



# Viskit: Visual Simkit Editor

Don Brutzman and Terry Norbraten

Modeling Virtual Environments Simulation (MOVES) Institute  
Naval Postgraduate School (NPS)

9 June 2025

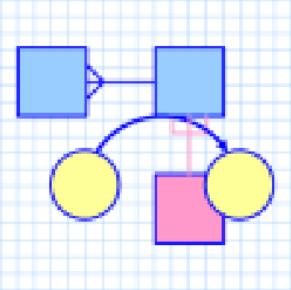
# Overview

- Viskit is a modeling application for defining Discrete Event Simulation (DES) models that autogenerate Java source using the Simkit library.
- The Viskit application allows you to build a collection of Simkit event graphs that describe the behavior of various entities in a simulation environment. Each model file is recorded in XML data file.
- Capabilities include **Event Graphs**, **Assemblies**, **Simulation Run** tools, and an **Analyst Report** generator suitable for thesis and research efforts.
- Current release includes a thorough refactoring of the underlying Viskit source code, facilitating ongoing maintenance and development.
- Open-source efforts are ongoing.

# About

About Viskit

X



Viskit Event Graph and Assembly Editor  
version 2.0.0, last modified: 2025-06-09

Copyright © 1995-2025 under BSD-style open-source license

**Developers:**

Don Brutzman

Terry Norbraten

with

Mike Bailey

Arnold Buss

Rick Goldberg

Don McGregor

Patrick Sullivan

Visit the Viskit home page at

<https://github.com/open-dis/viskit>

Visit the Simkit home page at

<https://github.com/ahbuss/simkit>

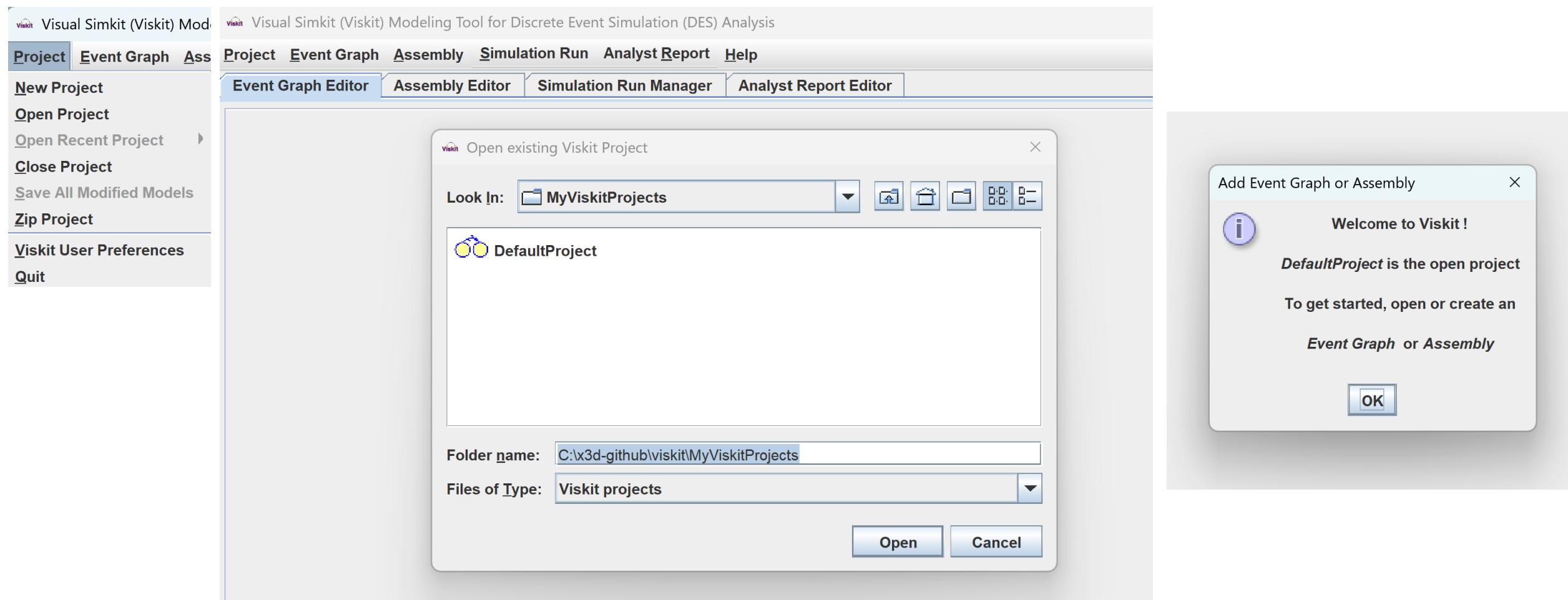
---

Simkit Version: 1.5.6

Java version: 24.0.1

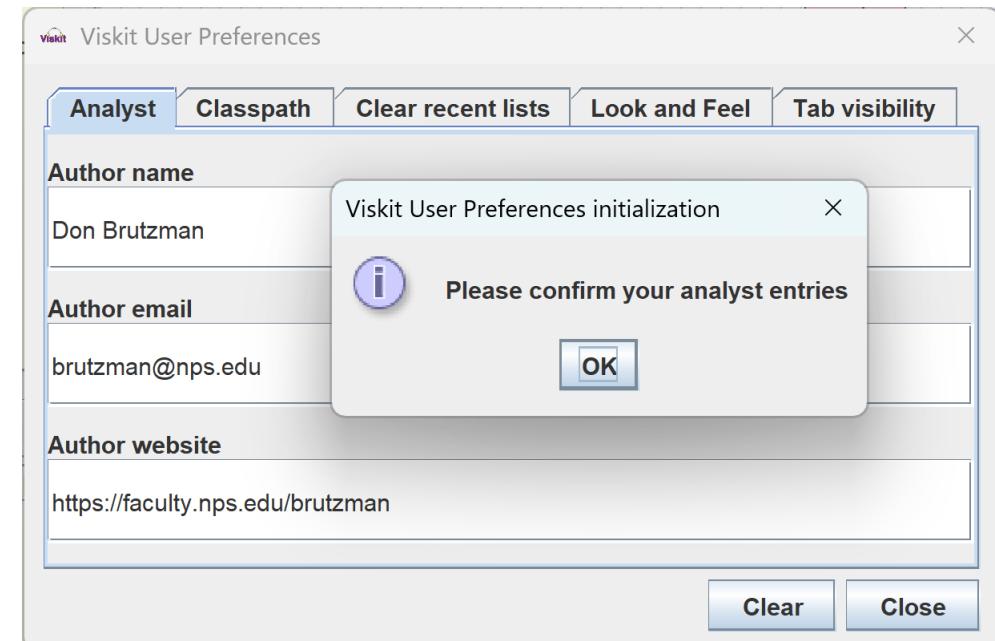
OK

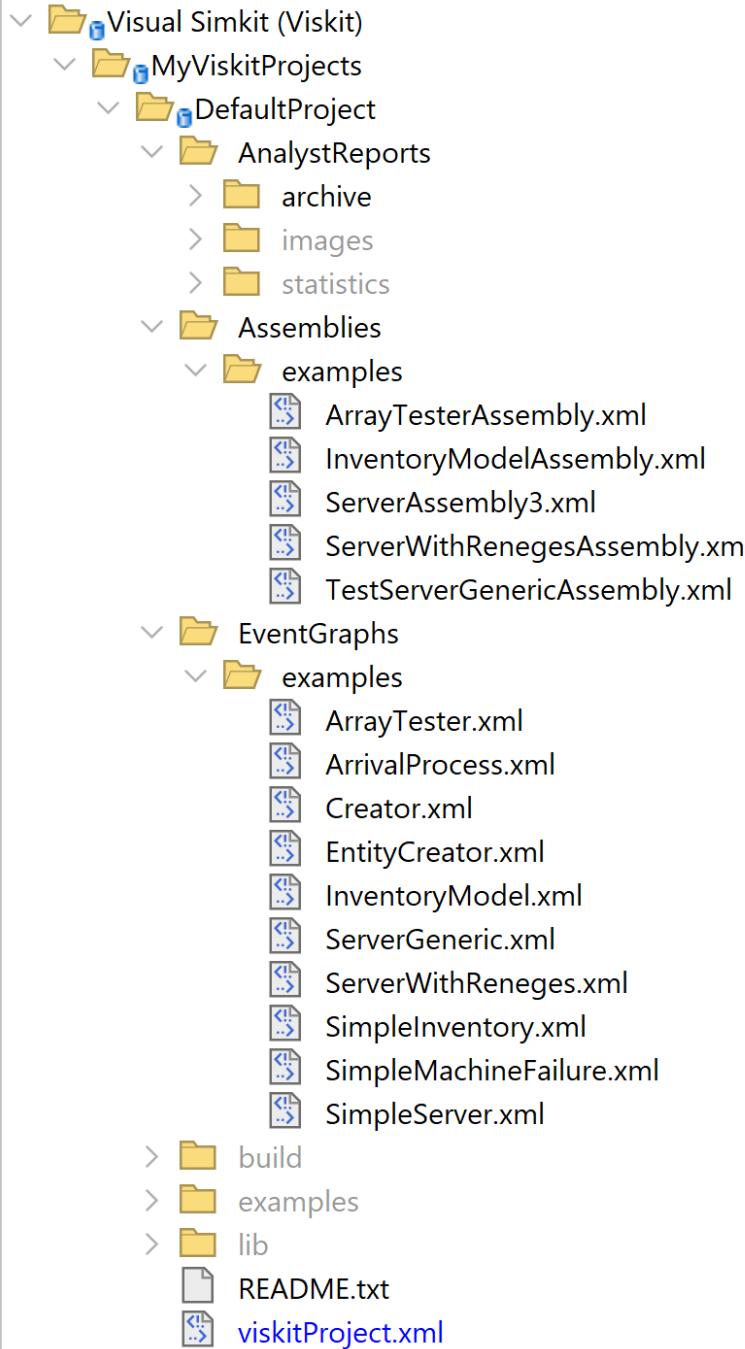
# Viskit is organized according to Viskit Projects



# Viskit User Preferences

Author information is applied when creating an Analyst Report



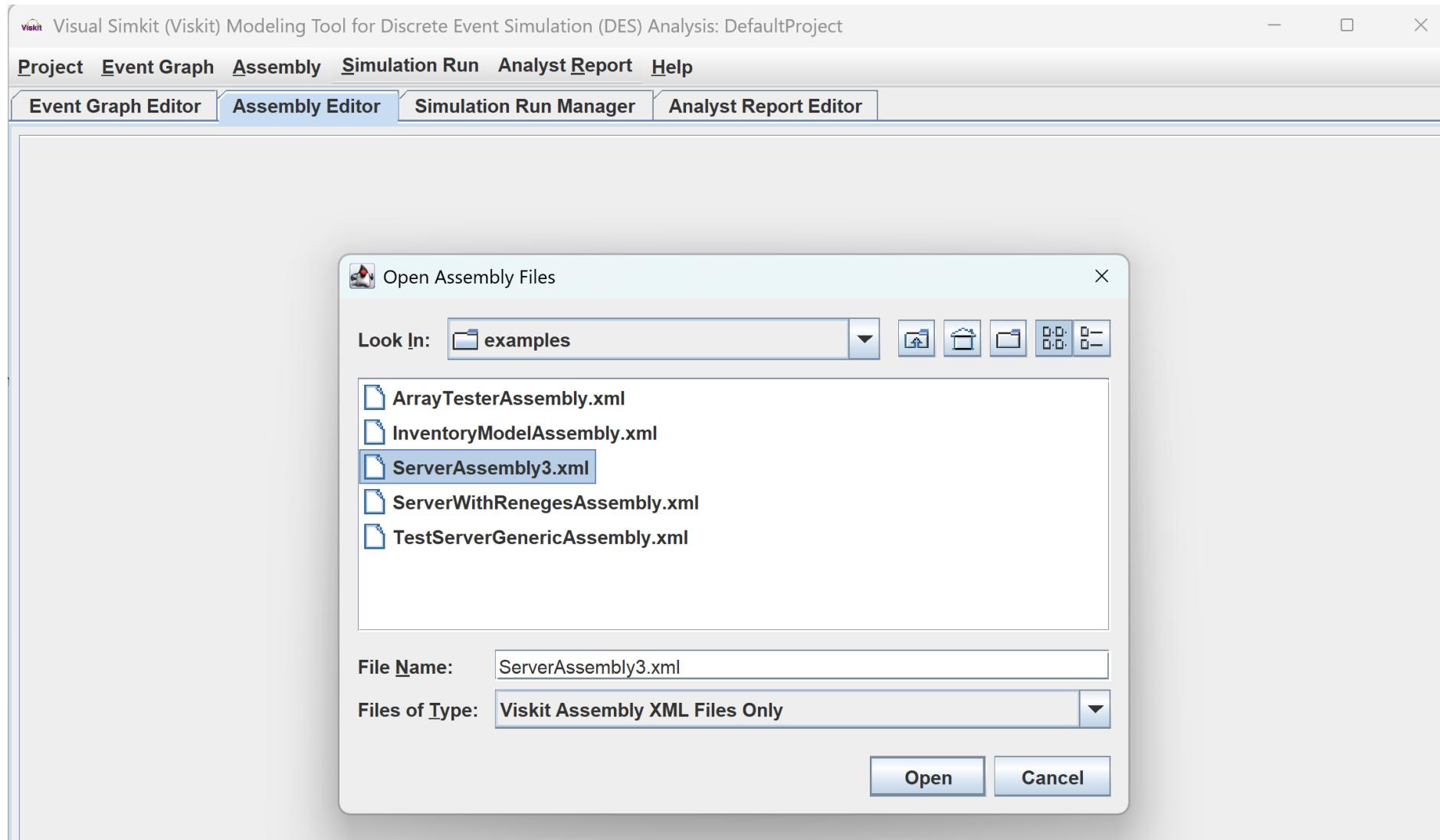


# Viskit project files contain multiple assets

The screenshot shows an IDE interface with the tab "viskitProject.xml" selected. The code editor displays the XML structure of the project file:

