如果JID的resource缺省，则登陆后XMPP服务器会自动分配一个resource，如tigase-10.由于XMPP是支持多点登陆的，所以不同客户端用同一个JID登陆后，服务器会为每一个客户端分配一个不同的resource以区分他们。比如A客户端的resource是tigase-10，B客户端是tigase-11，这时A的客户端的完整jid是[borissun@iteye.com](mailto:borissun@iteye.com)/tigase-10,B客户端的jid是[borissun@iteye.com](mailto:borissun@iteye.com)/tigase-11。这样如果C客户端向[borissun@iteye.com](mailto:borissun@iteye.com)/tigase-10发消息，则A会收到，向[borissun@iteye.com](mailto:borissun@iteye.com)/tigase-11则B会收到，但是向[borissun@iteye.com](mailto:borissun@iteye.com)发消息，服务器不知道到底是给谁发了，因此一定有一个客户端会丢失消息。

  如果想将XMPP做成单点登陆，则jid的resource就要一致，比如[borissun@iteye.com](mailto:borissun@iteye.com)/boris .XMPP 的server对于resource一致的情况有两种处理的可能：

If there is already an active resource of the same name, the server MUST either (1) terminate the active resource and allow the newly-requested session, or (2) disallow the newly-requested session and maintain the active resource. Which of these the server does is up to the implementation, although it is RECOMMENDED to implement case #1. In case #1, the server SHOULD send a <conflict/> stream error to the active resource, terminate the XML stream and underlying TCP connection for the active resource, and return a IQ stanza of type "result" (indicating success) to the newly-requested session. In case #2, the server SHOULD send a <conflict/> stanza error to the newly-requested session but maintain the XML stream for that connection so that the newly-requested session has an opportunity to negotiate a non-conflicting resource identifier before sending another request for session establishment.

简单的说也就是如果resource一致，第一个办法是终结活动着的那个resource，然后让最新请求的会话留下来，也就是A先登陆，B再登陆，A会掉线。第二个办法是拒绝最新的请求会话，保持第一个，也就是A先登陆，B再登陆，B登不上去。   
  
我的实例中的server是采用的第一种方法，不管哪一种方法，要掉线的客户端会收到server发的一个error： 

**Xml代码**

1. <stream:error xmlns:stream="http://etherx.jabber.org/streams">
2. <conflict xmlns="urn:ietf:params:xml:ns:xmpp-streams"></conflict>
3. </stream:error>

复制代码

conflict代表了登陆冲突，因此只要收到了这个消息，就可以确定有其他客户端登陆了同样的jid，因此我们只用处理这个error就好。   
在OC的XMPPFramework中可以这样处理： 

**C代码**

1. - (void)xmppStream:(XMPPStream \*)sender didReceiveError:(id)error
2. {
3. HWLOGI(@"didReceiveError:%@",error);
4. DDXMLNode \*errorNode = (DDXMLNode \*)error;
5. //遍历错误节点
6. for(DDXMLNode \*node in [errorNode children])
7. {
8. //若错误节点有【冲突】
9. if([[node name] isEqualToString:@"conflict"])
10. {
11. //停止轮训检查链接状态
12. [\_timer invalidate];
13. NSString \*message = [NSString stringWithFormat:@"%@さんが別の端末でログインしたため、自動的にログアウトしました。",[[[SharedAppDelegate myInfo] myCardDetial]fullNameWithSpace]];
14. //弹出登陆冲突,点击OK后logout
15. UIAlertView \*alert = [[UIAlertView alloc]initWithTitle:nil message:message delegate:self cancelButtonTitle:@"OK" otherButtonTitles:nil, nil];
16. alert.tag = 9999;
17. alert.delegate = SharedAppDelegate;
18. [alert show];
19. [alert release];
20. }
21. }
22. }

复制代码