

# Applying the Java Fork-Join Framework's ManagedBlocker Interface

**Douglas C. Schmidt**

**[d.schmidt@vanderbilt.edu](mailto:d.schmidt@vanderbilt.edu)**

**[www.dre.vanderbilt.edu/~schmidt](http://www.dre.vanderbilt.edu/~schmidt)**

**Professor of Computer Science**

**Institute for Software  
Integrated Systems**

**Vanderbilt University  
Nashville, Tennessee, USA**



# Learning Objectives in this Part of the Lesson

- Understand how the common fork-join pool helps to maximize processor core utilization
- Recognize how the ManagedBlocker interface helps avoid starvation & improve performance
- Be able to apply the ManagedBlocker interface on blocking synchronizers & queues

```
class ManagedLocker implements
    ManagedBlocker {
    final ReentrantLock mLock;
    boolean mHasLock = false;

    ManagedLocker(ReentrantLock
        lock)
    { mLock = lock; }

    public boolean block() {
        if (!mHasLock)
            mLock.lock();
        return true;
    }
    ...
}
```

---

# Applying the Managed Blocker Interface

# Applying the ManagedBlocker Interface

- This example applies a ManagedBlocker on a ReentrantLock (from Java docs)

```
class ManagedLocker implements ManagedBlocker {  
    final ReentrantLock mLock;  
    boolean mHasLock = false;  
  
    ManagedLocker(ReentrantLock lock) { mLock = lock; }  
  
    public boolean isReleasable()  
    { return mHasLock || (mHasLock = mLock.tryLock()); }  
  
    public boolean block() {  
        if (!mHasLock) mLock.lock();  
        return true;  
    }  
}
```

*Handles a blocking  
synchronizer*

# Applying the ManagedBlocker Interface

- This example applies a ManagedBlocker on a ReentrantLock (from Java docs)

```
class ManagedLocker implements ManagedBlocker {
    final ReentrantLock mLock;
    boolean mHasLock = false;

    ManagedLocker(ReentrantLock lock) { mLock = lock; }

    public boolean isReleasable()
    { return mHasLock || (mHasLock = mLock.tryLock()); }

    public boolean block() {
        if (!mHasLock) mLock.lock();
        return true;
    }
}
```

*Implement Managed  
Blocker interface*

# Applying the ManagedBlocker Interface

- This example applies a ManagedBlocker on a ReentrantLock (from Java docs)

```
class ManagedLocker implements ManagedBlocker {  
    final ReentrantLock mLock;  
    boolean mHasLock = false;  
  
    ManagedLocker(ReentrantLock lock) { mLock = lock; }  
  
    public boolean isReleasable()  
    { return mHasLock || (mHasLock = mLock.tryLock()); }  
  
    public boolean block() {  
        if (!mHasLock) mLock.lock();  
        return true;  
    }  
}
```

*Constructor  
stores the lock*

# Applying the ManagedBlocker Interface

---

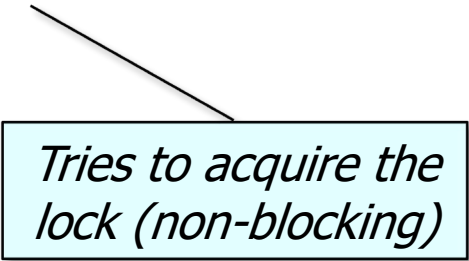
- This example applies a ManagedBlocker on a ReentrantLock (from Java docs)

```
class ManagedLocker implements ManagedBlocker {
    final ReentrantLock mLock;
    boolean mHasLock = false;

    ManagedLocker(ReentrantLock lock) { mLock = lock; }

    public boolean isReleasable()
    { return mHasLock || (mHasLock = mLock.tryLock()) ; }

    public boolean block() {
        if (!mHasLock) mLock.lock();
        return true;
    }
}
```



*Tries to acquire the lock (non-blocking)*

# Applying the ManagedBlocker Interface

---

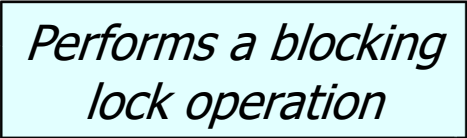
- This example applies a ManagedBlocker on a ReentrantLock (from Java docs)

```
class ManagedLocker implements ManagedBlocker {
    final ReentrantLock mLock;
    boolean mHasLock = false;

    ManagedLocker(ReentrantLock lock) { mLock = lock; }

    public boolean isReleasable()
    { return mHasLock || (mHasLock = mLock.tryLock()); }

    public boolean block() {
        if (!mHasLock) mLock.lock();
        return true;
    }
}
```



*Performs a blocking  
lock operation*



# Applying the ManagedBlocker Interface

---

- This example applies a ManagedBlocker on a ReentrantLock

```
void testManagedLocker() throws InterruptedException {
```

```
    ManagedLocker locker = new ManagedLocker(sLock);
```

```
    sLock.lock();
```

