

# 在 ZS 中使用多媒体机制 CompleteBuffer

## 目录

性能 .....	1
安全 .....	1
解决安全问题.....	1
使用和陷阱.....	2

## 性能

CompleteBuffer 的性能是 ZS 中性能最高的命令类型，它的性能接近与裸 IO 的收发，它不会对 Buffer 做多余的打包，压缩，编码等工作

## 安全

CompleteBuffer 因为需要在服务器预留内存中，所以，我们处理这类命令时，需要限制它，不能让它过大，否则服务器将会受到安全威胁

## 解决安全问题

在我们的 TcommunicationFramework 基类中有个 MaxCompleteBufferSize 参量，它限制了客户端能收发的 CompleteBuffer 大小，我们根据业务场景，限制好它的尺寸即可

# 使用和陷阱

在服务器注册一条 **CompleteBuffer** 的接收命令

```
RegisterCompleteBuffer(xx).OnExecute := procedure(Sender: TPeerClient; InData: PByte; DataSize: NativeInt)
```

```
Begin
```

```
    //触发该事件时，InData 为 Buff 指针，我们必须在这里把给 Buff 处理工作做完，该事件结束时，buff 会被自动释放  
end;
```

- 1, **CompleteBuffer** 事件处于性能和回收机理的设计，是在线程中以同步方式触发的，我们不能在 **OnExecute** 事件中再去用 **WaitSend** 这类指令，因为后台线程正在等待你的事件结束，否则会造成卡死情况。我们也不能在 **OnExecute** 事件中使用异步，因为 **OnExecute** 事件一旦结束，**InData** 的内存控件就会被立即释放
- 2, **CompleteBuffer** 的大小受接收端的 **MaxCompleteBufferSize** 参量限制，如果发生发送断线，那么就是被限制了，检查接收端的 **MaxCompleteBufferSize** 参量即可
- 3, 使用 **CompleteBuffer** 建议把限制设置设置为 8M 以下，否则并发多了，服务器将会受到威胁

By qq 600585

2018-1-24