Chapter 3:
More About Transaction Creation and Movement
Chapter 3
More About Transaction Creation and Movement

- Temporarily Suspending Xact Movement: The ADVANCE Block

- A Test-Mode Trace of the ADVANCE Block at Work

- The Step Command Revisited (Extended Step Counts)

- The GENERATE Block Revisited (Offset-Interval and Limit-Count Operands)

- Transaction Time-Ties

- Transaction Priority Level

- Block Labels and Nonsequential Xact Movement
Suspending Transaction Movement: The ADVANCE Block

**ADVANCE**

A, B

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

Figure 3.1

**HT: Holding Time**

\[(A - B) < HT < (A + B)\]

(uniformly distributed)

\[(A-B)^+\] A \[(A+B)^-\]

**HTs**
Example of an ADVANCE Block

ADVANCE
12.5, 3.2

Hypothesized time that an Xact moves into the "ADVANCE 12.5, 3.2" Block

Hypothesized holding time: 14.3

16.7 Simulated Time 31.0

Time the Xact moves out of the "ADVANCE 12.5, 3.2" Block
Example of a Deterministic ADVANCE Block

Hypothesized time that an Xact moves into the "ADVANCE 60" Block

holding time: 60.0

Time the Xact moves out of the "ADVANCE 60" Block

25.6 85.6

Simulated Time
Examples of ADVANCE Blocks with Holding Times of 0.0 (Zero)

All three of these ADVANCE Blocks specify a holding time of 0.0:

- ADVANCE 0
- ADVANCE 0, 0
- ADVANCE

Hypothesized time that an Xact moves into any of the three ADVANCE shown above

Time the Xact moves out of the ADVANCE Block

132.5

Simulated Time
A Model File for Tracing
ADVANCE-Block Operation

SIMULATE
*
GENERATE  50
ADVANCE   60
TERMINATE 1
*
START    1
*
END

Figure 3.2
Block-Diagram for Figure 3.2
(Termination Counter, START, and Time Axis also Shown)

The Model's Termination Counter
Screen 1 of Five Computer Screens
Resulting when the Figure 3.2 Model File
is Run in Test Mode

<table>
<thead>
<tr>
<th>BLOCK</th>
<th>CURRENT</th>
<th>TOTAL</th>
<th>fig32.gps SOURCE CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td>GENERATE 50</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0</td>
<td>ADVANCE 60</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0</td>
<td>TERMINATE 1</td>
</tr>
</tbody>
</table>

S/C: OFF  ABS CLOCK: 50.0000  REL CLOCK: 50.0000  TTG: 1

XACT: 1  CURBLK: 1  NEXTBLK: 2  CHAINS: CEC  PC:
MARK-TIME: 50.0000  MOVE-TIME: ------  PRIORITY: 0

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

Ready!
: step

XACT 1 POISED AT BLOCK 2. RELATIVE CLOCK: 50.0000
:
### Figure 3.3, Screen 2

<table>
<thead>
<tr>
<th>BLOCK CURRENT</th>
<th>TOTAL</th>
<th>fig32.gps SOURCE CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>GENERATE 50</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>ADVANCE 60</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>TERMINATE 1</td>
</tr>
</tbody>
</table>

---

S/C: OFF ABS CLOCK: 50.0000 REL CLOCK: 50.0000 TTG: 1

---

XACT: 1 CURBLK: 2 NEXTBLK: 3 CHAINS: FEC PC:

MARK-TIME: 50.0000 MOVE-TIME: 110.0000 PRIORITY: 0

---

ANNANDALE, VIRGINIA 22003, USA

Ready!
: step

XACT 1 POISED AT BLOCK 2. RELATIVE CLOCK: 50.0000
: step

XACT 1 PLACED ON FEC AT BLOCK 2. RELATIVE CLOCK: 50.0000
: 
Figure 3.3, Screen 3

<table>
<thead>
<tr>
<th>BLOCK</th>
<th>CURRENT</th>
<th>TOTAL</th>
<th>fig32.gps SOURCE CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>2</td>
<td>GENERATE 50</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>ADVANCE 60</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td></td>
<td>TERMINATE 1</td>
</tr>
</tbody>
</table>

S/C: OFF ABS CLOCK: 100.0000 REL CLOCK: 100.0000 TTG: 1

XACT: 2 CURBLK: 1 NEXTBK: 2 CHAINS: CEC PC:

MARK-TIME: 100.0000 MOVE-TIME: ------ PRIORITY: 0

: step

XACT 1 POISED AT BLOCK 2. RELATIVE CLOCK: 50.0000
: step

XACT 1 PLACED ON FEC AT BLOCK 2. RELATIVE CLOCK: 50.0000
: step

XACT 2 POISED AT BLOCK 2. RELATIVE CLOCK: 100.0000
: 
<table>
<thead>
<tr>
<th>BLOCK CURRENT</th>
<th>TOTAL</th>
<th>fig32, gps SOURCE CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>GENERATE 50</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>ADVERTISE 60</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>TERMINATE 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S/C: OFF</th>
<th>ABS CLOCK: 100.000</th>
<th>REL CLOCK: 100.000</th>
<th>TTG: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>XACT: 1</td>
<td>CURBLK: 2</td>
<td>NEXTBLK: 3</td>
<td>CHAIN: FFC</td>
</tr>
<tr>
<td>MARK-TIME: 100.000</td>
<td>MOVE-TIME: 160.000</td>
<td>PRIORITY: 0</td>
<td></td>
</tr>
</tbody>
</table>

- step
- XACT 1 PLACED ON FFC AT BLOCK 2. RELATIVE CLOCK: 50.0000
- step
- XACT 2 POISED AT BLOCK 2. RELATIVE CLOCK: 100.0000
- step
- XACT 2 PLACED ON FFC AT BLOCK 2. RELATIVE CLOCK: 100.0000
Figure 3.3, Screen 5

<table>
<thead>
<tr>
<th>BLOCK CURRENT</th>
<th>TOTAL</th>
<th>fig32.gps SOURCE CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>GENERATE 50</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>ADVANCE 60</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>TERMINATE 1</td>
</tr>
</tbody>
</table>

S/C: OFF  ABS CLOCK: 110.0000  REL CLOCK: 110.0000  TTG: 0

XACT: 1  CURBLK: 2  NEXTBLK: 3  CHAINS:  PC: 

MARK-TIME: 50.0000  MOVE-TIME: -----  PRIORITY: 0

: step

XACT 2 POISED AT BLOCK 2.  RELATIVE CLOCK: 100.0000
: step

XACT 2 PLACED ON FEC AT BLOCK 2.  RELATIVE CLOCK: 100.0000
: step

REQUESTING OUTPUT IN (CONTROL) STATEMENT NUMBER 7.  RELATIVE CLOCK: 110.0000

Note: no "destroy" message is issued when Xact 1 TERMINATEs at time 110.0; instead, just a "requesting output" message is issued (because the value of the model's Termination Counter has been reduced to 0)
At what simulated times do Xacts move into the TERMINATE Block, and what are their id numbers? Are there any occasions when one Xact "passes" another in the ADVANCE Block? (Xact B passes Xact A in an ADVANCE Block if Xact B goes into the ADVANCE Block after Xact A does, and then comes out of the ADVANCE Block before Xact A does.

Table E4

<table>
<thead>
<tr>
<th>Xact Id</th>
<th>Time into ADVANCE Block</th>
<th>Time into TERMINATE Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50</td>
<td>185.1</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>349.6</td>
</tr>
<tr>
<td>3</td>
<td>150</td>
<td>519.8</td>
</tr>
<tr>
<td>4</td>
<td>200</td>
<td>302.0</td>
</tr>
<tr>
<td>5</td>
<td>250</td>
<td>310.3</td>
</tr>
</tbody>
</table>
The Step Command Revisited
(Explicit Step Count Provided)

- A step count can be appended to a step command

   \[
   \text{step \ step\_count}
   \]

- The step\_count provides a count of the total number of Blocks that GPSS/H is to execute before returning control to the user; for example:

   \[
   \begin{align*}
   \text{step 2} \\
   \text{step 5} \\
   \text{step 1}
   \end{align*}
   \]

- The default value of the step\_count is 1

   \[
   \begin{align*}
   \text{step} \\
   \text{step 1}
   \end{align*}
   \]

- Recall that pressing F10 causes a step 1 command to be issued
The GENERATE Block's Offset-Interval and Limit-Count Operands

<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>
<tr>
<td>C</td>
<td>Offset Interval (deterministic time of first Xact arrival)</td>
<td>No Offset is in effect</td>
</tr>
<tr>
<td>D</td>
<td>Limit Count (maximum number of arrivals)</td>
<td>No Limit Count is in effect</td>
</tr>
</tbody>
</table>

Figure 3.4
An Example for Use of the GENERATE Block's Offset Interval

Start of the simulation

Time the first Xact moves from the GENERATE Block (dictated by the Offset Interval)

IAT: 25.0

Interarrival Time: 100.0

Time the second Xact moves from the GENERATE Block (based on the A/B Operands)

Time the third Xact moves from the GENERATE Block (based on the A/B Operands)

0.0 25.0 125.0 225.0 etc.

