

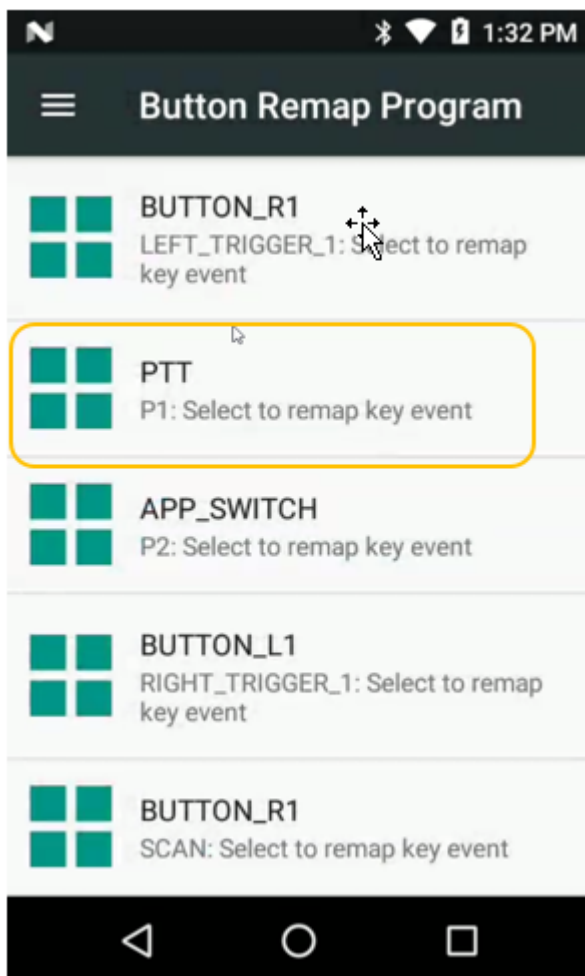
In this sample we wish to map the [P1] physical key on a Zebra MC33 terminal.

Working with TELNET\_VT emulation:

First, we have to open *Settings - Key Programmer* to assign a non-used Android function to [P1] key.

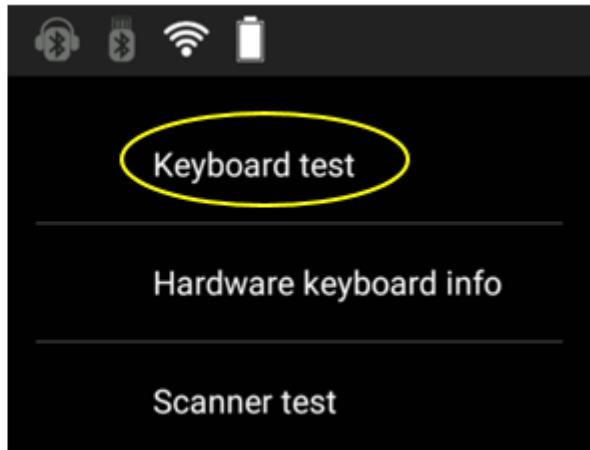
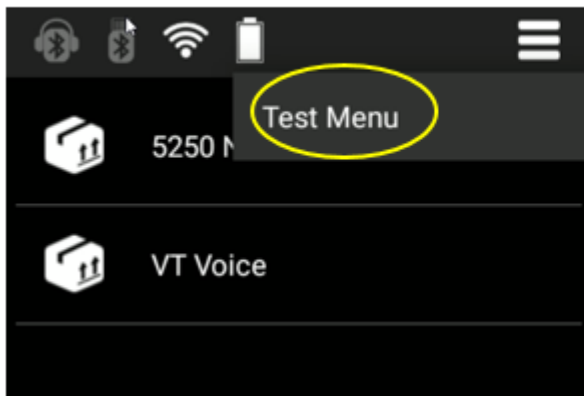
That is because we are using a Zebra device for the example; in other devices, find out the corresponding menu option to remap keys.

Select the [P1] button to remap and touch a function you don't use on Android, to map to the button (in our example, PPT).



