Chapter 2:

Transactions - Their Creation, Movement, and Destruction
Chapter 2
Transactions (Xacts):
Their Creation, Movement, and Destruction

• Xacts: Units of Traffic in GPSS/H Models

• Pathways Along Which Xacts Move:
  Blocks and Block Diagrams

• Simulated Time and the Simulated Clock

• Creating Xacts: The GENERATE Block

• Destroying Xacts: The TERMINATE Block
  (and the Model's Termination Counter)

• A Two-Block GPSS/H Model

• The Structure of a Basic GPSS/H Model File

• Fundamental Control Statements:
  SIMULATE; START; and END

• Batch-Mode and Test-Mode Use of GPSS/H

• DOS gpssh Command Lines

• An Example of Test-Mode Tracing of Xact
  Creation, Movement, and Destruction
Transactions:
Units of Traffic in GPSS/H Models

- **Xacts:** objects that move
  (dynamic entities in a model)

- The meaning of Xacts is by analogy (or analogies)
  that exist only in the mind of the model builder

  **Examples**
  
  A patient in a hospital's emergency room
  A unit of work-in-process in a manufacturing system
  A ship in a harbor
  A worker in a manufacturing system

- Xacts are differentiated by id number
  (and in other ways as well, as we will see)

- Xacts can be visualized as stickpersons

- Xacts are created and (usually) destroyed
  as a simulation proceeds, so the number of them that exist in a model typically *varies at random*

- Xacts move one-by-one
  (GPSS/H only pays attention to one Xact at a time)

- In general, however, two or more Xacts can move concurrently (that is, at the same *simulated* time)
  (GPSS/H accomplishes concurrent Xact movement simply by moving the Xacts, one by one, at the same simulated time)
Xacts Shown as Stickpersons

Figure 2.3
Pathways Along Which Xacts Move: Blocks and Block Diagrams

Figure 2.1
A silhouette of a Block Diagram
Figure 2.2
Another silhouette of a Block Diagram
Still More About the Pathways Along Which Xacts Move

Figure 2.4
A repetition of Figure 2.2, with general Block details shown
Initiating and Ending Xact Movement

• The conditions under which movement of an Xact is initiated will be described later

• After movement of an Xact has been initiated, that Xact moves from Block to Block until it can't move any farther at that simulated time

• Three alternative reasons why an Xact might not be able to move any farther

  *(ever, or for the time being)*:

1. The Xact moves into a Block that destroys it

   *(TERMINATE Block, Chapter 2)*

2. The Xact moves into a Block at which a simulated time lag is applied to it

   *(ADVANCE Block, Chapter 3)*

3. The Xact experiences blocking; that is, it can't move from its Current Block into its Next Block Attempted

   *(SEIZE Block, Chapter 6; ENTER Block, Chapter 11)*
Simulated Time and the Simulated Clock

- Whenever an Xact moves, it does so at a particular simulated time

- For example:
  
  ...one Xact might move at time 21.9;
  
  ...then another Xact might move at time 34.6;
  
  ...then the one that moved at time 21.9 might move again at time 37.5;
  
  ...and so on

- And so simulated time is an important aspect of a GPSS/H model

- GPSS/H automatically maintains a simulated clock, and automatically keeps track of what simulated time it is
Some Characteristics of the Simulated Clock

1. The initial value of the simulated clock is 0.0

2. The value is recorded as a floating-point number
   (internally, the value is recorded with a precision of 56 binary digits, equivalent to about 16 base-10 digits)

3. Clock values can only increase
   (It's only possible to move forward in time)

4. In general, clock values increase in variable time increments. For example:

<table>
<thead>
<tr>
<th>time increment: 21.9</th>
<th>time increment: 10.6</th>
<th>time increment: 13.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>21.9</td>
<td>32.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45.2</td>
</tr>
<tr>
<td>Simulated Time</td>
<td></td>
<td>etc.</td>
</tr>
</tbody>
</table>

5. The simulated clock "stands still" while GPSS/H uses real time to move Xacts, one by one, at that simulated time

6. When no more Xacts can move at the current simulated time, GPSS/H advances the clock to the next simulated time, then moves Xacts at that time. And so on.