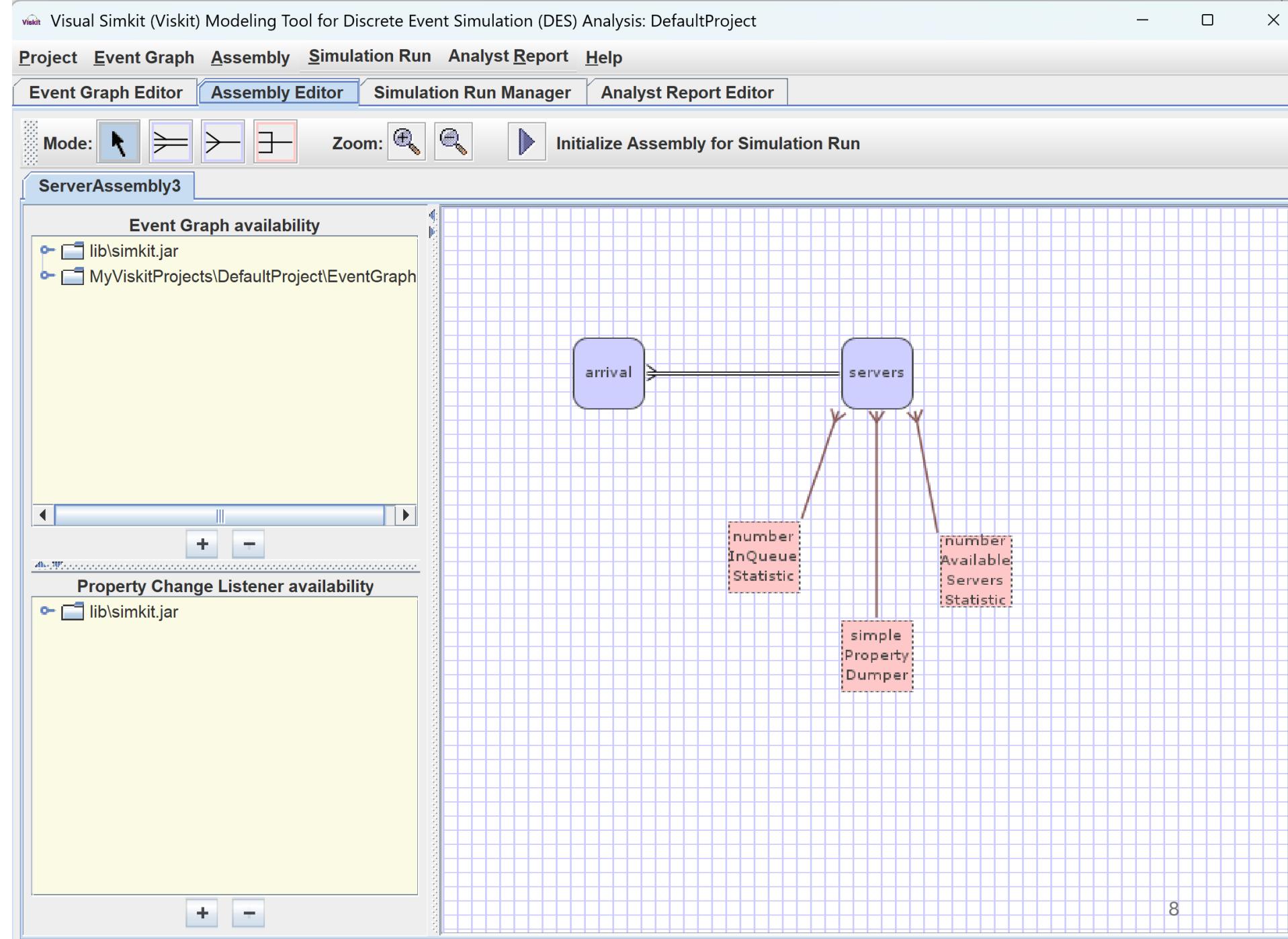
```
<?xml version="1.0" encoding="UTF-8"?><ViskitProject>
    <!-- viskitProject.xml -->
    <AnalystReports name="AnalystReports"/>
    <AssembliesDirectory name="Assemblies"/>
    <EventGraphsDirectory name="EventGraphs"/>
    <BuildDirectory name="build">
        <ClassesDirectory name="classes"/>
        <SourceDirectory name="src"/>
    </BuildDirectory>
    <DistDirectory name="dist"/>
    <LibDirectory name="lib"/>
    <Project name="DefaultProject">
        <Cached>
    </Project>
</ViskitProject>
```

# Next step: open an assembly



# For example

Server  
Assembly3  
.xml



Project Event Graph Assembly Simulation Run Analyst Report Help

Event Graph Editor

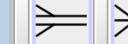
Assembly Editor

Simulation Run Manager

Analyst Report Editor

Metadata:

Mode:



Zoom:

Initialize Assembly for Simulation Run

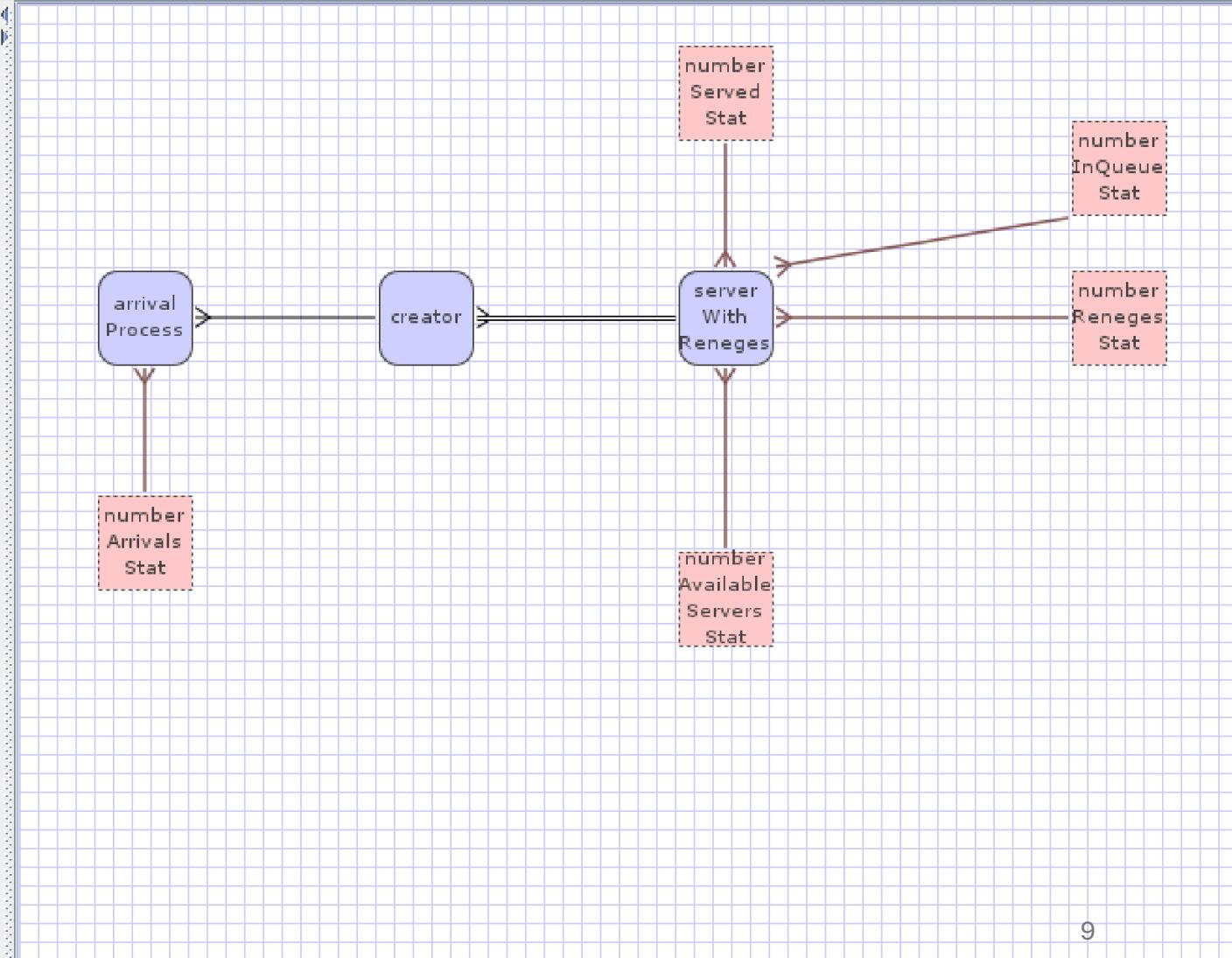
ServerWithRenegesAssembly

Event Graph availability

- lib\simkit.jar
- MyViskitProjects\DefaultProject\E

Property Change Listener availability

- lib\simkit.jar



# Editing

You can edit Event Graphs, Assemblies, Simulation Run results, and Analyst Report outputs

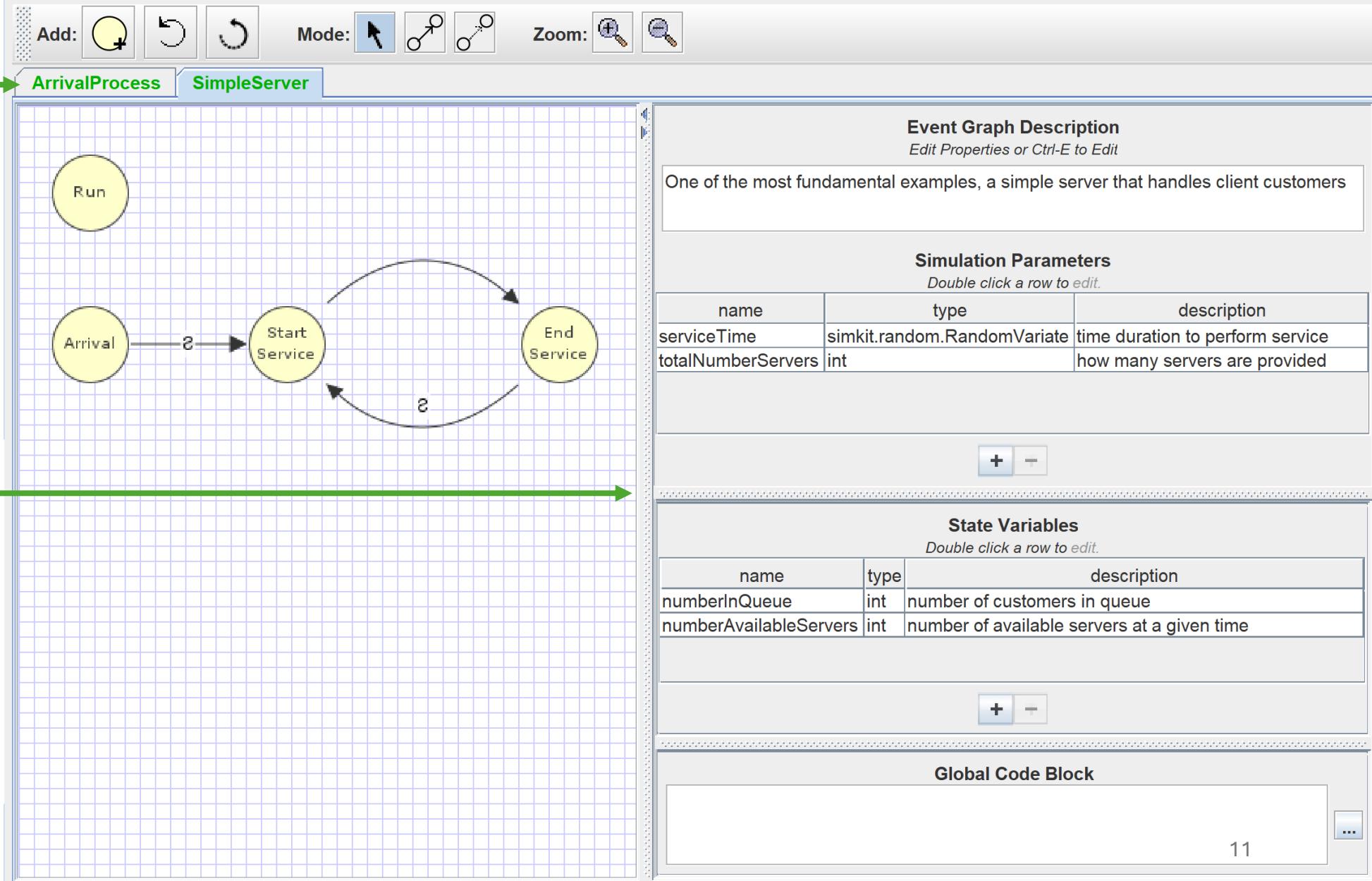
[Project](#) [Event Graph](#) [Assembly](#) [Simulation Run](#) [Analyst Report](#) [Help](#)[Event Graph Editor](#)[Assembly Editor](#)[Simulation Run Manager](#)[Analyst Report Editor](#)