Open TellNext app, select a Profile and open *Test Menu - Keyboard* test at the right-top of the screen

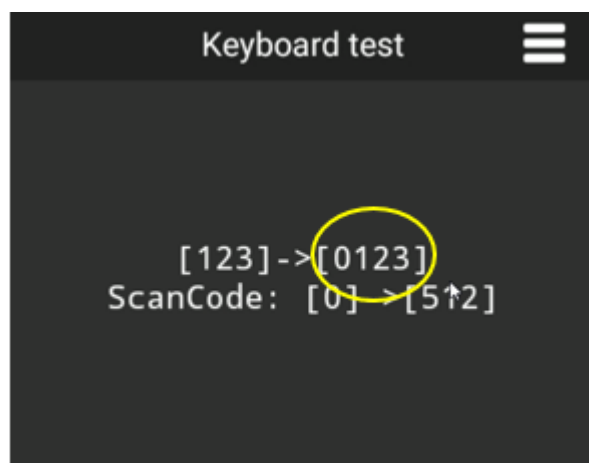
1



Change the process you want to test to the corresponding VT

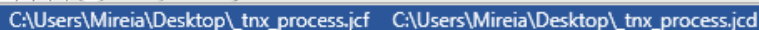


Press the P1 key and take note the scan code displayed (0123)



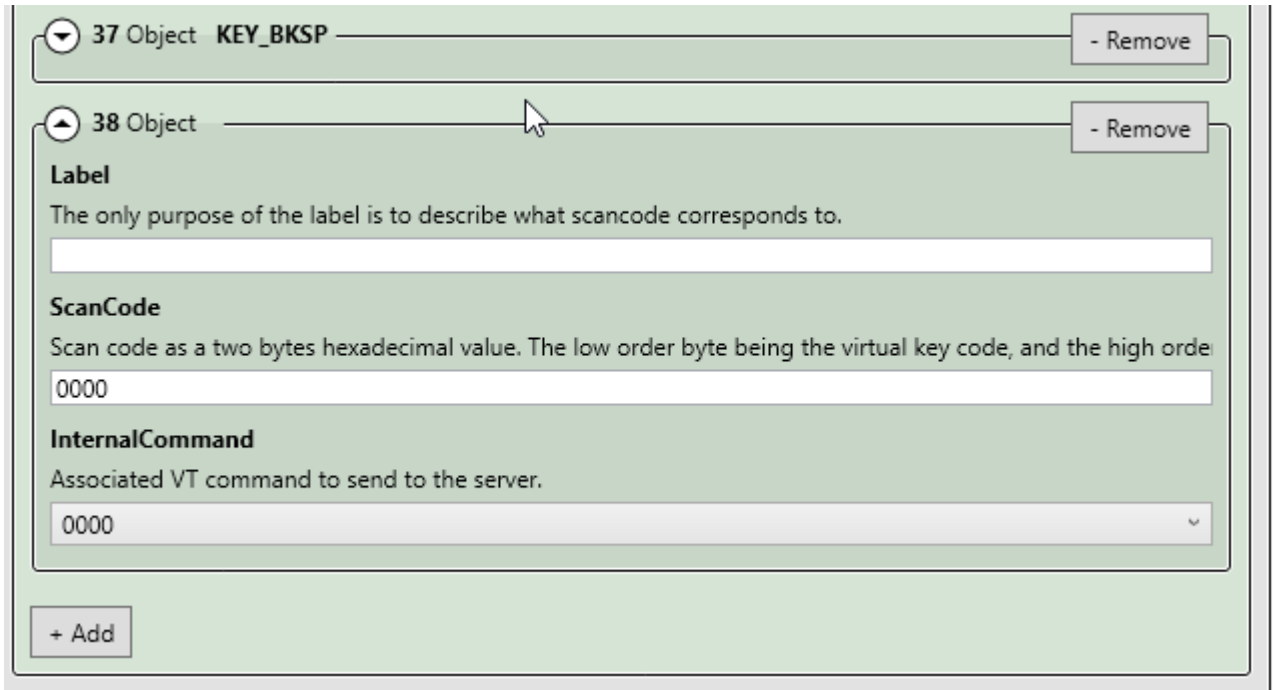
Exit TellNext in the terminal until get the Desktop screen.