   (Chapter 4 explains how this next simulated time is determined)
About the Unit of Simulated Time

- What unit of time does the simulated clock record?
  - Tenths of seconds?
  - Half seconds?
  - Seconds?
  - Minutes?
  - Hours?
  - Eight-hour shifts?
  - Days?

- The modeler decides what unit of time to use in a model

- The choice typically depends on the model

- In general, the time unit
  (sometimes called the Base Time Unit)

  is chosen to be small enough to express conveniently the shortest actions that are to be included in the model
More About the Simulated Time Unit

- Here are some examples of possible choices of a Base Time Unit:

<table>
<thead>
<tr>
<th>System</th>
<th>Shortest Action to be Modeled</th>
<th>Possible Choice of a Base Time Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>microcomputer executing control algorithms for a manufacturing system</td>
<td>microcomputer task execution</td>
<td>1 microsecond? 10 microseconds? 100 microseconds?</td>
</tr>
<tr>
<td>computer system dedicated to information retrieval</td>
<td>disk rotational delays</td>
<td>100 microseconds? 1 millisecond? 0.01 seconds?</td>
</tr>
<tr>
<td>manufacturing system</td>
<td>move times for work-in-process</td>
<td>1 second? 15 seconds? 30 seconds? 1 minute?</td>
</tr>
<tr>
<td>hospital emergency room</td>
<td>check-in at the registration desk</td>
<td>1 minute?</td>
</tr>
</tbody>
</table>

- How is the modeler's choice of a Base Time Unit reflected in the model?
- All time-valued data built into the model are expressed in terms of the Base Time Unit

(The modeler does not otherwise "tell GPSS/H" what the Base Time Unit is)
How the Choice of a Base Time Unit Influences Time Values Built Into a Model (An Example)

- In a Federal Express package-handling system, suppose it takes exactly 3 minutes to convey a box from one end of a conveyor to another.

Then the following table indicates how the conveying time built into a model depends on the choice of a Base Time Unit in the model.

<table>
<thead>
<tr>
<th>Base Time Unit</th>
<th>Conveying Time (Expressed in Base Time Units)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 minute</td>
<td>3.0 (or 3)</td>
</tr>
<tr>
<td>0.5 minutes</td>
<td>6.0 (or 6)</td>
</tr>
<tr>
<td>1 second</td>
<td>180.0 (or 180)</td>
</tr>
<tr>
<td>0.5 seconds</td>
<td>360.0 (or 360)</td>
</tr>
<tr>
<td>30 minutes</td>
<td>0.1</td>
</tr>
<tr>
<td>1 hour</td>
<td>0.05</td>
</tr>
</tbody>
</table>
Creating Transactions and Introducing Them into a Model: The GENERATE Block

<table>
<thead>
<tr>
<th>Operand</th>
<th>Significance</th>
<th>Default Value or Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Average interarrival time</td>
<td>0.0 (zero)</td>
</tr>
<tr>
<td>B</td>
<td>Half-range of the uniformly distributed interarrival time random variable</td>
<td>0.0 (zero)</td>
</tr>
</tbody>
</table>

Figure 2.6

IAT: Interarrival Time

\[(A - B)^+ < \text{IAT} < (A + B)^-\]
(uniformly distributed)
GENERATE-Block Initialization

- At simulated time 0.0, GPSS/H initializes each of the GENERATE Blocks in a model

  (After that, Xact movement starts)

- Initializing a GENERATE Block involves these steps:

  1. Create an Xact and give it an id number.

     (Xacts are numbered serially in the chronological order in which they are created in the model)

  2. Determine the (usually future) simulated time at which that Xact will come into the model from its GENERATE Block

  3. Make arrangements to bring that Xact into the model from its GENERATE Block when that simulated time is reached.

     (Detailed aspects of "making such arrangements" are covered in Chapter 4)

- If a model contains two or more GENERATE Blocks, they are initialized in the top-down order in which they appear in the file containing the model
Example of a GENERATE Block, Arrival Times, and Interarrival Times

\[ \text{GENERATE} \]

\[ 15.0, 4.5 \]

Figure 2.7

Start of the simulation

Time the first Xact moves from the GENERATE Block

Interarrival time: 16.7

Time the second Xact moves from the GENERATE Block

Interarrival time: 14.3

Time the third Xact moves from the GENERATE Block

Interarrival time: 17.8

Simulated Time

Figure 2.5
Example of a GENERATE Block at Which Interarrival Times Are Deterministic