ArrivalProcess

SimpleServer

Check that  
corresponding  
Event Graphs  
were opened

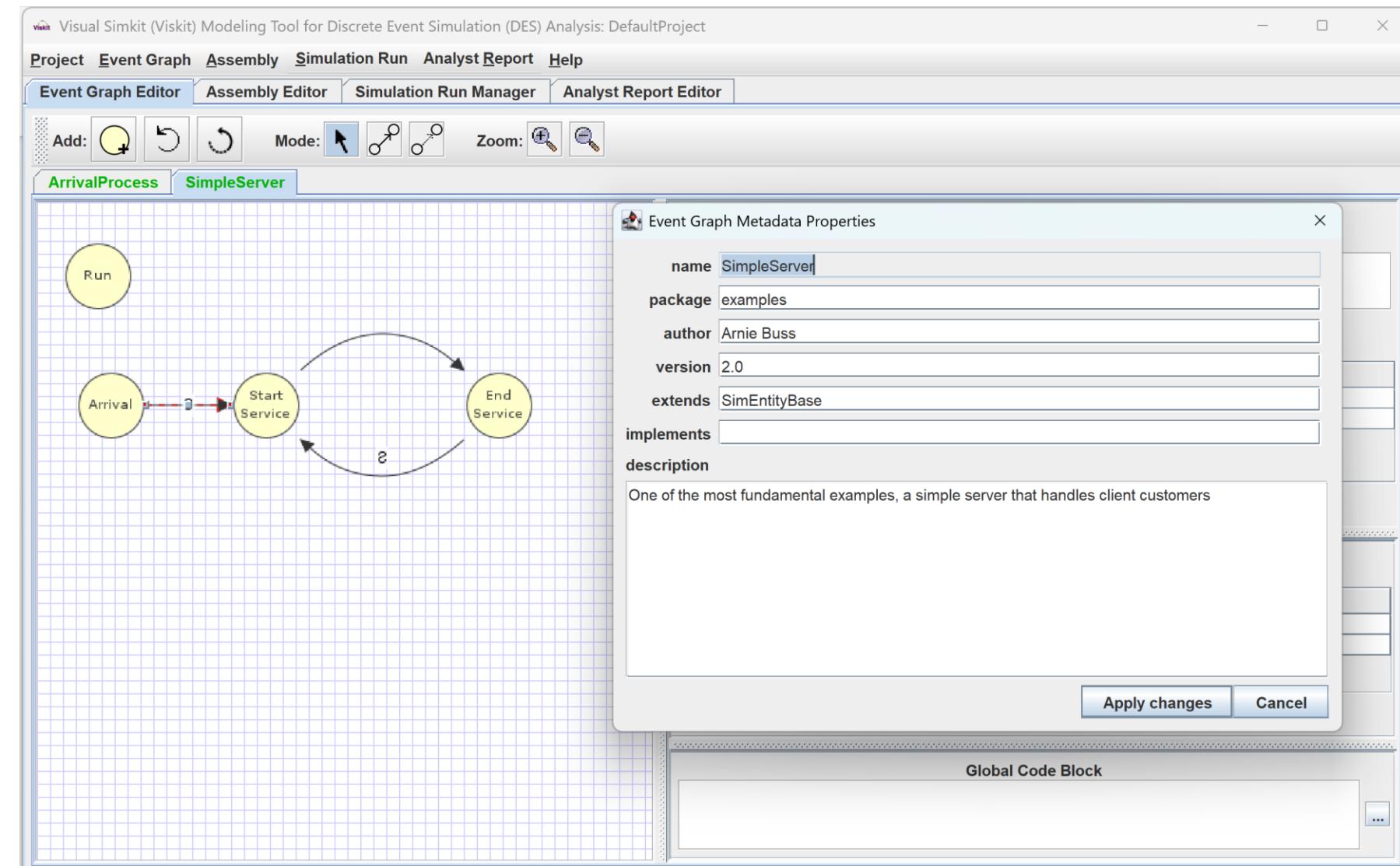
Check out  
Event Graphs  
Simulation  
Parameters and  
State Variables



- [Edit selected Event Graph...](#)
- [Edit selected Event Graph Metadata Properties... Ctrl-E](#)
- [New Event Graph Ctrl-N](#)
- [Open Event Graph Ctrl-O](#)
- [Open Recent Event Graph](#)
- [Close Event Graph Ctrl-W](#)
- [Close All Event Graphs](#)
- [Save Event Graph Ctrl-S](#)
- [Save Event Graph as... Ctrl-A](#)
- [Image Save for Event Graph Diagram Ctrl-I](#)
- [Java Source Generation for saved Event Graph Ctrl-J](#)
- [XML View of Saved Event Graph Ctrl-X](#)

