

An “event” object is a special kind of node whose main purpose is to act as a link between a DynRisk model and other applications. During a simulation an event sends messages to its target application, and receives messages back again.

You will soon realize that using events is an extremely powerful way of extending the modeling capabilities of DynRisk. However, this power comes with a price.

Firstly, events introduce a new level of complexity to modeling. Instead of just working within DynRisk, you now have to deal with a second application. Moreover, the complete model is no longer necessarily contained within a single document. This introduces many new possibilities for errors in your model.

Secondly, the AppleEvent communication protocol is relatively slow. This means that even on a fast CPU, your simulations will need much more time to finish.

Finally, having to run more applications simultaneously, means that you may need more RAM to make things work.

That said, we are still sure that you will find events very useful as a supplement to the internal model facilities in DynRisk.

The most obvious reason for using events, is of course that you can do things that otherwise would have been impossible or at least difficult to accomplish. Suddenly you have all the power of e.g., Microsoft Excel available to your risk analysis. At the same time, you have access to scripting languages such as HyperTalk and AppleScript.

Sometimes you can save a lot of work by using events as well. Assume e.g., that you have a large spreadsheet model which you want to do a risk analysis on. That is, you want to see how uncertainty about some input variables affects the results.

Without events, the only way to do this, would be to rebuild the entire spreadsheet model in DynRisk and run a simulation on the resulting model. If you later made changes to the spreadsheet model, you would need to update the DynRisk model as well.

By using events, however, you could just build a small model for the input variables and link these to the corresponding cells in the spreadsheet. To get results back, you would also create some links to the key output cells. During a simulation, DynRisk will feed numbers generated in the input model, over to the input cells, recalculate the spreadsheet, log the output

values back, and store them on file. Moreover, unless you make changes either to the input cells or output cells, the risk model will stay operative as you update your spreadsheet.