Figure 2.8
A GENERATE Block with an A Operand of 10.0, and with a default B Operand of 0.0
Destroying Xacts:
The TERMINATE Block
and the Model's Termination Counter

Figure 2.9

<table>
<thead>
<tr>
<th>Operand</th>
<th>Significance</th>
<th>Default Value or Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Decrement for the model's Termination Counter</td>
<td>0 (zero)</td>
</tr>
</tbody>
</table>

(The model's Termination Counter is a computer memory location in which an integer value can be stored)
A Two-Block GPSS/H Model

Figure 2.10

The Model's Termination Counter

(The model's Termination Counter is a computer memory location in which an integer value can be stored)

<table>
<thead>
<tr>
<th>Label (2 - 9)</th>
<th>Operation (11 - 20)</th>
<th>Operands (22 - 72)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATE</td>
<td>15.0, 4.5</td>
<td>Transactions arrive, one by one</td>
</tr>
<tr>
<td>TERMINATE</td>
<td>1</td>
<td>Transactions are destroyed, one by one, each reducing the value of the model's TC by 1</td>
</tr>
</tbody>
</table>

Figure 2.11
The Structure of a Basic GPSS/H Model File

Figure 2.17
## The SIMULATE Control Statement

<table>
<thead>
<tr>
<th>Label or Operand</th>
<th>Significance</th>
<th>Default Value or Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Never used</td>
<td>Must Default (Compile-time error otherwise)</td>
</tr>
<tr>
<td>A</td>
<td>Time Limit Operand (integer or decimal number; units are CPU minutes unless followed by an S, then units are CPU seconds)</td>
<td>No internal Time Limit will be set</td>
</tr>
</tbody>
</table>

Figure 2.12
### Examples of the SIMULATE Control Statement

<table>
<thead>
<tr>
<th></th>
<th>Label (2 - 9)</th>
<th>Operation (11 - 20)</th>
<th>Operands (22 - 72)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SIMULATE</td>
<td>3s</td>
<td>Example 1 (3 second CPU Time Limit)</td>
</tr>
<tr>
<td></td>
<td>SIMULATE</td>
<td>0.5</td>
<td>Example 2 (0.5 minute CPU Time Limit)</td>
</tr>
<tr>
<td></td>
<td>SIMULATE</td>
<td></td>
<td>Example 3 (no Time Limit)</td>
</tr>
</tbody>
</table>
The START Control Statement

<table>
<thead>
<tr>
<th>Label (2 - 9)</th>
<th>Operation (11 - 20)</th>
<th>Operands (22 - 72)</th>
</tr>
</thead>
<tbody>
<tr>
<td>see below</td>
<td>START</td>
<td>A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Label or Operand</th>
<th>Significance</th>
<th>Default Value or Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>START-statement identifier</td>
<td>Unlabeled statement</td>
</tr>
<tr>
<td>A</td>
<td>Initial value of the model's Termination Counter</td>
<td>Defaulting results in a compile-time error</td>
</tr>
</tbody>
</table>

Figure 2.14

The Model's Termination Counter

(The model's Termination Counter is a computer memory location in which an integer value can be stored)
## Examples of the START Control Statement

<table>
<thead>
<tr>
<th>Label (2 - 9)</th>
<th>Operation (11 - 20)</th>
<th>Operands (22 - 72)</th>
<th>Example</th>
<th>Initial TC Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td>25</td>
<td>Example 1</td>
<td>(initial TC value: 25)</td>
<td></td>
</tr>
<tr>
<td>START</td>
<td>1</td>
<td>Example 2</td>
<td>(initial TC value: 1)</td>
<td></td>
</tr>
<tr>
<td>START</td>
<td>100</td>
<td>Example 3</td>
<td>(initial TC value: 100)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.15

---

The Model's Termination Counter

(The model's Termination Counter is a computer memory location in which an integer value can be stored)
The END Control Statement

<table>
<thead>
<tr>
<th>Label (2 - 9)</th>
<th>Operation (11 - 20)</th>
<th>Operands (22 - 72)</th>
</tr>
</thead>
<tbody>
<tr>
<td>see below END</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Label or Operand</th>
<th>Significance</th>
<th>Default Value or Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>END-statement identifier</td>
<td>Unlabeled statement</td>
</tr>
</tbody>
</table>