# Editing an Event Graph

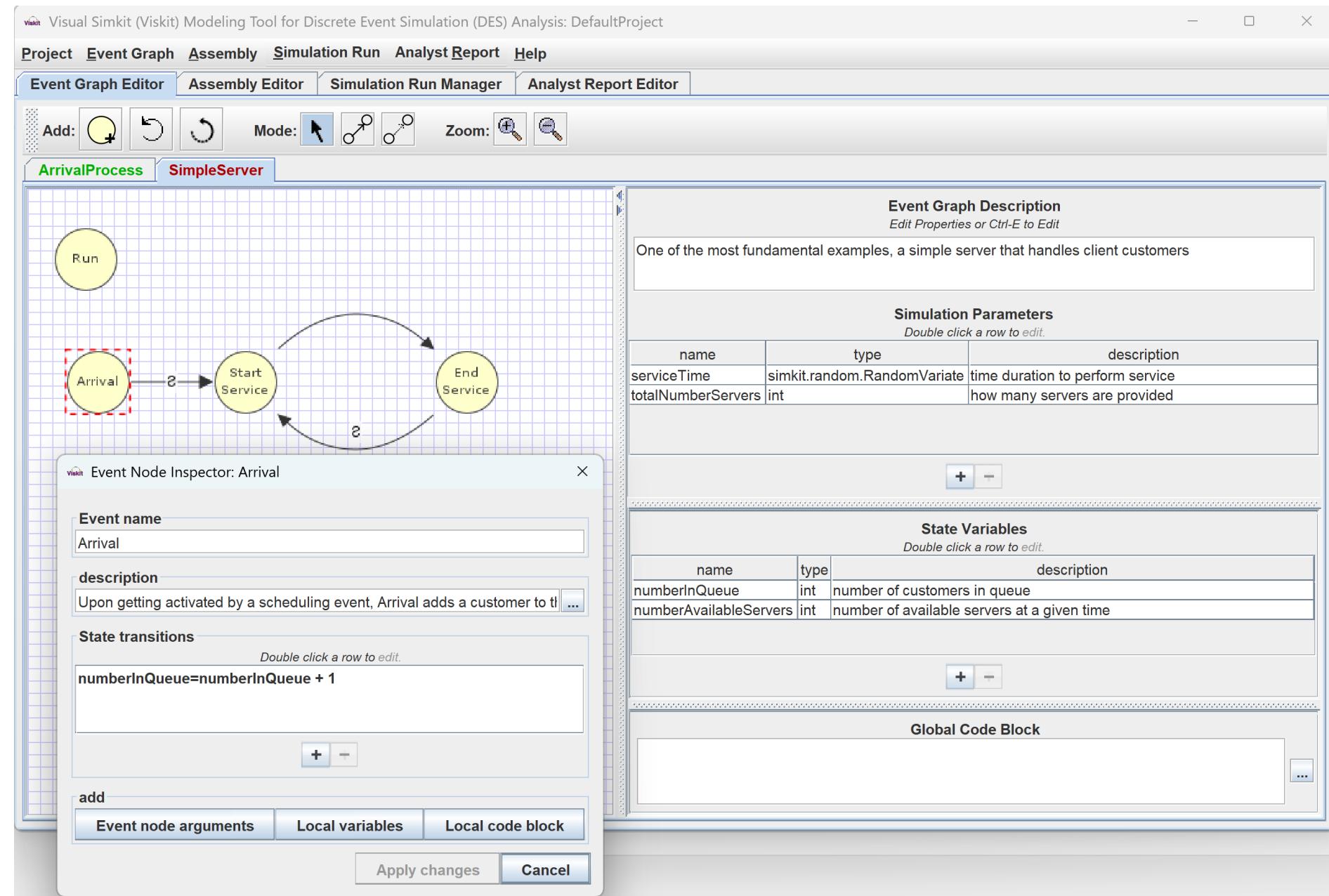
## Inspect Metadata



# Editing an Event Graph

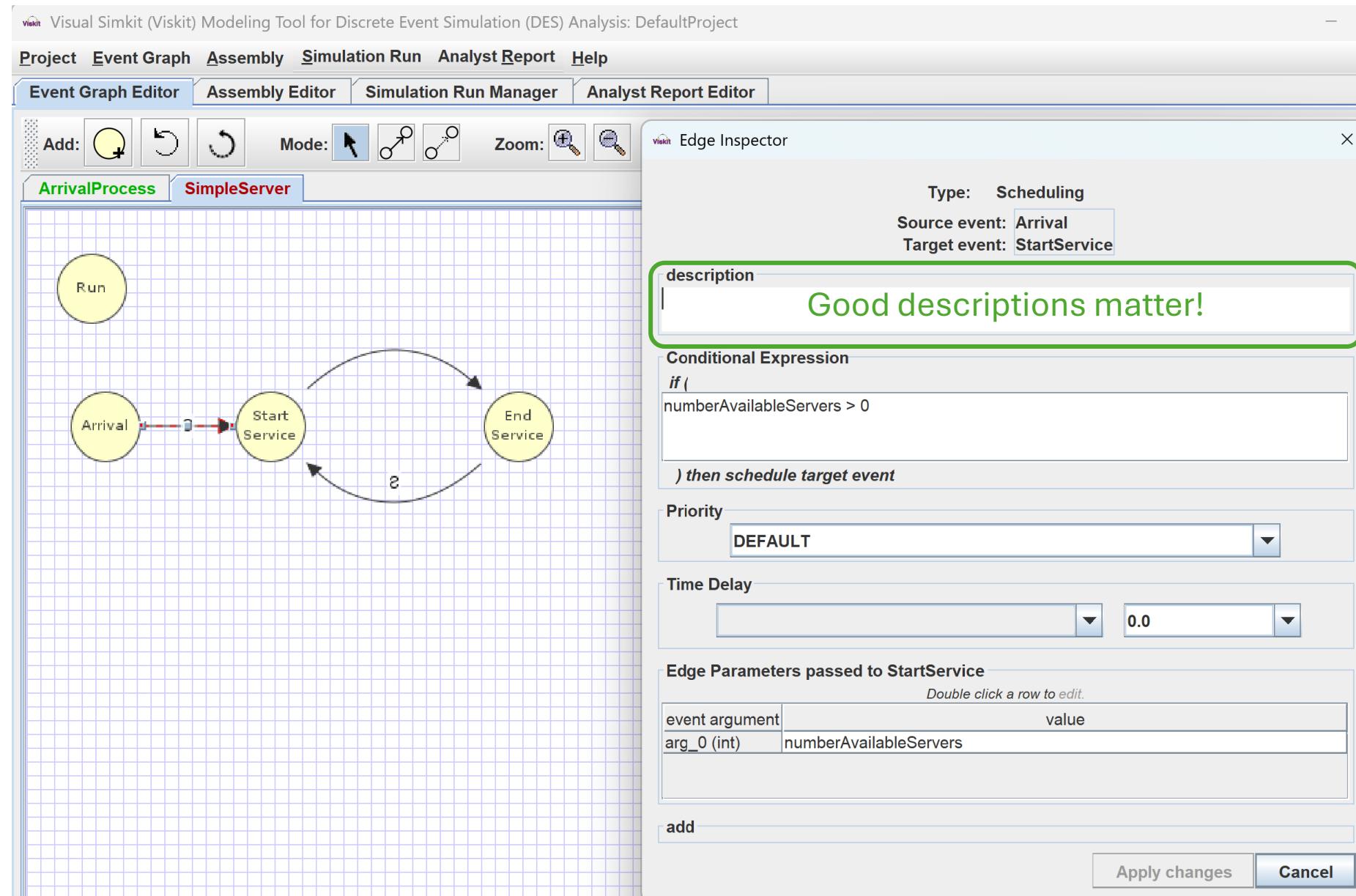
## Simple Server.xml

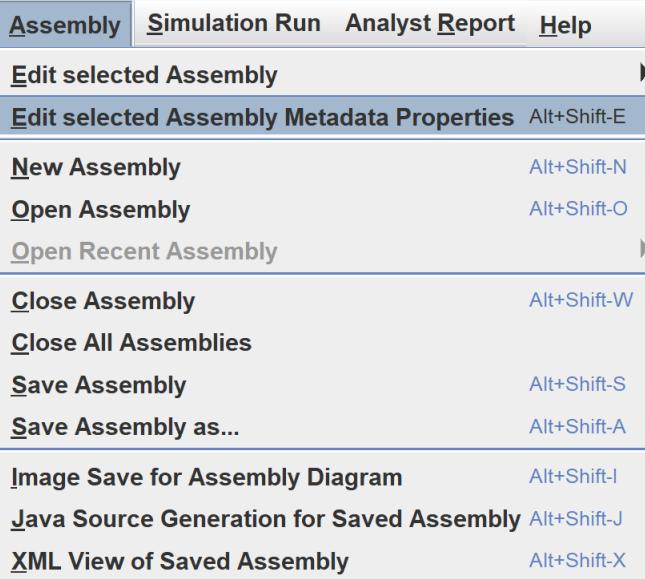
## Inspect State Nodes



# Editing an Event Graph

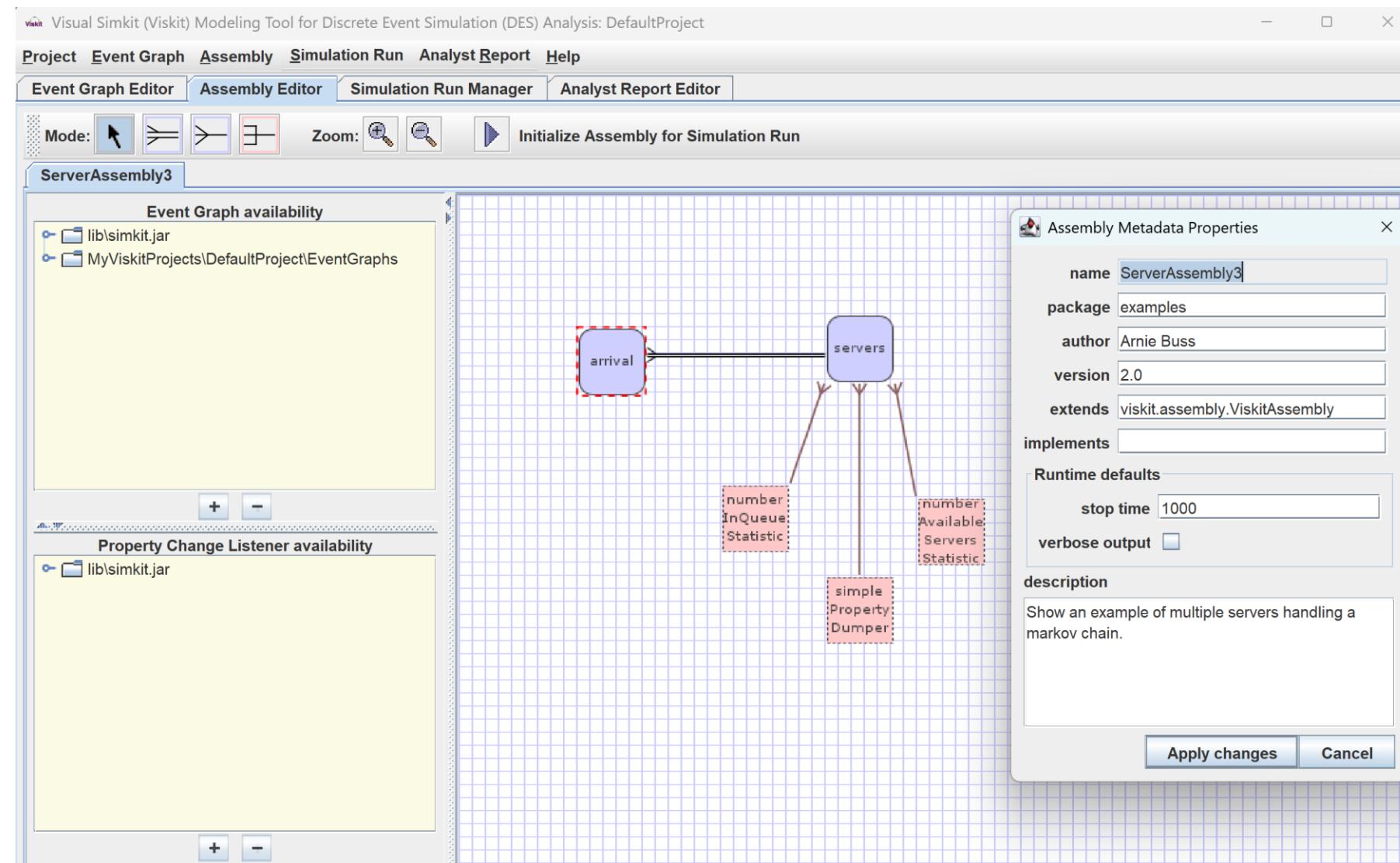
Inspect  
Scheduling  
(Cancelling)  
Edge  
Transitions





