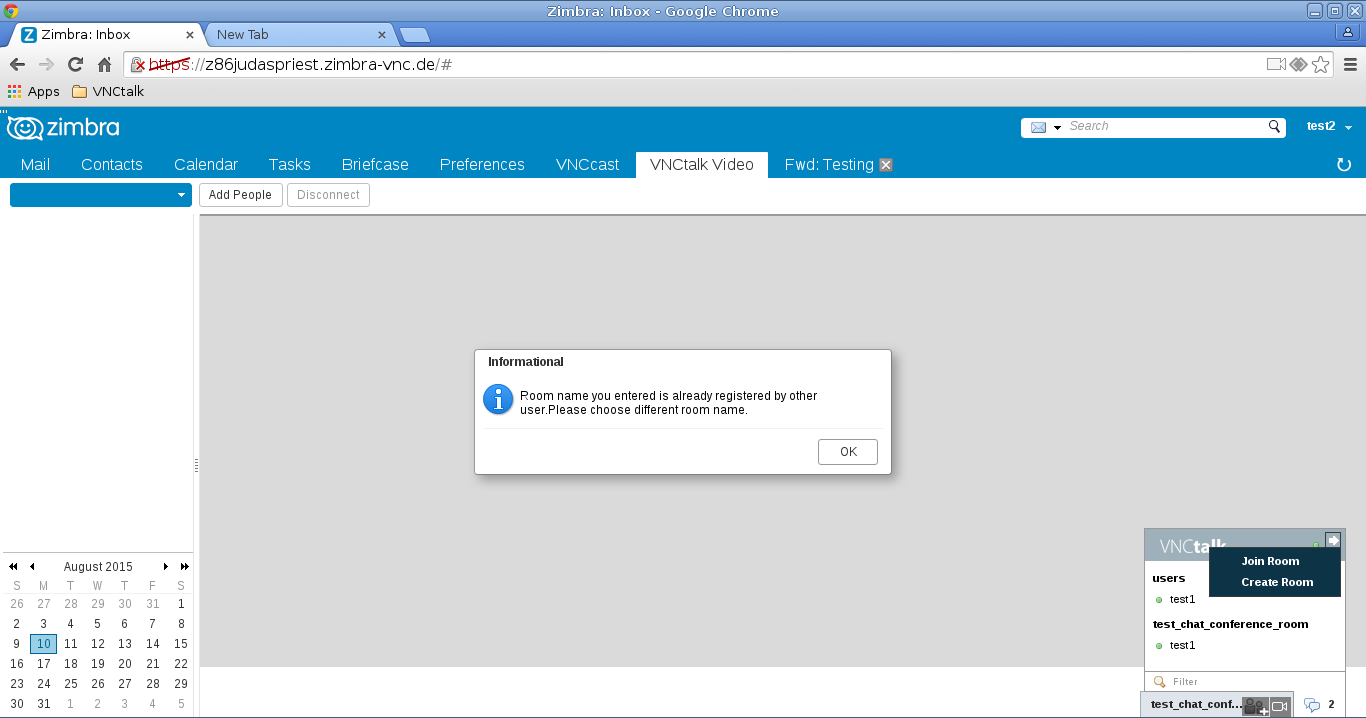
**#21524** - <https://redmine.vnc.biz/issues/21524>

1. **Description**:

Scenario 1:

* Alice and Bob are having a working video conference
* Bob creates a chatroom "abcd" using the "Roster->Create room" procedure
* Alice and Bob are still having a videoconference
* Alice decides to create a chatroom (also called) "abcd" using the "Roster->Create room" procedure
* As expected an info/error message is showed to alice, that the chatroom already exists
* But unexpectedly Alice is kicked out from the videoconference! Somehow when the message displaying that the room "abcd" already exists, closes Alice videoconference.
* This unexpected closing of the videoconference also leaves the videoconference chatroom window in the wrong state: displaying the icons only showed during the videoconference.



Scenario 2:

1) Alice and Bob are in a videoconference

2) Bob creates a new chatroom "abcd" while staying in the videoconference

3) wait 30 seconds (not necessary, but helps to understand)

4) Alice creates a new chatroom "abcd" while staying in the videoconference

4.1) A pop up signaling a chatroom-name-collission appears.

4.2) At the same time the videoconference between Alice and Bob is closed on Alice zimbra! This is the error.

1. **Analytic**
   1. **Root cause**:

* Scenario 2: Alice creates a new chatroom "abcd" the same name with which Bob created, so Alice get iqXml error with roomid = "abcd", and then video frame is removed.
  1. **Solutions**:
* Scenario 1 : check current conference and compare with new conference name before user send stanza to create new conference. If it is true, popup inform "Room name you entered is already registered. Please choose different room name.". Click OK button to retry input a difference room name.
* Scenario 2: When receive iqXml error with roomid="abcd", we need to check roomid of error stanza with roomId of video conference before remove video frame.

1. **Implementation**
   1. **Code:**

* Modified self.createGroupchatPrompt() function:

|  |
| --- |
| self.createGroupchatPrompt = function() { ...  if (biz\_zcs\_vnc\_talk\_zimlet\_HandlerObject.currentRoomId == biz\_zcs\_vnc\_talk\_zimlet\_this.checkValidSubject(join\_this)) {  var dialog = appCtxt.getMsgDialog(); // get a simple message dialog  dialog.reset(); // reset the dialog  dialog.setMessage("Room name you entered is already registered. Please choose different room name.",  DwtMessageDialog.INFO\_STYLE); // set the message "info" style  dialog.setButtonListener(DwtDialog.OK\_BUTTON,new AjxListener(this,function(){  dialog.popdown();  JappixMini.createGroupchatPrompt();  }));  dialog.popup();   }else{  JappixMini.acceptDefaultRoomConfigMini(chat\_room);  } ... }; |

* Modified self.handleAcceptDefault(iq) function:

|  |
| --- |
| self.handleAcceptDefault = function(iq){ ...  if(iqXML.find("error").size()>0){  var roomId=iqXML[0].getAttribute("from");  msg=biz\_zcs\_vnc\_talk\_zimlet.vnc\_talk\_existing\_room\_name;  JappixMini.presence('unavailable', '', '', '', roomid + '/' + MINI\_USER);  if(appCtxt){  var dialog = appCtxt.getMsgDialog(); // get a simple message dialog  dialog.reset(); // reset the dialog  dialog.setMessage(msg, DwtMessageDialog.INFO\_STYLE); // set the message "info" style  dialog.popup(); // display the dialog   // remove video frame when error  var currentRoom = biz\_zcs\_vnc\_talk\_zimlet\_HandlerObject.currentRoomId + "@conference." + MINI\_DOMAIN;  if (roomId == currentRoom) {  biz\_zcs\_vnc\_talk\_zimlet\_HandlerObject.currentRoomId = null;  var iframes = document.getElementsByName("vncTalkAppTabiFrame");  biz\_zcs\_vnc\_talk\_zimlet\_this.chatDisconnect.setEnabled(false);  for (var i = 0; i < iframes.length; i++) {  iframes[i].contentWindow.postMessage({key:"disconnect",value:""},"\*");  iframes[i].src = "about:blank";  }  };  }  }else{  ... } |