Figure 2.16
### A Complete GPSS/H Model File

<table>
<thead>
<tr>
<th>Label (2 - 9)</th>
<th>Operation (11 - 20)</th>
<th>Operands (22 - 72)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIMULATE</td>
<td>3S</td>
<td>set a 3 second Time Limit</td>
</tr>
</tbody>
</table>

* Model Segment 1 (Creation and Destruction of Xacts) *

<table>
<thead>
<tr>
<th>Operation</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATE</td>
<td>15.0,4.5</td>
<td>Transactions arrive, one by one</td>
</tr>
<tr>
<td>TERMINATE</td>
<td>1</td>
<td>Transactions are destroyed, one by one, each reducing the value of the model's TC by 1</td>
</tr>
</tbody>
</table>

* Run-Control Statements *

<table>
<thead>
<tr>
<th>Operation</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td>25</td>
<td>set the initial TC value = 25; initialize the GENERATE Block; enter the Xact-Movement Phase</td>
</tr>
<tr>
<td>END</td>
<td></td>
<td>end of Model-File execution</td>
</tr>
</tbody>
</table>
The Steps Followed to Execute a Model File in Batch Mode

1. Model Compilation and Compiler Report
   - Components of Compiler Report:
     - Source Echo (Enhanced Copy of Model File)
     - Compile-Time Warning and Error Messages
     - Dictionary
     - Cross-Reference Listing
     - Summary of Storage Requirements
   - SIMULATE
   - Compile-Time Error(s), or no "SIMULATE"
   - Stop

2. Control-Statement Execution
   - "START"
   - "END"
   - Computer-Usage Report
   - Stop
   - Execution-Time Error
   - Error Report
   - Stop

3. Transaction Movement (Block-Statement Execution)
   - TC ≤ 0
   - Postsimulation Report

4. Postsimulation Report

Figure 2.19
The Spirit of Test-Mode Use of GPSS/H

A GPSS/H Model File

The GPSS/H compiler and run-time support system

Test-Mode User at a Microcomputer or Terminal
The Spirit of Test-Mode Use of GPSS/H

A GPSS/H Model File

The GPSS/H compiler and run-time support system

1. `gpss` file name et cetera
2. `gpss` file name et cetera
3. `gpss` file name et cetera
4. "poised at" et cetera
5. `gpss` file name et cetera

Test-Mode User at a Microcomputer or Terminal
Forming DOS Command Lines to Run GPSS/H in Test Mode
(Reference: Appendix B)

```
gpssh  file_name.ext  [options]
```

Table B.2
Examples of `gpssh` command lines for Test-Mode simulations

<table>
<thead>
<tr>
<th>Example 1:</th>
<th><code>gpssh  model6a.gps  tvtnw</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Comments:</td>
<td>This is perhaps the most frequently used <code>gpssh</code> command-line form for a Test-Mode simulation. The screen is split into Source, Status, and Dialog Windows, Compile-Time Warning Messages are suppressed, and the Postsimulation Report is directed to the screen.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example 2:</th>
<th><code>gpssh  model9a.gps  tv</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Comments:</td>
<td>In this Test-Mode simulation, the screen is split into Source, Status, and Dialog Windows, Compile-Time Warning Messages are directed to the screen, and no Postsimulation Report will come to the screen.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example 3:</th>
<th><code>gpssh  model11a.gps  test</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Comments:</td>
<td>In this Test-Mode simulation, the entire screen is used as a Dialog Window. Compile-Time Warning Messages are directed to the screen, and no Postsimulation Report will come to the screen.</td>
</tr>
</tbody>
</table>
Why Simulate in Test Mode?

- There are a number of reasons to simulate in Test Mode. Among them are these:

1. **Learning How GPSS/H Works**

   By tracing the step-by-step moves made by Xacts as a simulation proceeds, the principles and logic on which GPSS/H is based can be personally experienced.

2. **Model Debugging**

   When a simulation run aborts, GPSS/H produces a message telling which Xact was trying to move from which Block into which next Block and at which clock time when it became necessary to pull the plug.

   To investigate what went wrong, the simulation can then be re-run in Test Mode to examine the behavior of the model in detail just before the error condition occurs. This often brings the modeling error into focus for the model builder.