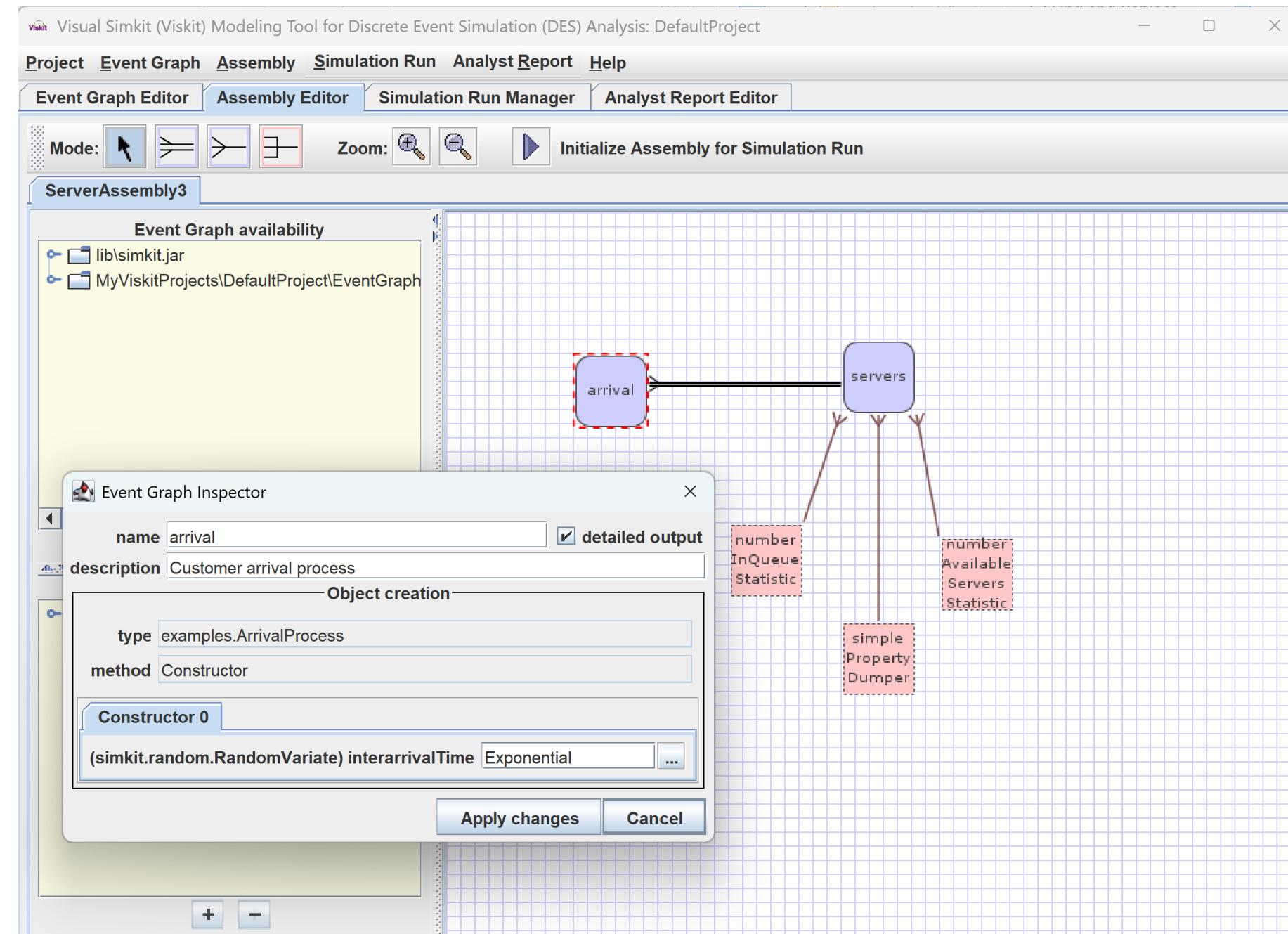
# Editing an assembly

## Inspect Metadata



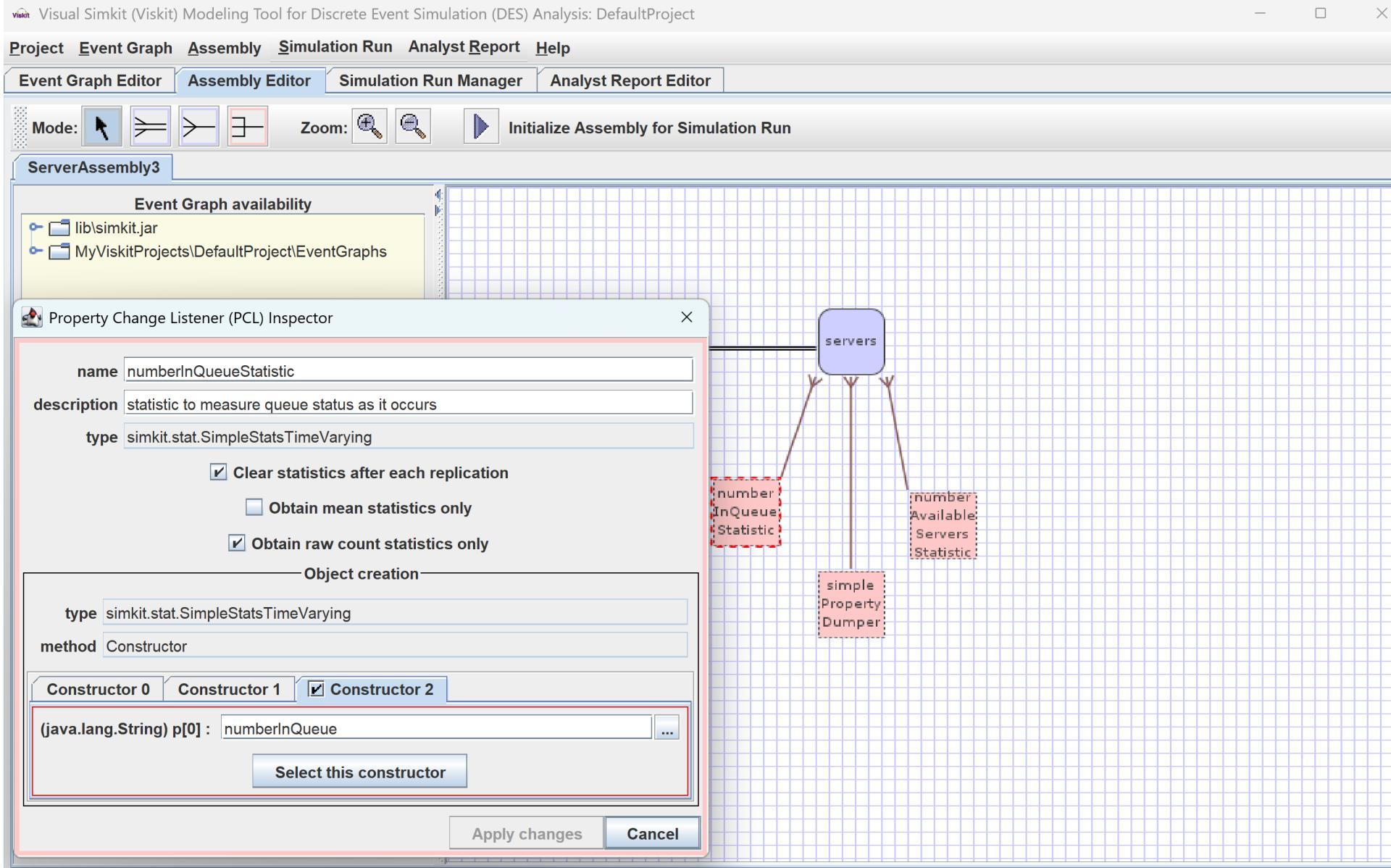
# Editing an assembly

## Inspect Event Graph nodes



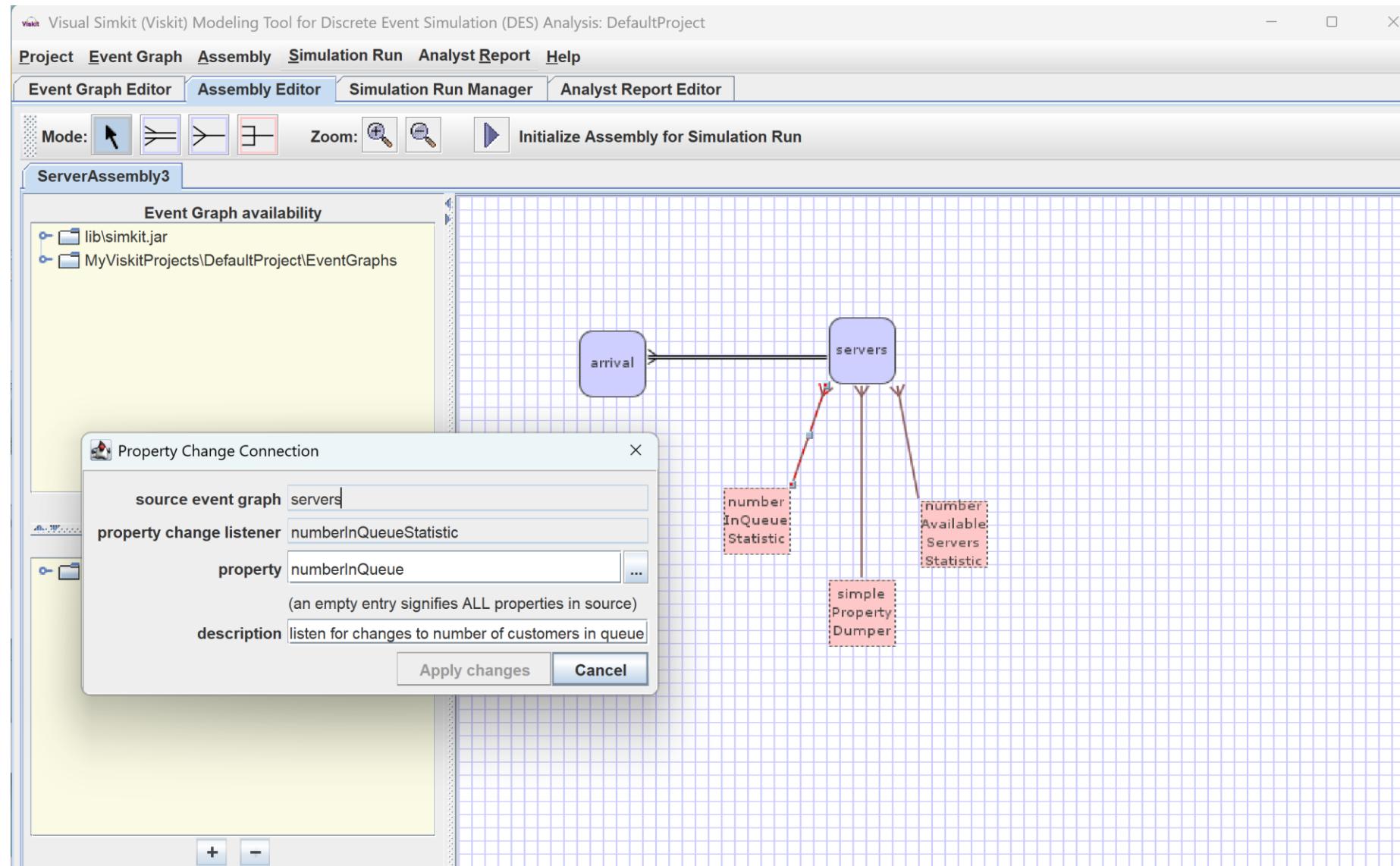
# Editing an assembly

## Inspect Property Change Listeners PCLs



# Editing an assembly

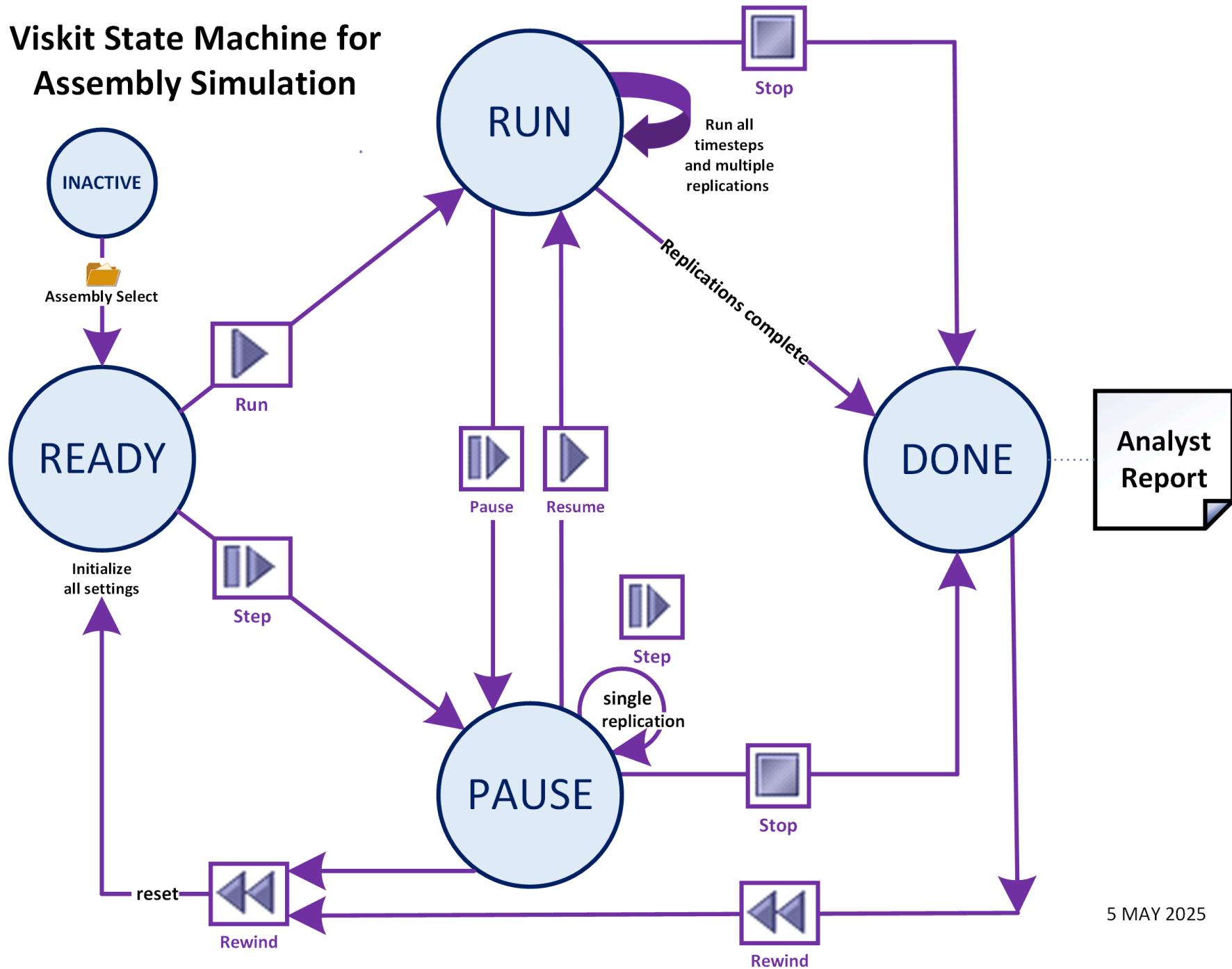
## Inspect Property Change Connection



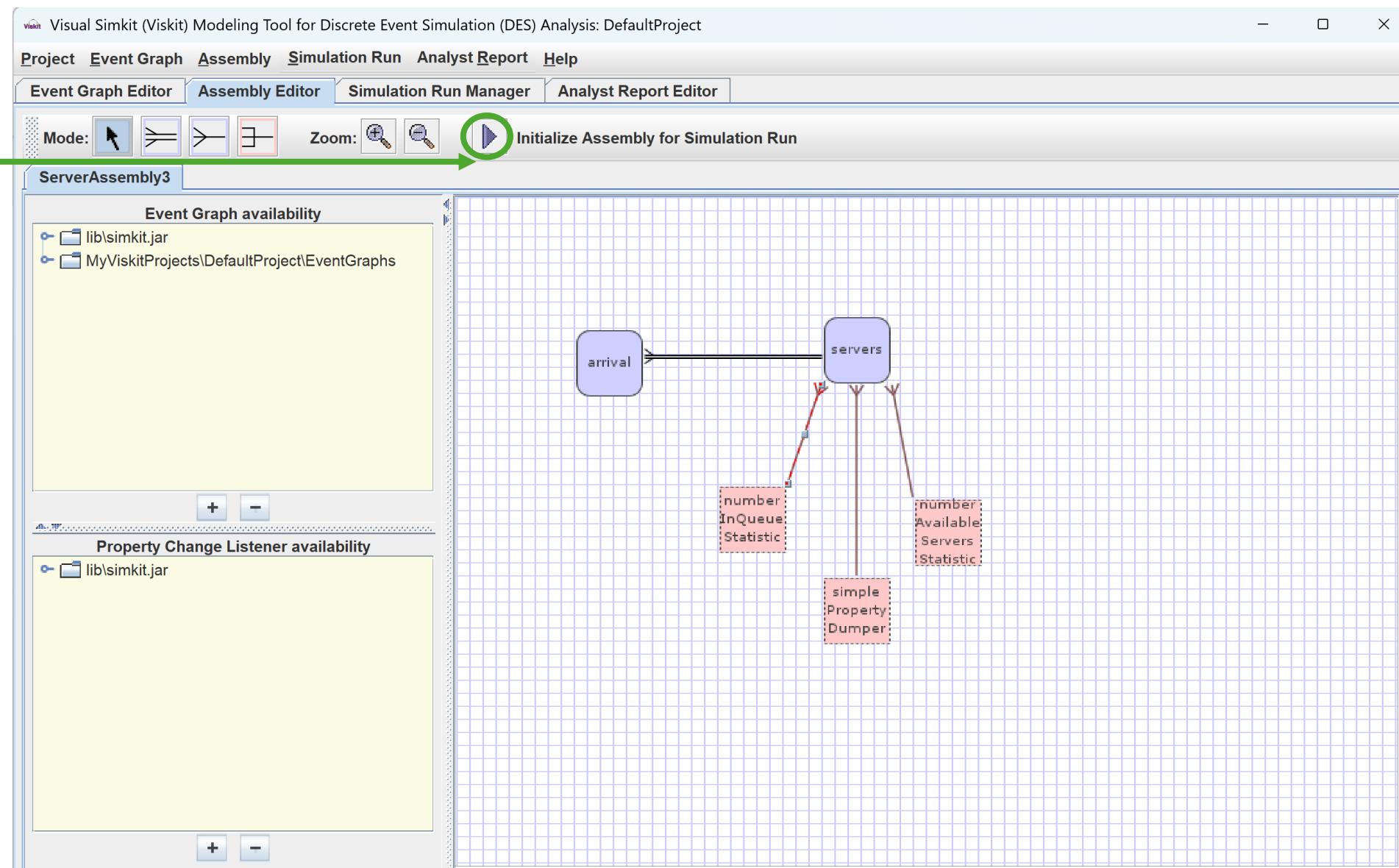
# Run Simulation Replications

Rehearse and repeat, either for preliminary tests or for numerous repetitions

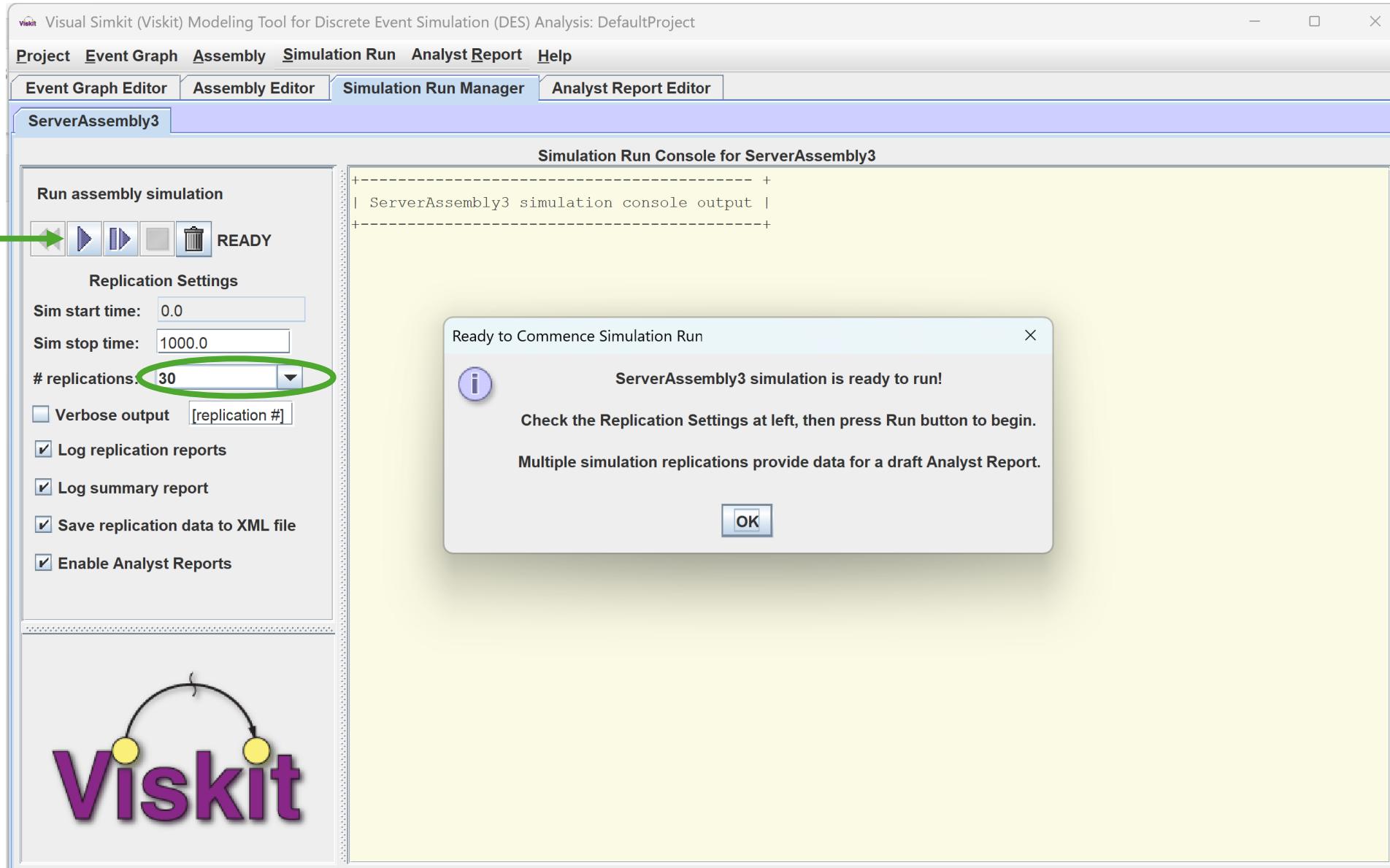
# Viskit State Machine for Assembly Simulation



