **-** |

## CONCLUSION

TreeAge Pro 2014 R1.0 builds on prior analysis performance enhancements achieved by TreeAge Pro 2011 – 2013. TreeAge Pro 2014 R1.0 also addresses some issues users have found creating and editing larger models in TreeAge Pro 2011-2013. This capability comes with the caveat that just because a model is larger does not mean that it simulates reality better or that it uses the available computing resources optimally.