**#19231** [- https://redmine.vnc.biz/issues/19231](https://redmine.vnc.biz/issues/19231)

1. **Description**:

|  |
| --- |
| Validate and check user input data in the SIP config fields for the VNCtalk zimlet preferences. |

1. **Analytic**
   1. **Solutions**:

|  |
| --- |
| SIP-DisplayName : do not be blank or include invalid character  SIP-Server/Registar : do not be blank and must be domain, ipv4 or ipv6  SIP-User : do not be blank  SIP-Password : do not be blank  SIP-Other : optional, not check |

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

Edit biz\_...\_zimlet.js

Edit code for check if VNC\_TALK\_SOFTPHONE\_FLAG, VNC\_TALK\_SIP\_DISPLAY\_NAME, VNC\_TALK\_SIP\_SERVER\_NAME, VNC\_TALK\_SIP\_USER, VNC\_TALK\_SIP\_PASSWORD, VNC\_TALK\_SIP\_OTHER have old value

|  |
| --- |
| document.getElementById("vnc\_talk\_softphone\_flag").checked = (typeof VNC\_TALK\_SOFTPHONE\_FLAG != "undefined") ? VNC\_TALK\_SOFTPHONE\_FLAG : 0; document.getElementById("vnc\_talk\_sip\_display\_name").value = (typeof VNC\_TALK\_SIP\_DISPLAY\_NAME != "undefined") ? VNC\_TALK\_SIP\_DISPLAY\_NAME : ""; document.getElementById("vnc\_talk\_sip\_server\_name").value = (typeof VNC\_TALK\_SIP\_SERVER\_NAME != "undefined") ? VNC\_TALK\_SIP\_SERVER\_NAME : ""; document.getElementById("vnc\_talk\_sip\_user").value = (typeof VNC\_TALK\_SIP\_USER != "undefined") ? VNC\_TALK\_SIP\_USER : ""; document.getElementById("vnc\_talk\_sip\_password").value = (typeof VNC\_TALK\_SIP\_PASSWORD != "undefined") ? VNC\_TALK\_SIP\_PASSWORD : ""; document.getElementById("vnc\_talk\_sip\_other").value = (typeof VNC\_TALK\_SIP\_OTHER != "undefined") ? VNC\_TALK\_SIP\_OTHER : ""; |

Define parameters '\_nameRegex' and '\_sipDomainRegex':

|  |
| --- |
| var \_nameRegex = RegExp(/[`~@!#$%\^&\*+=\[\]\\';,/{}|\\":<>\?\\]/g);  var \_sipDomainRegex = RegExp(/((^\s\*((([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5]))\s\*$)|(^\s\*((([0-9A-Fa-f]{1,4}:){7}([0-9A-Fa-f]{1,4}|:))|(([0-9A-Fa-f]{1,4}:){6}(:[0-9A-Fa-f]{1,4}|((25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)(\.(25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)){3})|:))|(([0-9A-Fa-f]{1,4}:){5}(((:[0-9A-Fa-f]{1,4}){1,2})|:((25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)(\.(25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)){3})|:))|(([0-9A-Fa-f]{1,4}:){4}(((:[0-9A-Fa-f]{1,4}){1,3})|((:[0-9A-Fa-f]{1,4})?:((25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)(\.(25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)){3}))|:))|(([0-9A-Fa-f]{1,4}:){3}(((:[0-9A-Fa-f]{1,4}){1,4})|((:[0-9A-Fa-f]{1,4}){0,2}:((25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)(\.(25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)){3}))|:))|(([0-9A-Fa-f]{1,4}:){2}(((:[0-9A-Fa-f]{1,4}){1,5})|((:[0-9A-Fa-f]{1,4}){0,3}:((25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)(\.(25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)){3}))|:))|(([0-9A-Fa-f]{1,4}:){1}(((:[0-9A-Fa-f]{1,4}){1,6})|((:[0-9A-Fa-f]{1,4}){0,4}:((25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)(\.(25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)){3}))|:))|(:(((:[0-9A-Fa-f]{1,4}){1,7})|((:[0-9A-Fa-f]{1,4}){0,5}:((25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)(\.(25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)){3}))|:)))(%.+)?\s\*$))|(^\s\*((?=.{1,255}$)(?=.\*[A-Za-z].\*)[0-9A-Za-z](?:(?:[0-9A-Za-z]|\b-){0,61}[0-9A-Za-z])?(?:\.[0-9A-Za-z](?:(?:[0-9A-Za-z]|\b-){0,61}[0-9A-Za-z])?)\*)\s\*$)/); |