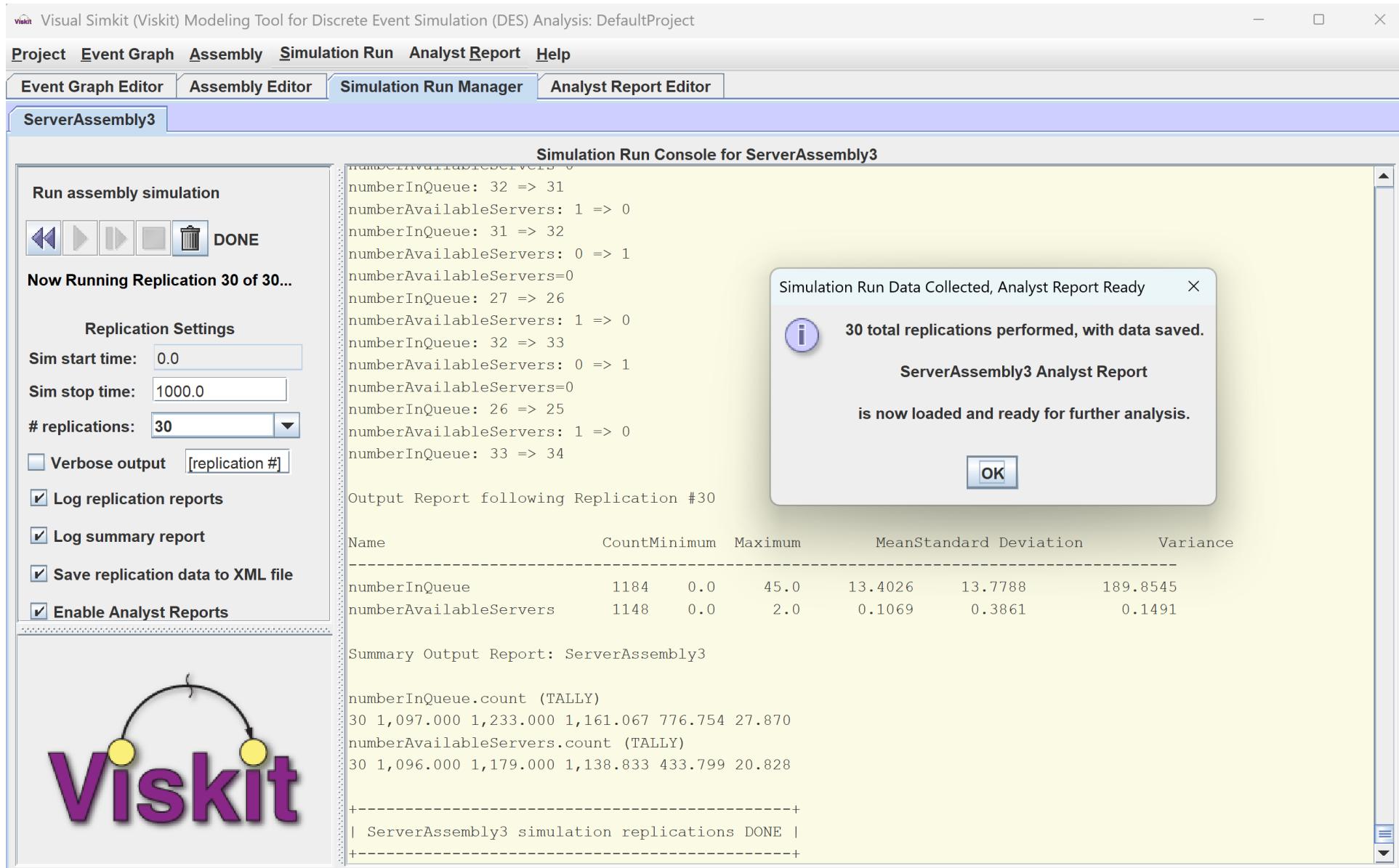
Ready  
to run  
simulations



# Get set to run simulations

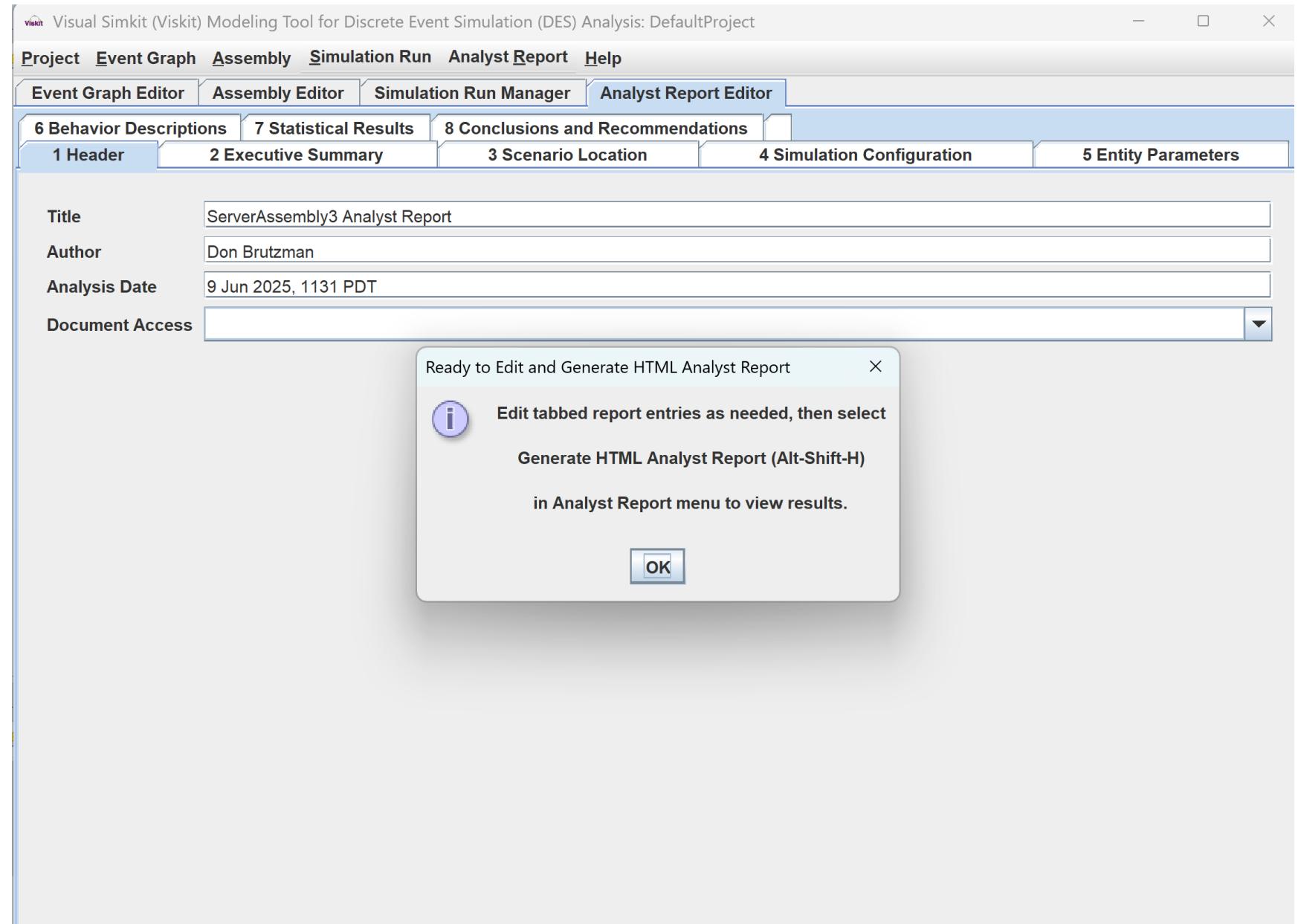


# Go! Run simulations

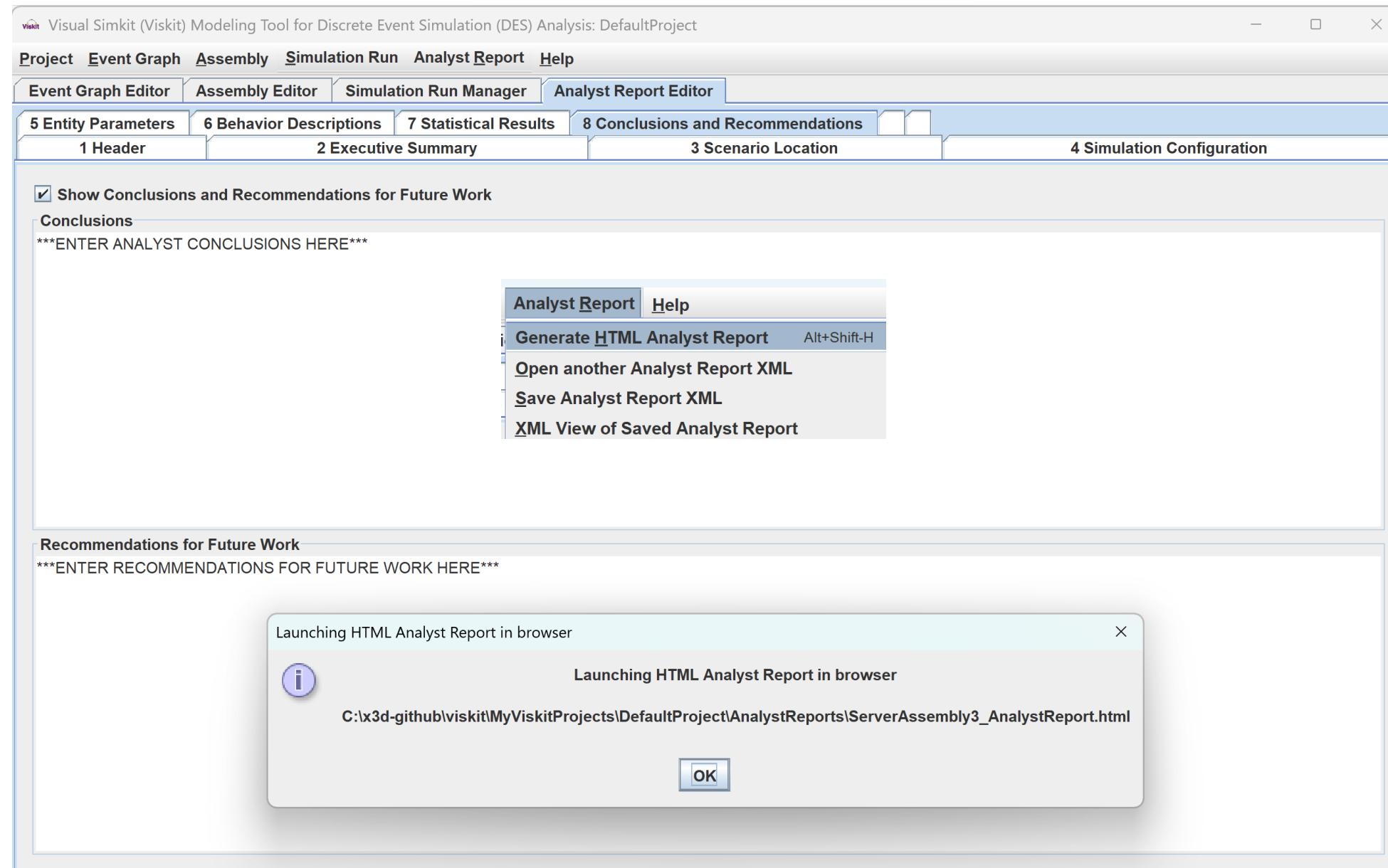


# Simulation replications complete

## Edit Analyst Report data



When  
ready,  
launch  
HTML  
Analyst  
Report  
in browser





# ServerAssembly3 Analyst Report

Analyst: **Don Brutzman**

Analysis date: **9 Jun 2025, 1131 PDT**

[Executive Summary](#) | [Scenario Location](#) | [Simulation Configuration](#) | [Entity Parameters](#) | [Behavior Descriptions](#) | [Statistical Results](#) | [Conclusions and Recommendations](#)

---

## Executive Summary

*Assessment Overview*

\*\*\*ENTER EXECUTIVE SUMMARY HERE\*\*\*

[Back to top](#)

---

## Scenario Location for the Simulation

*Description of Scenario Location Features*

\*\*\*ENTER SCENARIO LOCATION DESCRIPTION HERE\*\*\*

*Production Notes*

\*\*\*ENTER SCENARIO LOCATION PRODUCTION NOTES HERE\*\*\*

All units are meters and degrees unless otherwise noted.