Simulated Time
SIMULATE
GENERATE 100, 50, 25
ADVANCE 50
TERMINATE 1
START 3
END

At what time do Xacts come out of the GENERATE Block? At what time does the simulation stop?

<table>
<thead>
<tr>
<th>Xact id</th>
<th>Time out of GENERATE Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25.0</td>
</tr>
<tr>
<td>2</td>
<td>124.9</td>
</tr>
<tr>
<td>3</td>
<td>248.9</td>
</tr>
</tbody>
</table>

The simulation stops at time 298.9 (= 248.9 + 50)
Exercise 2, Section 3.8
(Bug Clinic)

SIMULATE
GENERATE 100,,4
TERMINATE 1
START 5
END

What is the problem with this model? What do you think will happen if a simulation is performed with the model?

Answer
A Limit Count of 4 is used on the GENERATE Block, so it only produces four Xacts in total. After the fourth of these Xacts has come into the model and been destroyed, the model's Termination Counter hasn't yet been reduced to zero in value, so the simulation tries to continue. But nothing else has been scheduled for future simulated time. In response, GPSS/H prints out

Error Message 410: No Next Event in System
Transaction Time-Ties

• Whenever two or more Xacts move at the same simulated time, they are said to be involved in a time-tie

• Such Xacts will be moved one after the other at the simulated time in question

• In what order will the time-tied Xacts move?
  Which time-tied Xact will move first?
  Which time-tied Xact will move next?
  And so on...

• In the absence of other considerations about to be explained, time-tied Xacts move in the order in which the planning of their movement took place

• For example, suppose GPSS/H at some point in an ongoing simulation plans to move Xact 9 at time 35.0, and at some later point in the ongoing simulation plans to move Xact 4 at time 35.0 as well.

  Then at simulated time 35.0, Xact 9 will move before Xact 4 moves
A Model in Which Two Transactions Move at the Same Simulated Time

| GENERATE 25 | the first GENERATE Block |
| TERMINATE 1 | * |
| GENERATE 25 | the second GENERATE Block |
| TERMINATE 1 |

Figure 3.5
Which Xacts move in which order and at which simulated times in this model?

**Hint:** Remember that GENERATE Blocks are initialized in the top-down order of their appearance in a model, and that the successor Xact at a GENERATE Block is created and scheduled when its predecessor moves out of the GENERATE Block.

**Answer**
Time 25.0: First the Xact with id #1 moves, then the Xact with id #2 moves.
Time 50.0: First the Xact with id #3 moves, then the Xact with id #4 moves.
Time 75.0: First the Xact with id #5 moves, then the Xact with id #6 moves.

And so on, until the model's Termination Counter is decremented to zero.
### The GENERATE Block's Priority-Level Operand

**Diagram:**

- **GENERATE**
  - A, B, C, D, E

### Table: Default Values for Operands

<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>
<tr>
<td>C</td>
<td>Offset Interval (deterministic time of first Xact arrival)</td>
<td>No Offset is in effect</td>
</tr>
<tr>
<td>D</td>
<td>Limit Count (maximum number of arrivals)</td>
<td>No Limit Count is in effect</td>
</tr>
<tr>
<td>E</td>
<td>Xact Priority Level</td>
<td>0 (zero)</td>
</tr>
</tbody>
</table>

**Figure 3.6**

\[-2^{31} + 10 < \text{Priority Level} < +2^{31} - 10\]
Example of a Model in Which Priority Level is Used to Control the Order of Movement of Time-Tied Transactions

| GENERATE 25,,10 the first GENERATE Block |
| TERMINATE 1 |
| GENERATE 25,,10 the second GENERATE Block |
| TERMINATE 1 |

Figure 3.7

Which Xacts move in which order and at which simulated times in this model?