3. **Model Verification** *(Did we build the model right?)*

   In complex models, Xacts often have multiple options open to them. Do Xacts always choose the correct options as the simulation proceeds? Or has the model builder made logical errors in building the model?

   *(Logical errors don't always lead to execution errors, as we will see, e.g. Exercise 1, Section 6.13.)*

   In Test Mode, the movement of arbitrarily chosen Xacts can be traced, step by step, to see if these Xacts make the right moves at the right times, according to the state of the model at those times.
The Steps in a Simulation Study

1. Formulate problem and plan the study
2. Collect data and define a model
3. Valid? [No]
4. Construct a computer program and verify
5. Make pilot runs
6. Valid? [No]
7. Design experiments
8. Make production runs
9. Analyze output data
10. Document and implement results

Figure 1.2
Test-Mode Commands

- GPSS/H provides a set of commands to support Test-Mode use of GPSS/H models.

- The commands are used to:
  
  ...set interrupt conditions,
  ...bring about Xact movement,
  ...display information about the model,
  ...control use of the screen, and
  ...scroll up and down and left and right

- Test-Mode commands are issued by the user during a Test-Mode simulation
  
  (they are not part of a Model File)

- The commands can be keyed in lower case, UPPER CASE, or MiXeD CaSe
  
  (In contrast, the non-comments portions of Model-File statements must be in UPPER CASE)

- To speed things up, Test-Mode commands can be abbreviated
  
  (Details will be provided when specific Test-Mode commands are introduced)

- The keyboard's Function keys (F-keys) are pre-programmed to correspond to some of the Test-Mode commands
  
  (Details are be provided in Chapter 4)
Test-Mode Interrupt Conditions

- An interrupt condition is a condition in an ongoing simulation under which:
  
  GPSS/H is to *interrupt*  
  (*suspend*) the simulation, and  
  
  *return control* to the Test-Mode user

- There are several types of interrupt conditions

- For example, an interrupt can be caused to occur when a
  
  *specified incremental Block-execution count has been reached*

  (Additional types of interrupt conditions will be considered later)

- The Test-Mode *step* command is used for this purpose

- Details of the *step* command will be discussed shortly
A Basic View of a Test-Mode Simulation

from compilation

(1) GPSS/H is poised to execute a START statement

Ready!

User In Test Mode
(the user can turn interrupt conditions on/off and display information, then resume Model File execution or quit)

(2) resume (⇒ START statement execution, etc.)

Transaction Movement
(Block-Statement Execution)

(3) interrupt

(4) resume

(5) quit

Stop

Figure 2.20
About the Message Issued When An Interrupt Occurs

• When an interrupt occurs, GPSS/H issues a message to the Test-Mode user providing some information about the state of the simulation.

• For example, when a step-based interrupt occurs, the following type of message is displayed on the Test-Mode user's screen:

  XACT 3 POISED AT BLOCK 6. RELATIVE CLOCK: 104.3

• "Poised at" means "is about to try to move into"

• As a result, when the simulation is later resumed, GPSS/H will begin by trying to move the indicated Xact into whatever Block occupied the indicated Location.

• As we will see later, the type of interrupt message issued by GPSS/H depends on the type of interrupt condition which has occurred.
The Step Command
(A Test-Mode Command)

step

- When a step command is issued, GPSS/H:
  1. resumes (or starts) Xact movement,
  2. continues until one more Block has been executed, (that is, "steps through one more Block execution,") then
  3. issues a "poised at" message and gives control back to the Test-Mode user

- step can be abbreviated to ste, or st, or s

ste
  st
  s

- Pressing F10 causes a step command to be issued
  (echoed onto the Test-Mode screen and executed)
Stepping Through a GPSS/H Simulation in Test Mode (One Block Execution at a Time)

1. the user has issued a `gppsh file_name.gps tvtnw` command and, after the Ready! response, is being prompted (:) for a command.

2. the user issues a command to execute one Block ("step"; or "s"; or press F10)

3. GPSS/H proceeds until one more Block has been executed, then issues a "poised at" message and gives control to the user.