*Post-Experiment Analysis of Significant Scenario Location Features*

\*\*\*ENTER SCENARIO LOCATION CONCLUSIONS HERE\*\*\*

## Simulation Configuration: Viskit Assembly Preparation for ServerAssembly3

Simulation configuration is defined by the Viskit Assembly which collects, lists, initializes, and connects all Event Graphs for participating entity models within a single scenario. The runnable assembly is then ready for repeated simulation replications, either for visual validation of behavior or statistical analysis of Measures of Effectiveness (MoEs).

Description: not provided

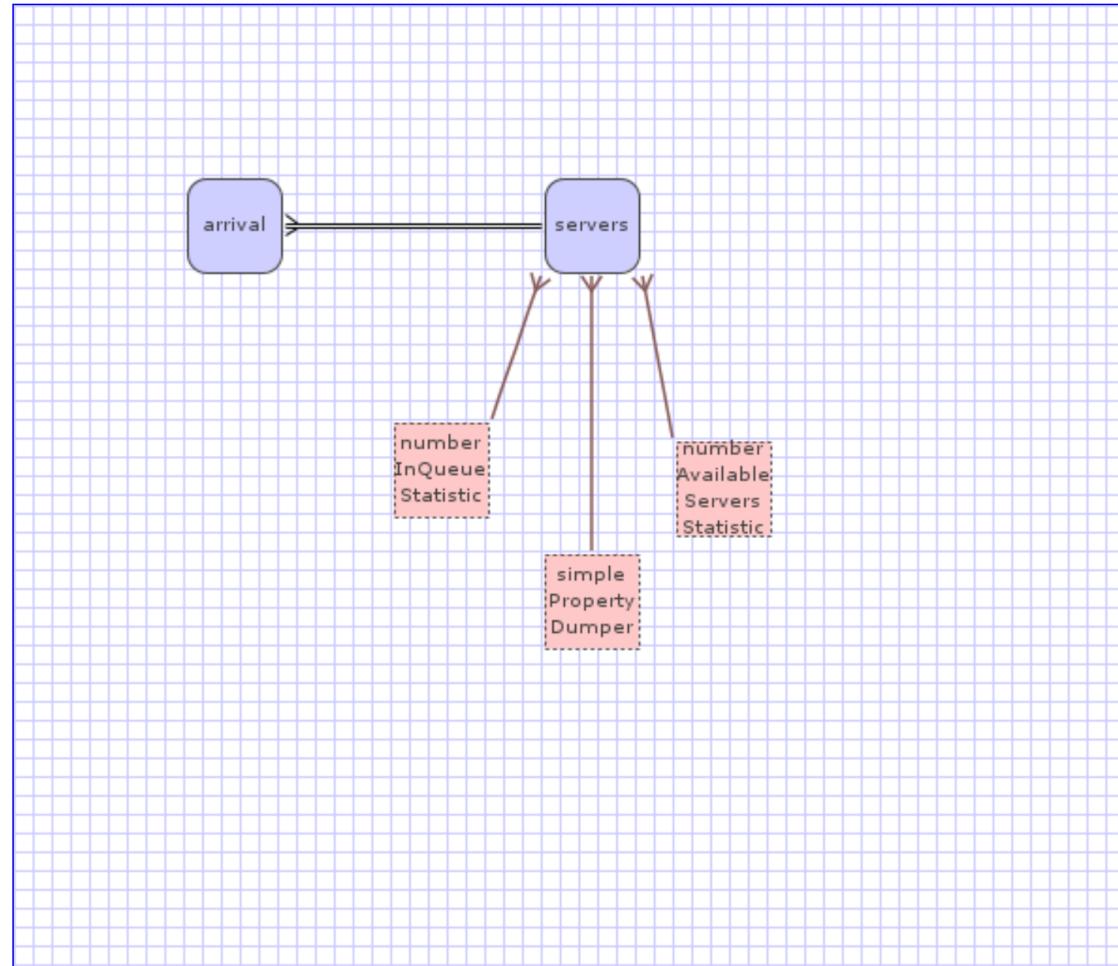


Figure 1: Simulation Assembly Combining all Simulation Entities for this Scenario Experiment

## Entity Parameters for configuring Event Graphs

Entity parameters configure the behaviors of connected event graphs.

Initialization Parameters for Simulation Entity

[Back to top](#)

### Entity Initialization Parameters for this Simulation Assembly

Initialization parameters are applied to individualize generic behavior models. These parameters customize the event-graph models.

*Entity Parameters Conclusions: Post-Experiment Analysis of Entity Behaviors*

\*\*\*ENTER ENTITY PARAMETERS CONCLUSIONS HERE\*\*\*

Initialization Parameters for Simulation Entity

---

[Back to top](#)

---

## Behavior Descriptions

*Description of Behavior Design*

**Event Graph Behavior:** examples.ArrivalProcess

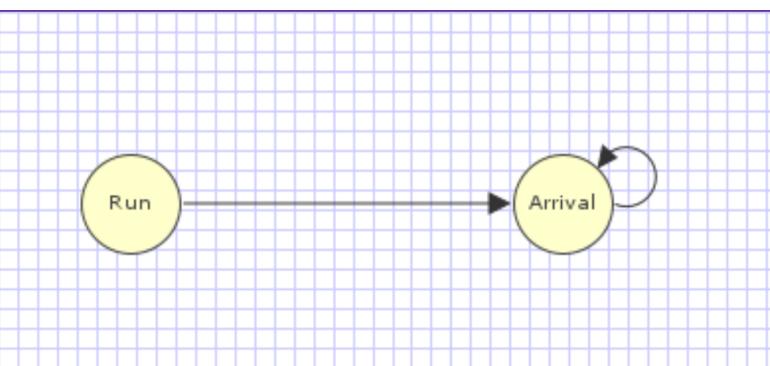


Figure 2: Event Graph for examples.ArrivalProcess

### Initialization Parameter Parameter Type Description

State Variable	Variable Type	Description
numberArrivals	int	no description found in Event Graph

[Back to top](#)

**Event Graph Behavior:** examples.SimpleServer

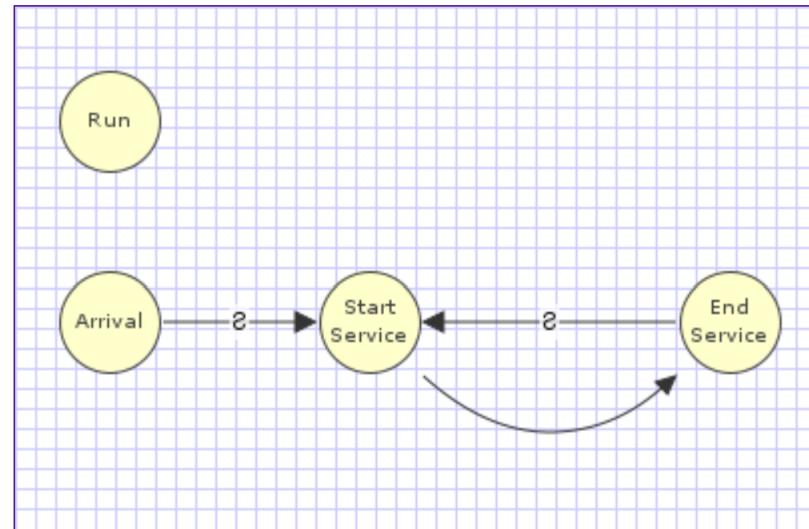


Figure 3: Event Graph for examples.SimpleServer

### Initialization Parameter Parameter Type Description

State Variable	Variable Type	Description
numberAvailableServers	int	no description found in Event Graph
numberInQueue	int	no description found in Event Graph

[Back to top](#)

## Statistical Results for the Simulation

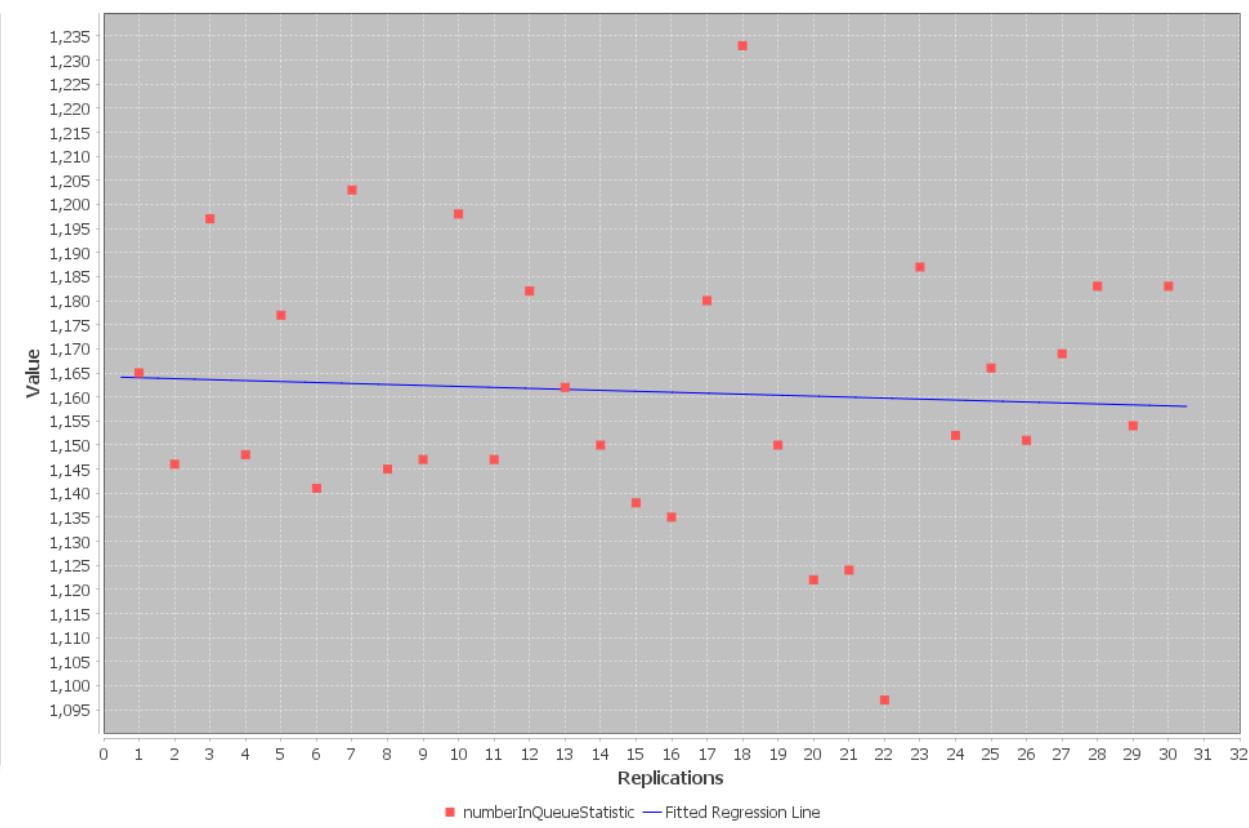
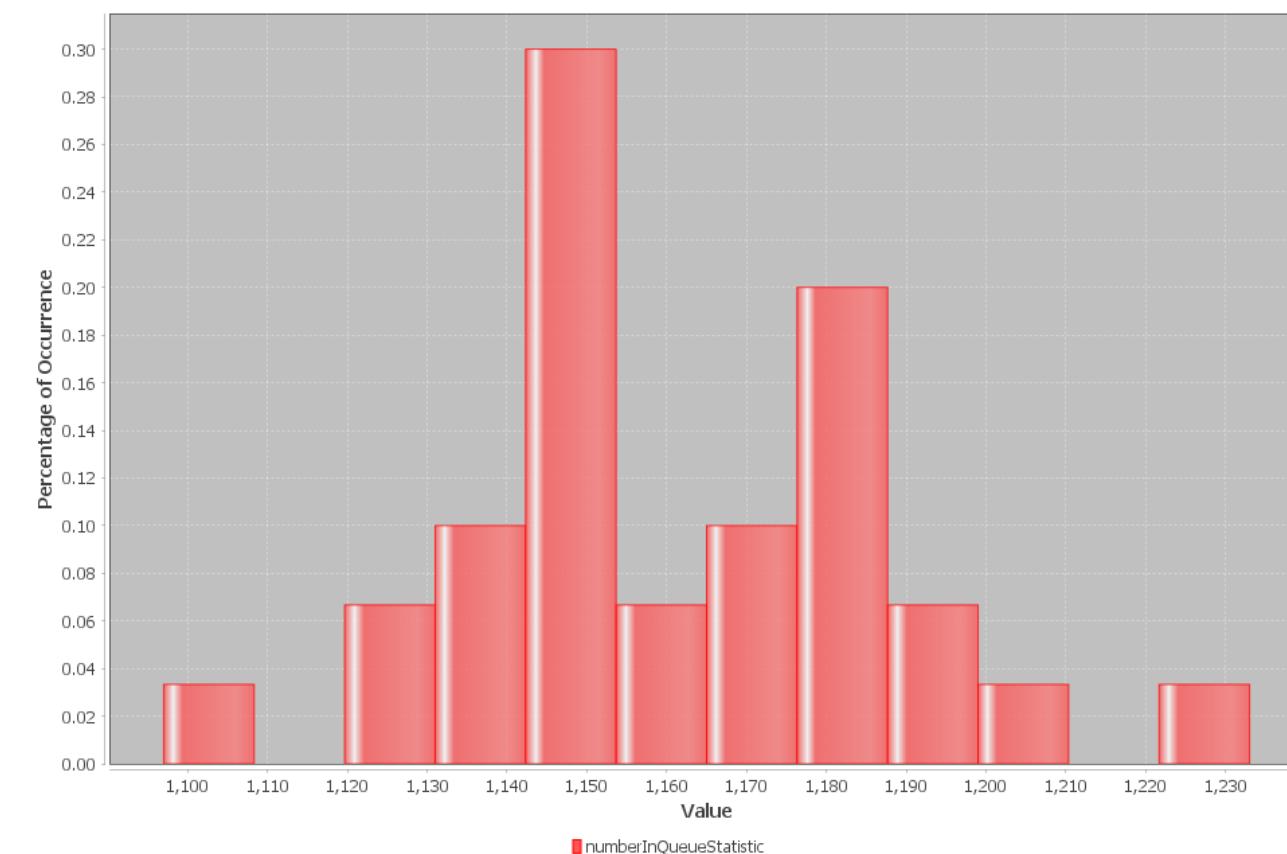
This simulation included 30 replication repetitions for the assembly of interest.

