A .NET效率陷阱之Attributes PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/286/2021_2022_A_NET_E6_95_88_E7_c67_286799.htm 众所周知，在编写WebCustomControl时，继承于WebControl基类的Attributes以及其Attributes.CssStyle属性是十分常用和重要的。但就是这两个重要的属性，如果开发中使用不当却会带来莫名其妙的效率问题。 由于html的灵活性和不完备性，导致了WebControl基类没有完整的表现html元素所提供和支持的所有标签属性和CSS属性（当然由于不同browser的兼容问题，要提供完备的属性是不可能的）。又由于很多html标签属性和CSS属性都是很生僻的，很少或极少被使用，如果要完备的支持，反而会成为WebControl的负担。所以Attributes和Attributes.CssStyle这两个属性很好的解决了这个问题，当然这两个属性除了支持应有的html标签属性和CSS属性外，还支持任何合法的自定义key/value对。这里要讨论的问题就来之这个对自定义key/value对的支持上。 Attributes属性的类型是一个AttributeCollection，本来很自然的一个东西，可是不知道怎么搞得，AttributeCollection的构造函数却需要一个StateBag参数： public AttributeCollection(StateBag bag){ this._bag = bag.} 这样的结果就是，Attributes和Attributes.CssStyle可能会被保存在ViewState中，事实上ASP.NET默认确实会保存其中的内容到ViewState中。 这种设计真的是让人觉得莫名其妙，在大家对ViewState效率问题的讨论中，觉得ViewState确实是鸡肋，用来保持一些服务器状态和数据让大家觉得方便也就算了。可是居然把和UI相关的内容都一股脑存到ViewState里，真的

是疯狂。 下面是使用Attributes定义了一些自定义内容后的ViewState的情形： // AnalysisReport自定义控件上定义了一些自定的内容 Attributes和Attributes.CssStyle被自动保存到ViewState中后，除了ViewState体积急增后，PostBack时Load ViewState的负担也同时增大了。上面这个事例中的页面PostBack的LoadState代价，如下图： 实际上我在编写控件时，从来没有想过要保持Attributes和Attributes.CssStyle，也没有想过要再次使用其中的数据。而且这个默认保存到ViewState的行为居然不能定制（至少我还没有发现），后来想到在ASP.NET页面生存期中，SaveState结束在PreRender中，所以在Render事件中使用Attributes和Attributes.CssStyle的就不会保存到ViewState中去。 修改代码：protected override void OnPreRender(EventArgs e){ this.Attributes["abc"] = "123". this.Attributes.CssStyle["abc-style"] = "123-style". base.OnPreRender(e).} 为如下形式：protected override void Render(HtmlTextWriter output){ this.Attributes["abc"] = "123". this.Attributes.CssStyle["abc-style"] = "123-style". output.Write(Text).} 就不会再将Attributes和Attributes.CssStyle保存到ViewState中了，上面那个AnalysisReport按上面的示例修改后，绑定同样数据的运行效果为： LoadState的代价也大大降低，其开销为： 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com