4. (4) resume
   the user decides whether to resume the simulation, or quit

5. (5) quit
   the user issues a command to quit quickly ("qq")

Stop

Figure 2.21
A Complete GPSS/H Model File

<table>
<thead>
<tr>
<th>Label (2 - 9)</th>
<th>Operation (11 - 20)</th>
<th>Operands (22 - 72)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIMULATE</td>
<td>3S</td>
<td>set a 3 second Time Limit</td>
</tr>
</tbody>
</table>

******************************************************************************

* Model Segment 1 (Creation and Destruction of Xacts) *

******************************************************************************

<table>
<thead>
<tr>
<th>Operation</th>
<th>Transactions arrive, one by one</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATE</td>
<td>15.0, 4.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operation</th>
<th>Transactions are destroyed, one by one, each reducing the value of the model's TC by 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>TERMINATE</td>
<td>1</td>
</tr>
</tbody>
</table>

******************************************************************************

* Run-Control Statements *

******************************************************************************

<table>
<thead>
<tr>
<th>Operation</th>
<th>Initial TC value = 25;</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td>25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>initialize the GENERATE Block;</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>enter the Xact-Movement Phase</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>END</th>
<th>end of Model-File execution</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLOCK CURRENT</td>
<td>TOTAL</td>
</tr>
<tr>
<td>---------------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

S/C: OFF  ABS CLOCK:  0.  REL CLOCK:  0.  TTG: 0

---

XACT:  CURBLK:  NEXTBLK:  CHAINS:  PC:

MARK-TIME:  MOVE-TIME:  PRIORITY:

---

GPSS/H IS A PROPRIETARY PRODUCT OF,
AND IS USED UNDER A LICENSE GRANTED BY,

WOLVERINE SOFTWARE CORPORATION
4115 ANNANDALE ROAD
ANNANDALE, VIRGINIA 22003, USA

Ready!

: 2-31
### Figure 2.18, Screen 2

<table>
<thead>
<tr>
<th>BLOCK</th>
<th>CURRENT</th>
<th>TOTAL</th>
<th>fig218.gps SOURCE CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td>GENERATE 15.0,4.5 Transactions a</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0</td>
<td>TERMINATE 1 Transactions a</td>
</tr>
</tbody>
</table>

S/C: OFF ABS CLOCK: 12.9327 REL CLOCK: 12.9327 TTG: 2

XACT: 1 CURBLK: 1 NEXTBLK: 2 CHAINS: CEC PC:
MARK-TIME: 12.9327 MOVE-TIME: ----- PRIORITY: 0

WOLVERINE SOFTWARE CORPORATION
4115 ANNANDALE ROAD
ANNANDALE, VIRGINIA 22003, USA

Ready!
: step

XACT 1 POISED AT BLOCK 2. RELATIVE CLOCK: 12.9327
:
### Figure 2.22b Repeated as Figure 2.22bb
(Source, Status, and Dialog Windows Highlighted)

<table>
<thead>
<tr>
<th>BLOCK CURRENT</th>
<th>TOTAL</th>
<th>fig218.gps SOURCE CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>GENERATE 15.0,4.5 Transactions a</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>TERMINATE 1 Transactions a</td>
</tr>
</tbody>
</table>

#### Source Window

- **S/C:** OFF
- **ABS CLOCK:** 12.9327
- **REL CLOCK:** 12.9327
- **TTG:** 2

<table>
<thead>
<tr>
<th>XACT: 1</th>
<th>CURBLK: 1</th>
<th>NEXTBLK: 2</th>
<th>CHAINS: CEC</th>
<th>PC:</th>
</tr>
</thead>
<tbody>
<tr>
<td>MARK-TIME: 12.9327</td>
<td>MOVE-TIME: -----</td>
<td>PRIORITY: 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Status Window

```
WOLVERINE SOFTWARE CORPORATION
4115 ANNANDALE ROAD
ANNANDALE, VIRGINIA 22003, USA

Ready!
: step

XACT 1 POISED AT BLOCK 2. RELATIVE CLOCK: 12.9327
: 
```
<table>
<thead>
<tr>
<th>TOTAL</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

|---------|---------------------|---------------------|------------|---------|------------|-------------------|-------------|--------------------------|

<table>
<thead>
<tr>
<th>XACT 1</th>
<th>POISED AT BLOCK 2. RELATIVE CLOCK: 12.9327</th>
</tr>
</thead>
<tbody>
<tr>
<td>XACT 1</td>
<td>DESTROYED AT BLOCK 2. RELATIVE CLOCK: 12.9327</td>
</tr>
<tr>
<td>Ready!</td>
<td>step</td>
</tr>
<tr>
<td>:</td>
<td>step</td>
</tr>
</tbody>
</table>
Figure 2.18, Screen 4