Edit, with VisualJsonEditor, the [\\_tnx\\_process.jcf](#) file associated with the VT emulation Process you want to use (in our case, *VT Voice*)



You can see here all the assigned [internal commands in for the VT](#) emulation.

Goto Keyboard.SccToCmdMappings.TnVT and add a new value to the array. This new value will be composed of the next parameters:



37 Object **KEY\_BKSP** - Remove

38 Object - Remove

**Label**  
The only purpose of the label is to describe what scancode corresponds to.

**ScanCode**  
Scan code as a two bytes hexadecimal value. The low order byte being the virtual key code, and the high order

0000

**InternalCommand**  
Associated VT command to send to the server.

0000

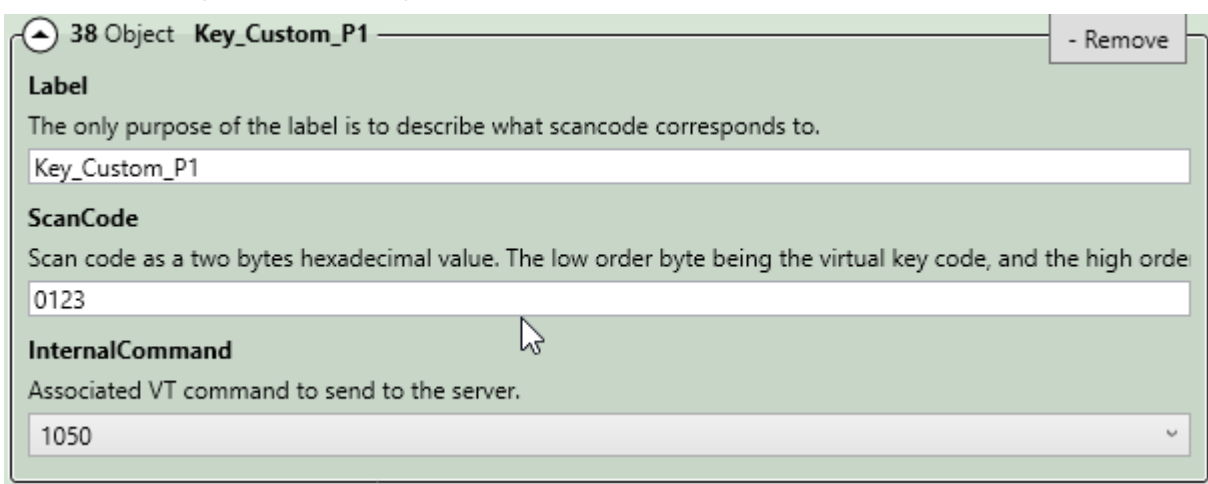
+ Add

Label: in this sample "Key\_Vustom\_P1"

ScanCode: write down the scan code associated to the [P1] key. The one displayed at Keyboard Test, in our case, 0123

InternalCommand: dropdown the enum value and select a non-selected before value.  
E.g. 1050

That is, to complete our example:



38 Object **Key\_Custom\_P1** - Remove

**Label**  
The only purpose of the label is to describe what scancode corresponds to.

Key\_Custom\_P1

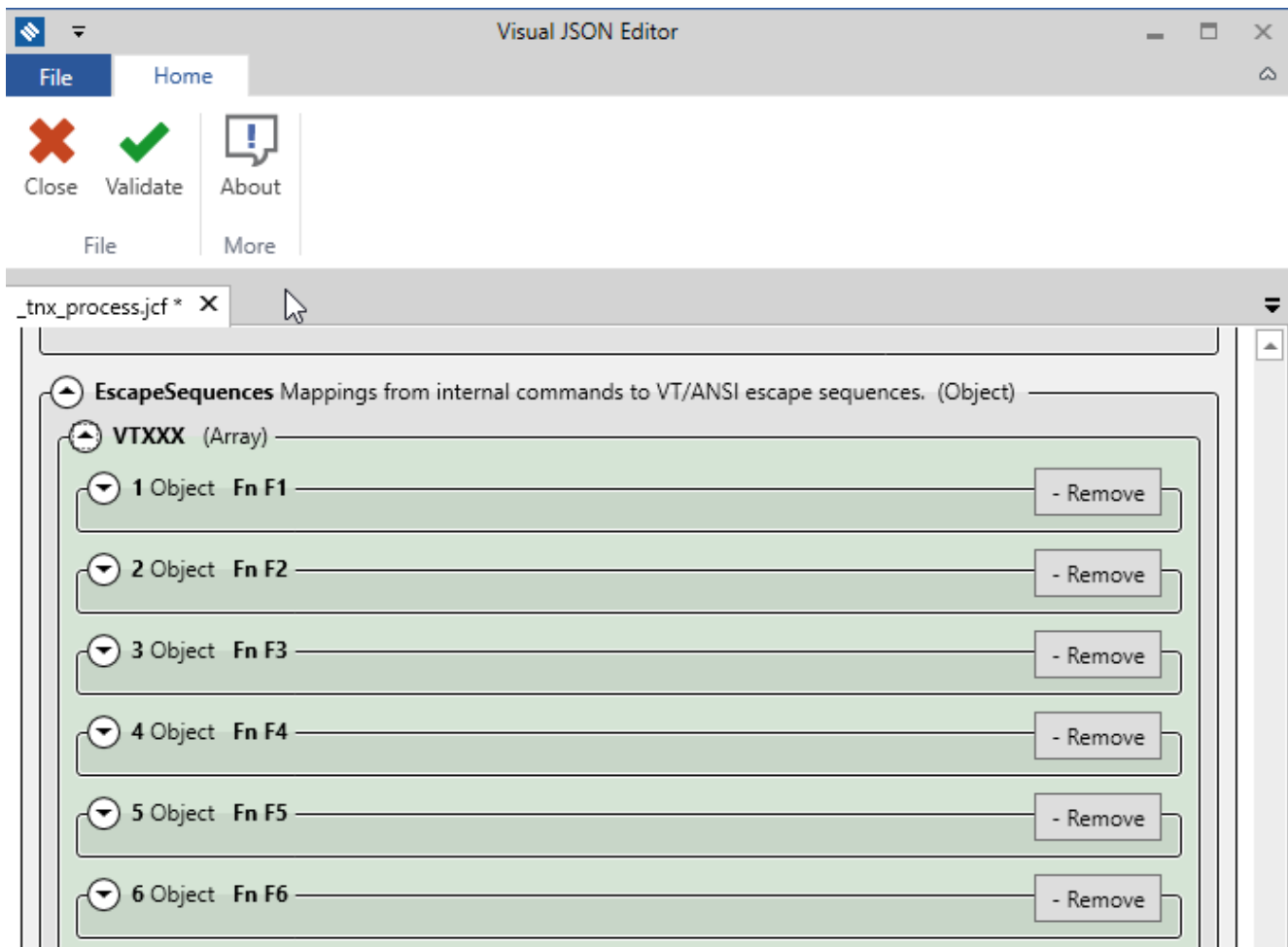
**ScanCode**  
Scan code as a two bytes hexadecimal value. The low order byte being the virtual key code, and the high order

