

I have put together a little client/server program to try out MiscStrings in a Distributed Objects environment. The server seems to leak memory on every transaction, though.

Here's the protocol definition for my server:

```
- (int)checkInString:(in MiscString *)s;
```

Here's the implementation:

```
- (int)checkInString:(MiscString *)s  
{  
    const char *cp = [s stringValue];  
    return strlen (cp);  
}
```

The problem is that every time the client sends a message to the server, the server leaks N bytes, when N is the length of the MiscString (including the terminating '\0') for the *previous* message.

MallocDebug.app shows the call chain to the leak as:

```
default 0x0625d120 4  
    _NXDecodeChars,  
    idecodeData,  
    -[NXPortPortal decodeData:ofType:],  
    -[NXMethodSignature decodeMethodRetFrom:withargs:atAddr:],  
    -[NXProxy forward:],  
    _objc_msgForward,  
    -[CommServer checkInString:],  
    objc_msgSendv
```

So how can this leak be avoided?