<table>
<thead>
<tr>
<th>BLOCK CURRENT</th>
<th>TOTAL</th>
<th>fig218.gps SOURCE CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>GENERATE 15.0,4.5</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>TERMINATE 1</td>
</tr>
</tbody>
</table>

S/C: OFF ABS CLOCK: 27.9268 REL CLOCK: 27.9268 TTG: 1

XACT: 2 CURBLK: 1 NEXTBLK: 2 CHAINS: CEC PC:
MARK-TIME: 27.9268 MOVE-TIME: ---- PRIORITY: 0

: step

XACT 1 POISED AT BLOCK 2. RELATIVE CLOCK: 12.9327
: step

XACT 1 DESTROYED AT BLOCK 2. RELATIVE CLOCK: 12.9327
: step

XACT 2 POISED AT BLOCK 2. RELATIVE CLOCK: 27.9268
:
### Figure 2.18, Screen 5

<table>
<thead>
<tr>
<th>BLOCK CURRENT</th>
<th>TOTAL</th>
<th>fig218.gps SOURCE CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>GENERATE</strong> 15.0,4.5 Transactions a</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>TERMINATE</strong> 1 Transactions a</td>
</tr>
</tbody>
</table>

---

S/C: OFF ABS CLOCK: 27.9268 REL CLOCK: 27.9268 TTG: 0

---

XACT: 2 CURBLK: 1 NEXTBLK: 2 CHAINS: PC:

MARK-TIME: 27.9268 MOVE-TIME: ----- PRIORITY: 0

---

: step

XACT 1 DESTROYED AT BLOCK 2. RELATIVE CLOCK: 12.9327

: step

XACT 2 POISED AT BLOCK 2. RELATIVE CLOCK: 27.9268

: step

REQUESTING OUTPUT IN (CONTROL) STATEMENT NUMBER 16. RELATIVE CLOCK: 27.9268

:
## Figure 2.18, Screen 6

<table>
<thead>
<tr>
<th>BLOCK CURRENT</th>
<th>TOTAL</th>
<th>fig218.gps SOURCE CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>GENERATE 15.0,4.5</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>TERMINATE 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transactions a</td>
</tr>
</tbody>
</table>

S/C: OFF  ABS CLOCK: 27.9268  REL CLOCK: 27.9268  TTG: 0

XACT: 2  CURBLK: 1  NEXTBLK: 2  CHAINS:  PC:

MARK-TIME: 27.9268  MOVE-TIME: -----  PRIORITY: 0

: step

XACT 1 DESTROYED AT BLOCK 2.  RELATIVE CLOCK: 12.9327
: step

XACT 2 POISED AT BLOCK 2.  RELATIVE CLOCK: 27.9268
: step

REQUESTING OUTPUT IN (CONTROL) STATEMENT NUMBER 16.  RELATIVE CLOCK: 27.9268
: qq
Simulation begins.

GPSS/H IS A PROPRIETARY PRODUCT OF,
AND IS USED UNDER A LICENSE GRANTED BY,

WOLVERINE SOFTWARE CORPORATION
4115 ANNANDALE ROAD
ANNANDALE, VIRGINIA 22003, USA

Ready!
: step

XACT 1 POISED AT BLOCK 2. RELATIVE CLOCK: 12.9327
: step

XACT 1 DESTROYED AT BLOCK 2. RELATIVE CLOCK: 12.9327
: step

XACT 2 POISED AT BLOCK 2. RELATIVE CLOCK: 27.9268
: step

REQUESTING OUTPUT IN (CONTROL) STATEMENT NUMBER 16. RELATIVE CLOCK: 27.9268
: qq

C:>
An Extended View of a Test-Mode Simulation

1. from compilation
   - GPSS/H is poised to execute a "START" statement

2. Ready!
   - resume (=> START-statement execution, etc.)

3. Transaction Movement (Block-Statement Execution)
   - (3) interrupt
   - (4) resume
   - (6) output pending ("requesting output in (control statement)..."

4. Execution-Error Report
   - (9) Error

5. quit
   - Stop
   - (8) resume

6. Postsimulation Report

Figure 2.23