Statistical results are collected and produced by Property Change Listener (PCL) definitions in the Assembly model.

### Assembly Simulation Replication Report

#### Measure of Effectiveness (MoE)

Property: numberInQueueStatistic



Replication #	numberAvailableServersStatistic	Min	Max	Mean	StdDev	Variance
1	1133	0	2	0.054	0.274	0.075
2	1143	0	2	0.100	0.350	0.123
3	1155	0	2	0.027	0.199	0.039
4	1131	0	2	0.028	0.189	0.036
5	1145	0	2	0.032	0.204	0.042
6	1131	0	2	0.175	0.465	0.216
7	1133	0	2	0.006	0.105	0.011
8	1143	0	2	0.187	0.508	0.258
9	1143	0	2	0.147	0.448	0.201
10	1161	0	2	0.056	0.258	0.067
11	1135	0	2	0.080	0.341	0.116
12	1179	0	2	0.071	0.336	0.113
13	1131	0	2	0.063	0.331	0.110
14	1145	0	2	0.113	0.393	0.155
15	1097	0	2	0.127	0.424	0.180
16	1115	0	2	0.160	0.474	0.225
17	1159	0	2	0.150	0.445	0.198
18	1173	0	2	0.019	0.164	0.027
19	1145	0	2	0.177	0.493	0.243
20	1113	0	2	0.093	0.356	0.127
21	1115	0	2	0.087	0.344	0.118
22	1096	0	2	0.233	0.552	0.305
23	1177	0	2	0.045	0.256	0.066
24	1137	0	2	0.030	0.210	0.044
25	1123	0	2	0.003	0.079	0.006
26	1115	0	2	0.117	0.401	0.161
27	1159	0	2	0.102	0.371	0.138
28	1147	0	2	0.056	0.293	0.086
29	1139	0	2	0.059	0.290	0.084
30	1147	0	2	0.107	0.386	0.149

# Analyst Report Completion

---

## Conclusions and Recommendations

*Conclusions*

\*\*\*ENTER ANALYST CONCLUSIONS HERE\*\*\*

*Recommendations for Future Work*

\*\*\*ENTER RECOMMENDATIONS FOR FUTURE WORK HERE\*\*\*

[Back to top](#)

---

This report was autogenerated by the Viskit Event Graph and Assembly modeling tool using Simkit discrete-event simulation (DES) libraries. Online at <https://github.com/open-dis/viskit> and <https://github.com/ahbuss/Simkit>.

- Thus well suited for iterative analysis and experimentation
- Report files can be saved and reopened as needed – archival.
- Improvements ongoing and suggestions are always welcome

# Help

The screenshot shows the Viskit Help interface. The top navigation bar includes 'Help' (selected), 'Contents' (F1), 'Search' (Shift-F1), 'Tutorial', 'Help Resources' (selected), 'Simkit DES Modeling Manual', 'Simkit Java Programming Videos', 'Github Simkit Repository', 'Github Viskit Issues', and 'Github Viskit Repository'. The main content area displays the help resources.

To learn everything you want from  
Professor Arnie Buss:

[Simkit Discrete Event Simulation  
\(DES\) Modeling Manual](#)

[Simkit Java Programming Videos](#)

The screenshot shows the Viskit Help application window. The title bar says 'Viskit Help'. The left sidebar contains a tree view of help topics: Overview (Introduction selected), Contributors, Terms, Developers, Additional Resources, Tutorial (Event Graph, Create an Assembly, Simulation Run (State Machine)), Menus (Project, Event Graph, Assembly, Simulation Run, Analyst Report, Help), and Tabs (Event Graph Editor, Assembly Editor, Simulation Run Manager, Analyst Report Editor). The right panel displays the 'Introduction' section.

## Introduction

Viskit is an open-source application to build Discrete Event Simulation (DES) models that autogenerate Java source for simulation using the Simkit library.

The Viskit application enables authors to assemble a collection of event graphs that describe the behavior of multiple entities in a simulation environment.

Capabilities include Simkit Event Graphs, Assemblies, simulation replication tools, and an Analyst Report generator suitable for thesis and research efforts.

The 2025 software release is a thorough refactoring of the underlying Viskit source code, facilitating ongoing project maintenance and development.

Expected to return in future distributions: Diskit, which is a Java package of Distributed Interactive Simulation (DIS) utilities and Simkit entities for 3D movement support. The [opendis7-java](#) library is used for state-distribution capabilities that integrates the Distributed Interactive Simulation (DIS) protocol for networked agent behaviors and Live Virtual Constructive (LVC) simulations.

Software and model development efforts are ongoing.

# Infrastructure

Open source version control

Viskit codebase, Simkit analytics codebase, model libraries

# Open-source license

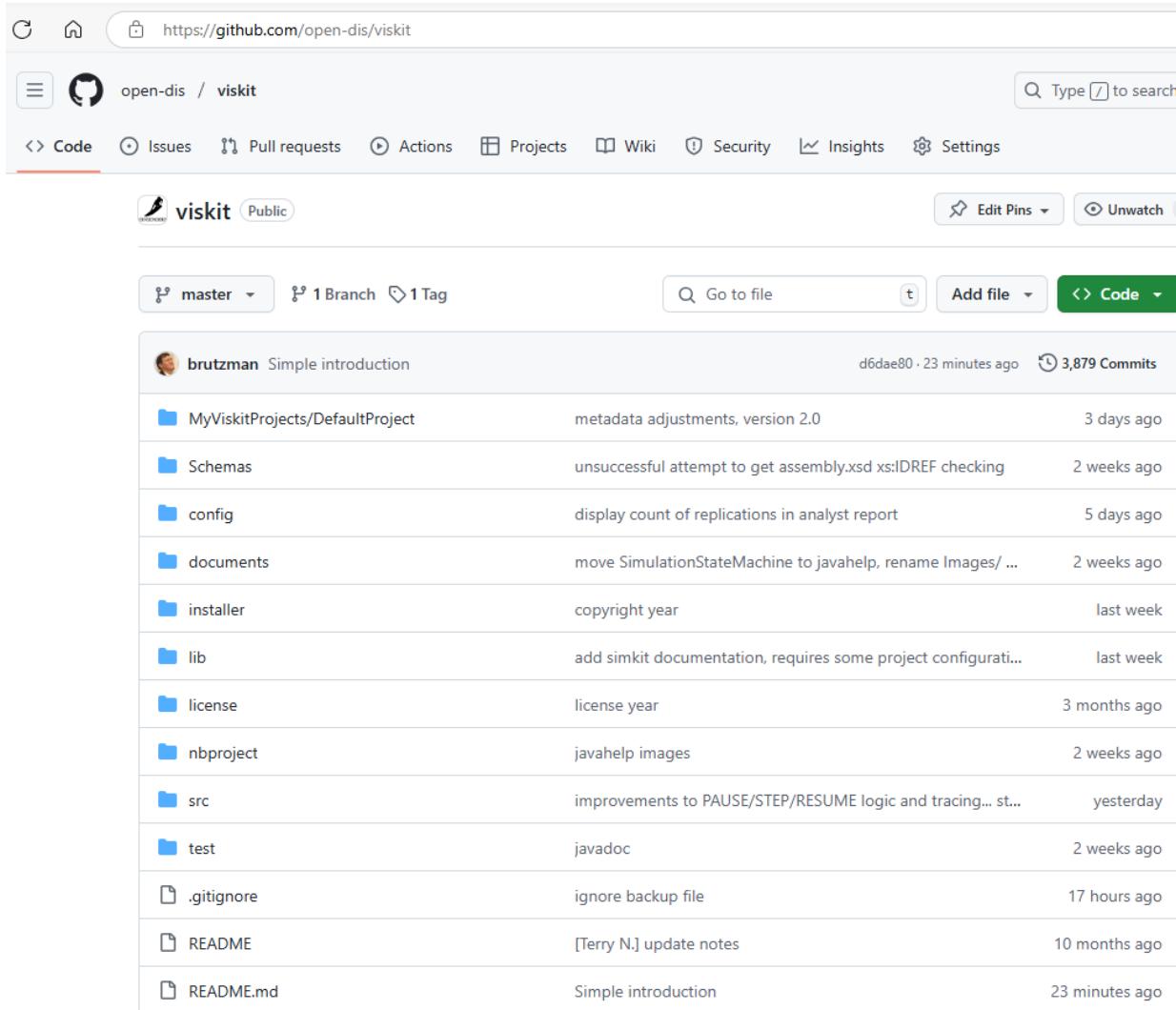
Copyright (c) 1995-2025 held by the author(s). All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the names of the [Naval Postgraduate School \(NPS\) Modeling Virtual Environments and Simulation \(MOVES\) Institute](#) nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
- THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# Open-source repository has moved to github

<https://github.com/open-dis/viskit>



- Previously hosted at <https://gitlab.nps.edu/Savage/viskit> to all NPS personnel
  - (U.S. and international)
- Inclusion under open-dis project will hopefully encourage broad use and further contributions

# MV3500 Networked Graphics for Simulation

<https://gitlab.nps.edu/Savage/NetworkedGraphicsMV3500>

The screenshot shows the GitLab interface for the 'NetworkedGraphicsMV3500' project. The left sidebar has an 'Explore' section with a search bar and a 'Project' section containing 'NetworkedGraphicsMV3500'. The main area displays the project's name, a 'master' branch dropdown, and a 'Code' button. A recent commit from 'brutzman' is shown: 'omit 20203 from SISO-REF-010, update some links' (commit ID 98408467). Below this is a table of files with their last commit and update times.

Name	Last commit	Last update
archive	remove build-impl.xml NetBeans a...	9 months ago
assignments	omit broken links in javadoc	2 months ago
conferences	Update file README.md	1 year ago
documentation	page title icon	2 years ago
examples	Security warnings in comments	2 months ago
lib	updated external jar files	2 months ago
nbproject	[Terry N.] enable viskit run from ID...	7 months ago
presentations	touchups	8 months ago
specifications	omit 20203 from SISO-REF-010, u...	2 months ago
viskit	[Terry N.] update	7 months ago

Course slidesets and examples soon to migrate to github as well

# Viskit demo video

<https://savage.nps.edu/Savage/videos/Viskit-demo-2023MAR4.mp4>

**Event Graph**

```
graph LR; Run((Run)) -- t_A --> Arrival((Arrival)); Arrival -- "N - N + 1, q.add(N)" --> N[N]; N -- t_S --> StartService((Start Service)); StartService -- "S = S - 1, h = q.removeFirst()" --> Renegue((Renegue c)); Renegue -- "R = R + 1, q.remove(c)" --> StartService; StartService -- "Q > 0" --> EndService((End Service)); EndService -- "S = S + 1" --> Arrival;
```

**Figure 4-10. Multiple Server Queue with Customers who Renegue**

As shown by the Event Graph in Figure 4-10, an arriving customer schedules the Renegue event upon arrival, so that whenever it occurs, the queue is decremented. If the StartService event occurs first, however, then the Renegue event that corresponds to that customer is removed. Note that the expression  $M + R$  that is on the cancelling edge from StartService to Renegue gives the number of the customer who has just started service.

The state variables  $M$ ,  $R$ , and  $N$  can be used to compute the proportion of customers who renege ( $R/N$ ) and the proportion of customers who receive service ( $M/N$ ). The time-varying averages of  $S$  and  $Q$  can be used to estimate the average number in the queue and the average utilization of the servers.

Note that Little's formula cannot be applied in this situation to estimate the delays in queue or the time in the system for customers who received service. This is because the queue count includes all customers and can't distinguish (before a Renegue occurs) between those who will eventually renege and those who will receive service. Applying Little's formula would require two state variables, one for served customers and one for reneging customers. Since this cannot be done until after the fact, a model that explicitly computes these times is needed (see below).

## 4.5. Containers

For certain situations it is convenient to model using containers to hold data, as the previous model illustrated. In Event Graph models a container can be thought of as a state variable with "values" that go beyond simple numerical variables.

# Contact

Don Brutzman

[brutzman@nps.edu](mailto:brutzman@nps.edu) and [don.brutzman@gmail.com](mailto:don.brutzman@gmail.com)

Terry Norbraten

[tdnorbraten@nps.edu](mailto:tdnorbraten@nps.edu) and [terry.norbraten@gmail.com](mailto:terry.norbraten@gmail.com)