Validate and check user input data in the SIP config fields if they empty or have wrong format:

|  |
| --- |
| //validate SIP display name if(blankRegx.test(sipDisplayName)){  var msg = appCtxt.getMsgDialog()  msg.setMessage(biz\_zcs\_vnc\_talk\_zimlet\_this.getMessage("vnc\_talk\_sip\_blank\_display\_name"),DwtMessageDialog.WARNING\_STYLE);  msg.popup();  return; } if(\_nameRegex.test(sipDisplayName)){  var msg = appCtxt.getMsgDialog()  msg.setMessage(biz\_zcs\_vnc\_talk\_zimlet\_this.getMessage("vnc\_talk\_sip\_invalid\_display\_name"),DwtMessageDialog.WARNING\_STYLE);  msg.popup();  return; }  //validate SIP server name if(blankRegx.test(sipServerName)){  var msg = appCtxt.getMsgDialog()  msg.setMessage(biz\_zcs\_vnc\_talk\_zimlet\_this.getMessage("vnc\_talk\_sip\_blank\_server\_name"),DwtMessageDialog.WARNING\_STYLE);  msg.popup();  return; } if(\_sipDomainRegex.test(sipServerName)){  var msg = appCtxt.getMsgDialog()  msg.setMessage(biz\_zcs\_vnc\_talk\_zimlet\_this.getMessage("vnc\_talk\_sip\_invalid\_server\_name"),DwtMessageDialog.WARNING\_STYLE);  msg.popup();  return; }  //validate SIP user if(blankRegx.test(sipUser)){  var msg = appCtxt.getMsgDialog()  msg.setMessage(biz\_zcs\_vnc\_talk\_zimlet\_this.getMessage("vnc\_talk\_sip\_blank\_user"),DwtMessageDialog.WARNING\_STYLE);  msg.popup();  return; }  //validate SIP password  if(blankRegx.test(sipPassword)){  var msg = appCtxt.getMsgDialog()  msg.setMessage(biz\_zcs\_vnc\_talk\_zimlet\_this.getMessage("vnc\_talk\_sip\_blank\_password"),DwtMessageDialog.WARNING\_STYLE);  msg.popup();  return; } |

Define new messages:

By English (biz\_zcs\_vnc\_talk\_zimlet.properties):

|  |
| --- |
| vnc\_talk\_sip\_invalid\_display\_name = Invalid SIP DisplayName. vnc\_talk\_sip\_blank\_display\_name = SIP DisplayName can not be blank. vnc\_talk\_sip\_invalid\_server\_name = Invalid SIP Server/Registar. vnc\_talk\_sip\_blank\_server\_name = Server/Registar can not be blank. vnc\_talk\_sip\_blank\_user = SIP user can not be blank. vnc\_talk\_sip\_blank\_password = SIP password can not be blank. |

By German (biz\_zcs\_vnc\_talk\_zimlet\_de.properties):

|  |
| --- |
| vnc\_talk\_sip\_invalid\_display\_name = Ungültiger SIP Anzeigename. vnc\_talk\_sip\_blank\_display\_name = Der SIP Anzeigename darf nicht leer sein. vnc\_talk\_sip\_invalid\_server\_name = Ungültiger SIP Server/Registar. vnc\_talk\_sip\_blank\_server\_name = Der SIP Server/Register darf nicht leer sein. vnc\_talk\_sip\_blank\_user = Der SIP Benutzername darf nicht leer sein. vnc\_talk\_sip\_blank\_password = Das SIP Passwort darf nicht leer sein. |

Committed code to ZCS Zimlet VNCTalk on Release\_2.4.0 <https://redmine.vnc.biz/projects/vnc-zimlet-chat/repository/revisions/d52a180287c98cb35ba84648d5f3b98cfb21d53f>