0123

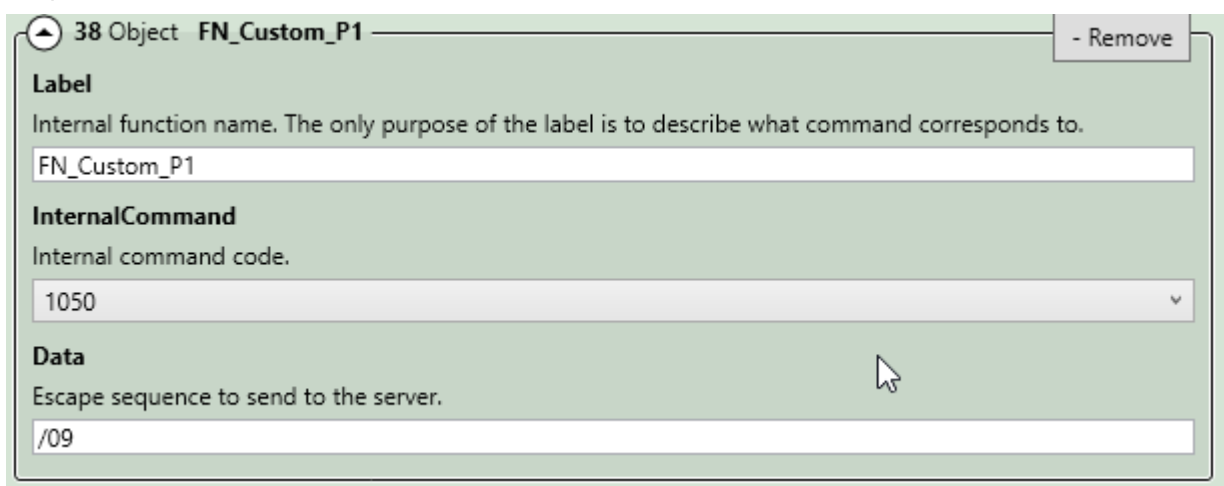
**InternalCommand**  
Associated VT command to send to the server.

1050

Next, goto EscapeSequences param



Add a new value to the array to associate the Internal Command *1050* to the corresponding Escape Sequence, in our case, to TAB (*/09*)



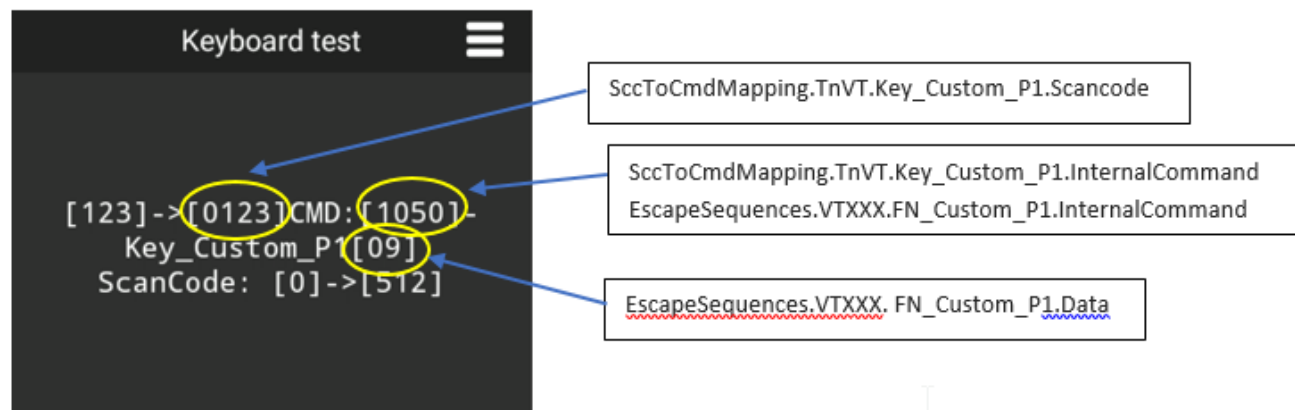
The screenshot shows a form for adding a new value to the 'EscapeSequences' array. The form is titled '38 Object FN\_Custom\_P1' and has a '- Remove' button. It contains three sections:

- Label**: Internal function name. The only purpose of the label is to describe what command corresponds to. The input field contains 'FN\_Custom\_P1'.
- InternalCommand**: Internal command code. The dropdown menu shows '1050'.
- Data**: Escape sequence to send to the server. The input field contains '/09'.

Validate, save and close the file

Push it to the device and open again TellNext in the Android terminal.

If we check again the association through *Keyboard test* menu, we can see the new function assigned:



So, if we load the Telnet login screen, we can test the TAB function pressing [P1] key.