

性能问题

垃圾收集器是周期性运行的，而且如果为变量分配的内存数量很可观，那么回收工作量也是相当大的。在这种情况下，确定垃圾收集的时间间隔是一个非常重要的问题。说到垃圾收集器多长时间运行一次，不禁让人联想到 IE 因此而声名狼藉的性能问题。IE 的垃圾收集器是根据内存分配量运行的，具体一点说就是 256 个变量、4096 个对象（或数组）字面量和数组元素（slot）或者 64KB 的字符串。达到上述任何一个临界值，垃圾收集器就会运行。这种实现方式的问题在于，如果一个脚本中包含那么多变量，那么该脚本很可能在其生命周期中一直保有那么多的变量。而这样一来，垃圾收集器就不得不频繁地运行。结果，由此引发的严重性能问题促使 IE7 重写了其垃圾收集例程。

随着 IE7 的发布，其 JavaScript 引擎的垃圾收集例程改变了工作方式：触发垃圾收集的变量分配、字面量和（或）数组元素的临界值被调整为动态修正。IE7 中的各项临界值在初始时与 IE6 相等。如果垃圾收集例程回收的内存分配量低于 15%，则变量、字面量和（或）数组元素的临界值就会加倍。如果例程回收了 85% 的内存分配量，则将各种临界值重置回默认值。这一看似简单的调整，极大地提升了 IE 在运行包含大量 JavaScript 的页面时的性能。



事实上，在有的浏览器中可以触发垃圾收集过程，但我们不建议读者这样做。在 IE 中，调用 `window.CollectGarbage()` 方法会立即执行垃圾收集。在 Opera 7 及更高版本中，调用 `window.opera.collect()` 也会启动垃圾收集例程。