

# Concept: Bank VI

## Goal

Eliminate a race condition in a VI.

## Description

You must identify and fix a problem with the server software in a bank. The bank server handles requests from many sources and must process the requests quickly. In order to increase its efficiency, the server uses two parallel loops—one to handle deposits to the account and another to handle withdrawals. The problem with the server is that some deposit or withdrawal requests are lost, thereby resulting in incorrect balances.

### Identify Race Condition

The files that you need to complete this exercise are here:

<NI eLearning>\LV Core 1\Race Condition\Exercise.

1. Open `Bank.vi` in the <Exercise> directory.
2. Run the VI.
3. Perform a deposit, a withdrawal, and a simultaneous transaction to familiarize yourself with the program.
4. Set the **Deposit Amount** to 20 and the **Withdrawal Amount** to 10.
5. Open the block diagram of the Bank VI while it is still running.
6. Arrange the block diagram of the Bank VI so that you can see it while operating the user interface.
7.  Enable execution highlighting on the block diagram by clicking **Highlight Execution**.
8. Click the **Simultaneous Transactions** button and watch the code as it executes. The balance should increase by 10.  
Notice that either the deposit or the withdrawal is lost, causing the balance to increase by 20 or decrease by 10.
9. Stop the VI.

You tracked the problem down to a race condition in a section of a code handling deposits and withdrawals for a single account. Although you can see the issue with execution highlighting enabled, during regular operation, the issue would occur sporadically.

## Remove Race Condition

Remove the race condition by protecting the critical section of code using a semaphore. In the VI, the critical sections of the code are those enclosed by a Sequence structure.

1. Save the Bank VI as `Bank with Semaphores.vi` in the <Exercise> directory.
2. Use semaphores to protect the critical sections of code, as shown in Figure 1.

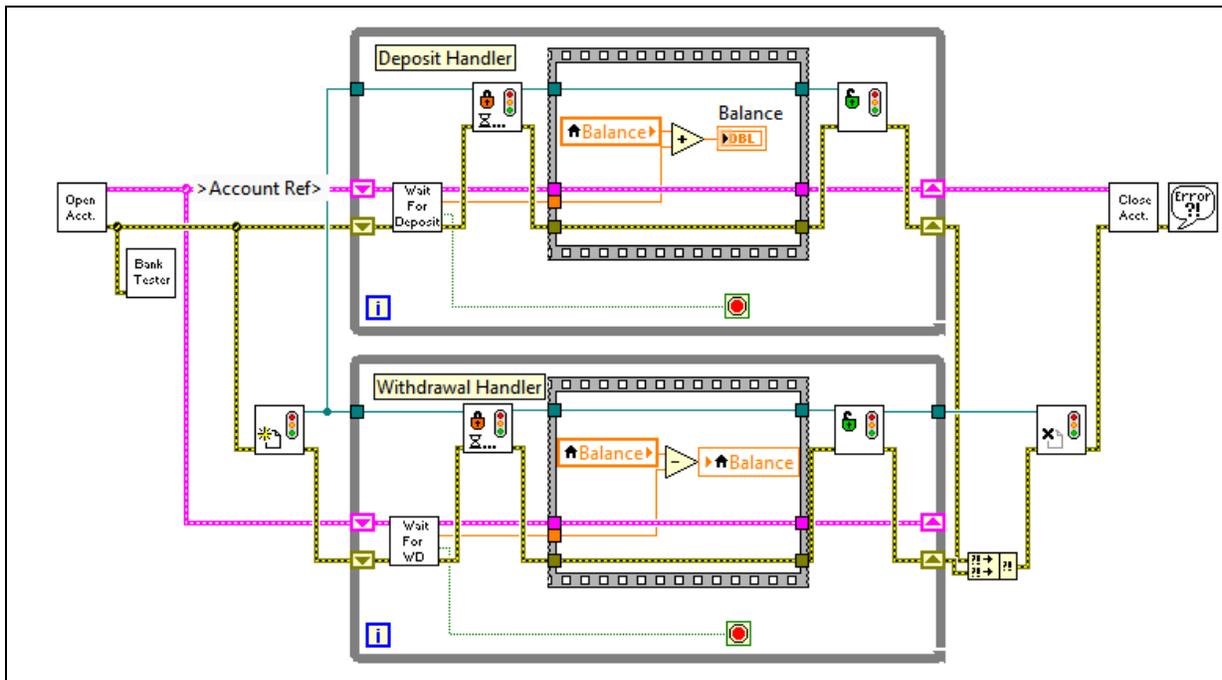


Figure 1. Bank with Semaphore



- Add an **Obtain Semaphore Reference VI** to the left of the While Loops.

- Wire the **Obtain Semaphore Reference VI** as shown in Figure 1.



- Add an **Acquire Semaphore VI** to the Deposit Handler loop, to the left of the Sequence structure.

- Add a second **Acquire Semaphore VI** to the Withdrawal Handler loop to the left of the Sequence structure.

- Wire the Acquire Semaphore VIs as shown in Figure 1.



- Add a **Release Semaphore VI** to the Deposit Handler loop, to the right of the Sequence structure.

Add a second **Release Semaphore VI** to the Withdrawal Handler loop, to the right of the Sequence structure.

Wire the Release Semaphore VIs as shown in Figure 1.



Add a **Release Semaphore Reference VI** to the right of the While Loops.

Wire the Release Semaphore Reference VI as shown in Figure 1. Notice that the Release Semaphore Reference VI requires only the reference to the semaphore.

3. Save the VI.

4. Repeat the steps detailed in the *Identify Race Conditions* section to test the modification to this VI.

5. Close the VI when you are finished.

## End of Exercise

# Notes

---