*Create a ManagedLocker to encapsulate ReentrantLock*

```
    ForkJoinPool.commonPool().execute(() -> {
```

```
        rethrowRunnable(() -> ForkJoinPool.managedBlock(locker));
```

```
        System.out.println("acquired the lock");
```

```
    }
```

```
    Thread.sleep(1000);
```

```
    sLock.unlock();
```

```
}
```

# Applying the ManagedBlocker Interface

---

- This example applies a ManagedBlocker on a ReentrantLock

```
void testManagedLocker() throws InterruptedException {
    ManagedLocker locker = new ManagedLocker(sLock);
    sLock.lock();
    ForkJoinPool.commonPool().execute(() -> {
        rethrowRunnable(() -> ForkJoinPool.managedBlock(locker));

        System.out.println("acquired the lock");
    })

    Thread.sleep(1000);
    sLock.unlock();
}
```

*Acquire the lock*

# Applying the ManagedBlocker Interface

---

- This example applies a ManagedBlocker on a ReentrantLock

```
void testManagedLocker() throws InterruptedException {  
    ManagedLocker locker = new ManagedLocker(sLock);  
    sLock.lock();
```

```
    ForkJoinPool.commonPool().execute(() -> {  
        rethrowRunnable(() -> ForkJoinPool.managedBlock(locker));
```

*Perform a blocking lock operation in a FJP worker thread*

```
        System.out.println("acquired the lock");  
    }
```

```
    Thread.sleep(1000);  
    sLock.unlock();  
}
```

# Applying the ManagedBlocker Interface

---

- This example applies a ManagedBlocker on a ReentrantLock

```
void testManagedLocker() throws InterruptedException {
    ManagedLocker locker = new ManagedLocker(sLock);
    sLock.lock();

    ForkJoinPool.commonPool().execute(() -> {
        rethrowRunnable(() -> ForkJoinPool.managedBlock(locker));

        System.out.println("acquired the lock");
    })

    Thread.sleep(1000);
    sLock.unlock();
}
```

*Block the calling thread for 1 second*

# Applying the ManagedBlocker Interface

- This example applies a ManagedBlocker on a ReentrantLock

```
void testManagedLocker() throws InterruptedException {
    ManagedLocker locker = new ManagedLocker(sLock);
    sLock.lock();

    ForkJoinPool.commonPool().execute(() -> {
        rethrowRunnable(() -> ForkJoinPool.managedBlock(locker));

        System.out.println("acquired the lock");
    })

    Thread.sleep(1000);
    sLock.unlock();
}
```

*Release the lock, which in turn unblocks the managed Block() call in the FJP worker thread*

# Applying the ManagedBlocker Interface

---

- This example applies a ManagedBlocker on a ReentrantLock

```
void testManagedLocker() throws InterruptedException {  
    ManagedLocker locker = new ManagedLocker(sLock);  
    sLock.lock();  
  
    ForkJoinPool.commonPool().execute(() -> {  
        rethrowRunnable(() -> ForkJoinPool.managedBlock(locker));  
  
        System.out.println("acquired the lock");  
    })  
  
    Thread.sleep(1000);  
    sLock.unlock();  
}
```

*FJP Thread prints this message  
only after the lock is released*

# Applying the ManagedBlocker Interface

- This example applies a ManagedBlocker on a BlockingQueue (from Java docs)

```
class QueueTaker<E> implements ManagedBlocker {
    final BlockingQueue<E> mQueue;
    volatile E mItem = null;

    QueueTaker(BlockingQueue<E> q) { mQueue = q; }

    public boolean isReleasable()
    { return mItem != null || (mItem = mQueue.poll()) != null; }

    public boolean block() throws InterruptedException
    { if (mItem == null) mItem = mQueue.take(); return true; }

    public E getItem() { return mItem; }
}
```

*Handles a blocking queue*

# Applying the ManagedBlocker Interface

- This example applies a ManagedBlocker on a BlockingQueue (from Java docs)


```
class QueueTaker<E> implements ManagedBlocker {
    final BlockingQueue<E> mQueue;
    volatile E mItem = null;

    QueueTaker(BlockingQueue<E> q) { mQueue = q; }

    public boolean isReleasable()
    { return mItem != null || (mItem = mQueue.poll()) != null; }

    public boolean block() throws InterruptedException
    { if (mItem == null) mItem = mQueue.take(); return true; }

    public E getItem() { return mItem; }
}
```