Answer

Time 25.0: First the Xact with id #2 moves, then the Xact with id #1 moves.
Time 50.0: First the Xact with id #4 moves, then the Xact with id #3 moves.
Time 75.0: First the Xact with id #6 moves, then the Xact with id #5 moves.

And so on, until the model's Termination Counter is decremented to zero.

(Compare this model with that in Figure 3.5.)
A repetition of Figure 2.2, with general Block details shown
The Statement Form of a Labeled Block:
An Example

<table>
<thead>
<tr>
<th>Label</th>
<th>Operation</th>
<th>Operands</th>
<th>Advance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CONVEYOR</td>
<td>(2 - 9)</td>
<td>11 - 20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(22 - 72)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>250</td>
</tr>
</tbody>
</table>

Figure 3.8
The TRANSFER Block in Unconditional Transfer Mode

```
< TRANSFER  (B) >
```

<table>
<thead>
<tr>
<th>Operand</th>
<th>Significance</th>
<th>Default Value or Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Must be omitted in unconditional-mode TRANSFER</td>
<td>The user must default</td>
</tr>
<tr>
<td>B</td>
<td>Pointer to a Block (copy of the Label attached to that Block)</td>
<td>Compile-time error</td>
</tr>
</tbody>
</table>

Figure 3.9
Views of a Simple Conveyor
(flow schematic for the simple conveyor model of Figure 3.10)

Side View

Top View

Objects Are Placed onto the Conveyor from Source 1 at Times 15, 45, 75, 105, 135, ...

Objects Are Placed onto the Conveyor from Source 2 at Times 30, 60, 90, 120, 150, ...

Travel Time: 250 Time Units

Instantaneous Removal of the Objects
Block Diagram for the Conveyor Model
(not given in ISU; the START Statement and the Model's Termination Counter are also pictured here)

START 25

The Model's Termination Counter
The Conveyor Model File and its Block-Diagram Equivalent

(Figure 3.10)

```
SIMULATE
*
GENERATE 30,,15 objects arrive from source 1
TRANSFER ,CONVEYOR go onto a conveyor
*
GENERATE 30 objects arrive from source 2
CONVEYOR ADVANCE 250 conveying time
TERMINATE 1 conveyed objects leave system
*
START 25 initialize the model's TC
* and the GENERATE Blocks;
* start the Xact-Movement Phase
* end of Model-File execution
END
```

```
GENERATE
30,,15

TRANSFER (CONVEYOR)

GENERATE
30 (CONVEYOR)

ADVANCE 250

TERMINATE 1
```
Example of a Model in Which Some Xacts Are Never Destroyed
(Exercise 3, Section 3.15)

<table>
<thead>
<tr>
<th>Command</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATE</td>
<td>10, 4</td>
</tr>
<tr>
<td>REPEAT</td>
<td></td>
</tr>
<tr>
<td>ADVANCE</td>
<td>40, 10</td>
</tr>
<tr>
<td>TRANSFER</td>
<td>, REPEAT</td>
</tr>
<tr>
<td>*</td>
<td></td>
</tr>
<tr>
<td>GENERATE</td>
<td>200</td>
</tr>
<tr>
<td>TERMINATE</td>
<td>1</td>
</tr>
</tbody>
</table>
On the Need to Distinguish Between 
Blocks (and Their Execution) 
and Control Statements (and Their Execution)

(Exercise 4, Section 3.16)

SIMULATE
*
GENERATE 75,40
TRANSFER , ZAPPO
*
GENERATE 35,5
ADVANCE 10
TERMINATE 1
*
ZAPPO START 5
*
END
The Steps Followed to Execute a Model File in Batch Mode
(Figure 2.19)

Start

(1) Model Compilation and Compiler Report

"SIMULATE"

Compile-Time Error(s), or no "SIMULATE"

Stop

(2) Control-Statement Execution
(Execution-Time Warning Messages might be issued)

"START"

"END"

Computer-Usage Report

Stop

(3) Transaction Movement
(Block-Statement Execution)
(Execution-Time Warning Messages might be issued)

Execution-Time Error

TC ≤ 0

(4) Postsimulation Report

..."SIMULATE"

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

Components of Postsimulation Report:
Clock time; Block Counts;
Facility Report (see Chapter 6);
Queue Report (see Chapter 9);
Storage Report (see Chapter 11);
Random-Numbers Report (see Chapter 14); etc.