

# Another way to sort GUIDs: Java

 [devblogs.microsoft.com/oldnewthing/20190913-00](https://devblogs.microsoft.com/oldnewthing/20190913-00)

September 13, 2019



Raymond Chen

Some time ago, I surveyed a number of GUID-sorting algorithms. At the time, I noted, “Thankfully, it never occurred to anyone to try to sort GUID components as signed integers!”

How wrong I was.

For the purpose of sorting, Java treats each GUID as a pair of signed 64-bit integers in big-endian format.

This means that the following list of GUIDs is sorted according to Java:

{ 80000000-0000-0000-8000-000000000000 }

{ 80FFFFFF-FFFF-FFFF-7FFF-FFFFFFFFFFFFFF }

{ FFFFFFFF-FFFF-FFFF-7FFF-FFFFFFFFFFFFFF }

{ 00FFFFFF-FFFF-FFFF-7FFF-FFFFFFFFFFFFFF }

{ 7F00FFFF-FFFF-FFFF-7FFF-FFFFFFFFFFFFFF }

{ 7FFF00FF-FFFF-FFFF-7FFF-FFFFFFFFFFFFFF }

{ 7FFFFFF0-FFFF-FFFF-7FFF-FFFFFFFFFFFFFF }

{ 7FFFFFFF-00FF-FFFF-7FFF-FFFFFFFFFFFFFF }

{ 7FFFFFFF-FF00-FFFF-7FFF-FFFFFFFFFFFFFF }

{ 7FFFFFFF-FFFF-00FF-7FFF-FFFFFFFFFFFFFF }

{ 7FFFFFFF-FFFF-FF00-7FFF-FFFFFFFFFFFFFF }

{ 7FFFFFFF-FFFF-FFFF-80FF-FFFFFFFFFFFFFF }

{ 7FFFFFFF-FFFF-FFFF-FFFF-FFFFFFFFFFFFFF }

{ 7FFFFFFF-FFFF-FFFF-00FF-FFFFFFFFFFFFFF }

{ 7FFFFFFF-FFFF-FFFF-7F00-FFFFFFFFFFFFFF }

---

```
{7FFFFFFF-FFFF-FFFF-7FFF-00FFFFFFFF}
```

---

```
{7FFFFFFF-FFFF-FFFF-7FFF-FF00FFFFFFFF}
```

---

```
{7FFFFFFF-FFFF-FFFF-7FFF-FFFF00FFFFFF}
```

---

```
{7FFFFFFF-FFFF-FFFF-7FFF-FFFFFF00FFFF}
```

---

```
{7FFFFFFF-FFFF-FFFF-7FFF-FFFFFFFF00FF}
```

---

```
{7FFFFFFF-FFFF-FFFF-7FFF-FFFFFFFFFFFF00}
```

---

```
{7FFFFFFF-FFFF-FFFF-7FFF-FFFFFFFFFFFFFF}
```

The most significant bit of each 64-bit portion is a sign bit. This means that the smallest possible GUID is

```
{80000000-0000-0000-8000-000000000000}
```

and the largest possible GUID is

```
{7FFFFFFFFF-FFFF-FFFF-7FFF-FFFFFFFFFFFFFF}
```

In the highlighted red columns (corresponding to bits 0 and 64), the sort order is 89ABCDEF01234567. In the other columns, the sort order is 0123456789ABCDEF.

Raymond Chen

**Follow**