The blocking queue



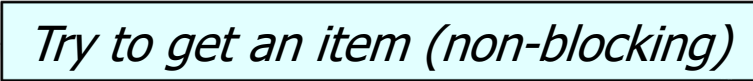
# Applying the ManagedBlocker Interface

---

- This example applies a ManagedBlocker on a BlockingQueue (from Java docs)

```
class QueueTaker<E> implements ManagedBlocker {
    final BlockingQueue<E> mQueue;
    volatile E mItem = null;

    QueueTaker(BlockingQueue<E> q) { mQueue = q; }

    public boolean isReleasable() 
    { return mItem != null || (mItem = mQueue.poll()) != null; }

    public boolean block() throws InterruptedException
    { if (mItem == null) mItem = mQueue.take(); return true; }

    public E getItem() { return mItem; }
}
```

# Applying the ManagedBlocker Interface

- This example applies a ManagedBlocker on a BlockingQueue (from Java docs)

```
class QueueTaker<E> implements ManagedBlocker {
    final BlockingQueue<E> mQueue;
    volatile E mItem = null;

    QueueTaker(BlockingQueue<E> q) { mQueue = q; }

    public boolean isReleasable()
    { return mItem != null || (mItem = mQueue.poll()) != null; }

    public boolean block() throws InterruptedException
    { if (mItem == null) mItem = mQueue.take(); return true; }

    public E getItem() { return mItem; }
}
```

*Block until an item is available*

# Applying the ManagedBlocker Interface

- This example applies a ManagedBlocker on a BlockingQueue (from Java docs)

```
class QueueTaker<E> implements ManagedBlocker {
    final BlockingQueue<E> mQueue;
    volatile E mItem = null;

    QueueTaker(BlockingQueue<E> q) { mQueue = q; }

    public boolean isReleasable()
    { return mItem != null || (mItem = mQueue.poll()) != null; }

    public boolean block() throws InterruptedException
    { if (mItem == null) mItem = mQueue.take(); return true; }

    public E getItem() { return mItem; }
}
```

*Call after pool.managedBlock() completes to get item*

# Applying the ManagedBlocker Interface

---

- This example applies a ManagedBlocker on a BlockingQueue

```
void testQueueTaker() throws InterruptedException {  
    QueueTaker<String> taker = new QueueTaker<>(sQueue);
```

*Create a QueueTaker to encapsulate a BlockingQueue*

```
ForkJoinPool.commonPool().execute(() -> {  
    rethrowRunnable(() -> ForkJoinPool.managedBlock(taker));  
  
    System.out.println("Took item = " + taker.getItem());  
}  
  
Thread.sleep(1000);  
  
sQueue.put("hello");  
}
```

# Applying the ManagedBlocker Interface

---

- This example applies a ManagedBlocker on a BlockingQueue

```
void testQueueTaker() throws InterruptedException {  
    QueueTaker<String> taker = new QueueTaker<>(sQueue);
```

```
    ForkJoinPool.commonPool().execute(() -> {  
        rethrowRunnable(() -> ForkJoinPool.managedBlock(taker));  
  
        System.out.println("Took item = " + taker.getItem());  
    }  
  
    Thread.sleep(1000);  
  
    sQueue.put("hello");  
}
```

*Block in a FJP worker thread  
until an item is available*

# Applying the ManagedBlocker Interface

---

- This example applies a ManagedBlocker on a BlockingQueue

```
void testQueueTaker() throws InterruptedException {  
    QueueTaker<String> taker = new QueueTaker<>(sQueue);
```

```
    ForkJoinPool.commonPool().execute(() -> {  
        rethrowRunnable(() -> ForkJoinPool.managedBlock(taker));
```

```
        System.out.println("Took item = " + taker.getItem());
```

```
    }
```

*Block the calling thread for 1 second*

```
    Thread.sleep(1000);
```

```
    sQueue.put("hello");
```

```
}
```

# Applying the ManagedBlocker Interface

- This example applies a ManagedBlocker on a BlockingQueue

```
void testQueueTaker() throws InterruptedException {
    QueueTaker<String> taker = new QueueTaker<>(sQueue);

    ForkJoinPool.commonPool().execute(() -> {
        rethrowRunnable(() -> ForkJoinPool.managedBlock(taker));

        System.out.println("Took item = " + taker.getItem());
    })

    Thread.sleep(1000);

    sQueue.put("hello");
}
```

*Put an item in the queue, which in turn unblocks the managedBlock() call in the FJP worker thread*

# Applying the ManagedBlocker Interface

---

- This example applies a ManagedBlocker on a BlockingQueue

```
void testQueueTaker() throws InterruptedException {
    QueueTaker<String> taker = new QueueTaker<>(sQueue);

    ForkJoinPool.commonPool().execute(() -> {
        rethrowRunnable(() -> ForkJoinPool.managedBlock(taker));

        System.out.println("Took item = " + taker.getItem());
    })

    Thread.sleep(1000);

    sQueue.put("hello");
}
```

*FJP Thread prints this message  
only after lock is released*



---

# End of Applying the Java Fork- Join Framework's Managed Blocker Interface