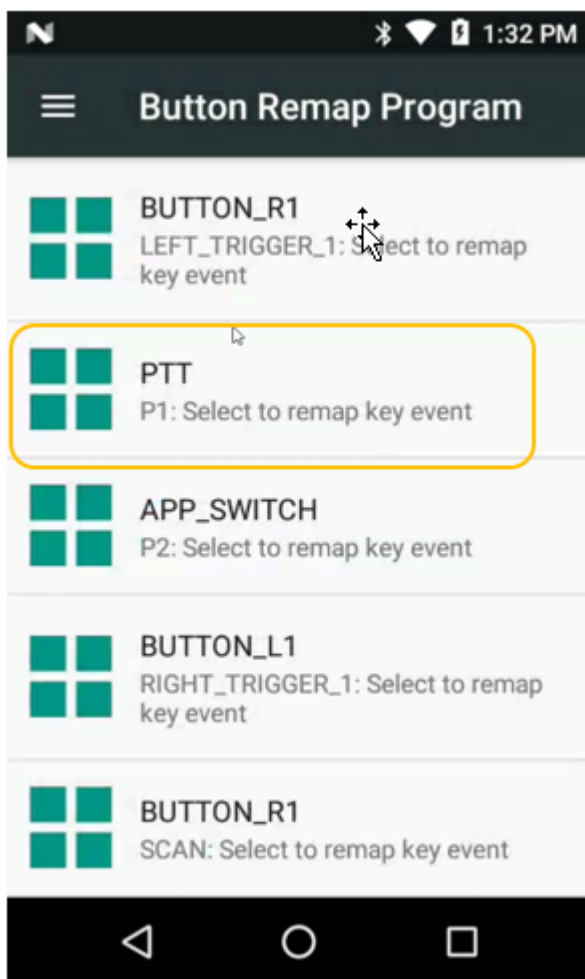In this sample we wish to map the [P1] physical key on a Zebra MC33 termnal.

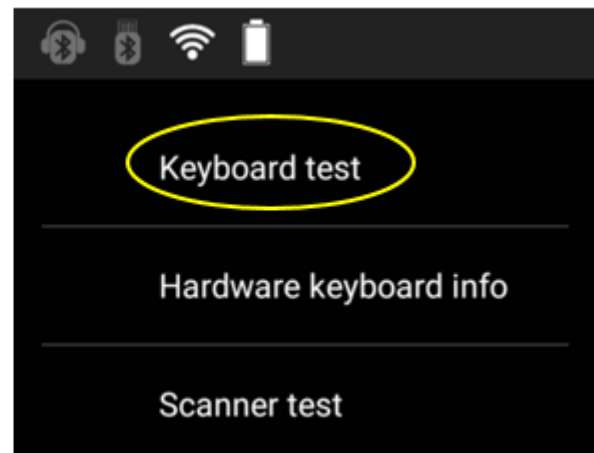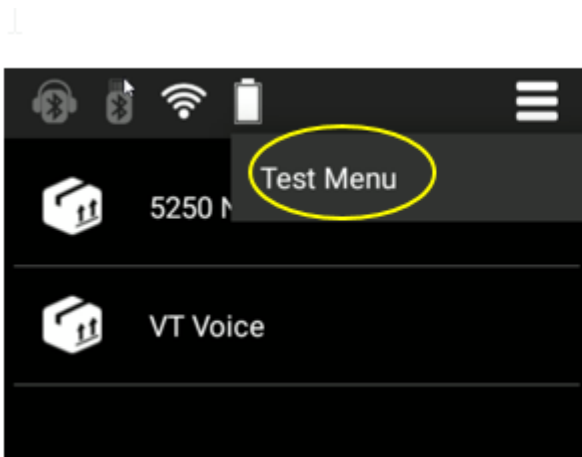Working with IBM5250 emulation:

First, we have to open *Settings – Key Programmer* to assign a non-used Telnet 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 button to remap [P1] and touch an Android function you don't use, to map to the button (in our example, PTT).
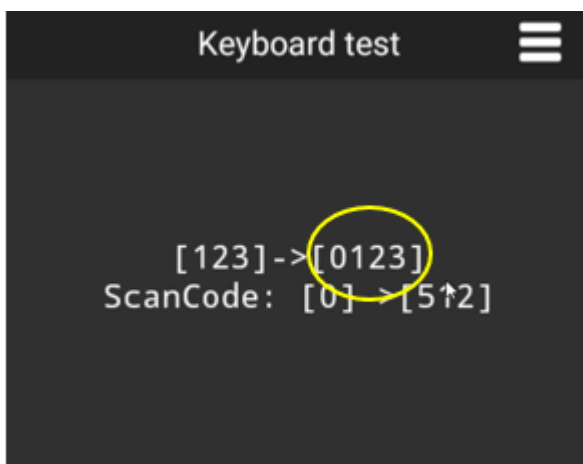

Open TellNext app, select a Profile and open *Test Menu – Keyboard* test at the right–top of the screen

Change the process you want to test to the corresponding 5250 Process.

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

Exit TellNext in the terminal until get the Android Desktop screen.

The key mapping, in this case, is limited to the Internal Command association. We can assign any function defined by IBM for 5250 emulation.
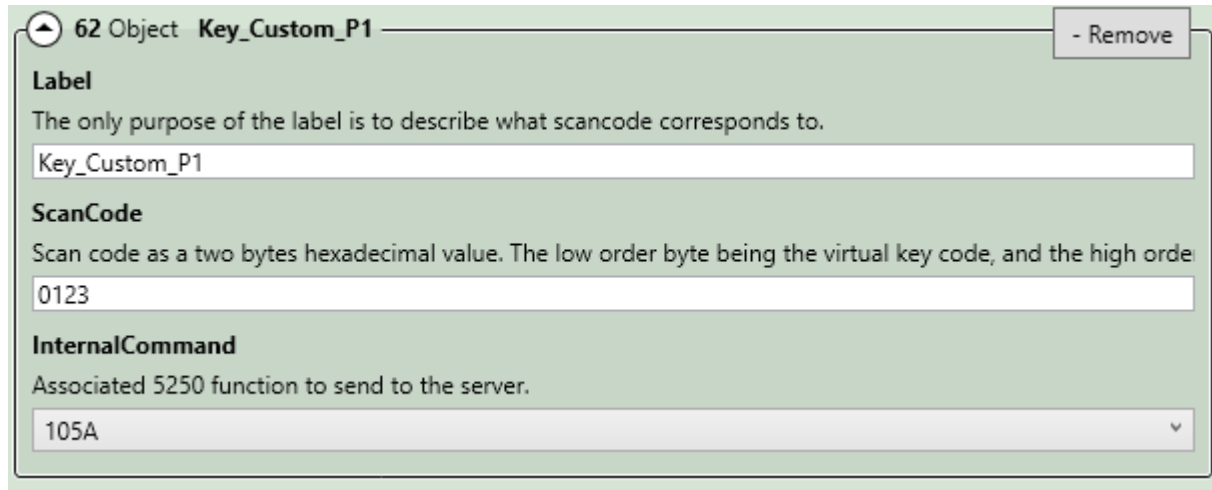See 5250 internal commands here.

Edit, with VisualJsonEditor, the _tnx_process.jcf file associated with the 5250 emulation Process you want to use.

Hub One

Visual JSON Editor — □ ×

File | **Home**

✖ Close — ✔ Validate — 💬 About

File — More

_tnx_process - Copy.jcf ✕

▼ **General** General settings. (Object) ─────────

▼ **Hosts** List of available hosts to connect to for this profile. (Array) ─────

▼ **Emulation** Login information to login automatically once connected to the server. (Object) ─────

▼ **Network** Server, proxy, protocol, telnet screen, etc. (Object) ─────

▼ **Display** Screen size, user interface, etc. (Object) ─────

▼ **Scripting** Log file, error messages, etc. (Object) ─────

▲ **Keyboard** Virtual keyboard, key mapping, etc. (Object) ─────

LockKbdKey

Key which activates or deactivate the keyboard depending on its current state. When the keyboard is deactivated, only thi

`0000`

LockKbdDo

☐ Performs the keyboard Locking feature. When True, the keyboard is locked and it is needed to press the LockKbdKey t

  ▼ **VirtualKeyboard** Virtual keyboard related settings. (Object) ─────

  ▼ **CustomKeyboard** Custom keyboard related settings. (Object) ─────

  ▼ **SccToSccMappings** (Array) ─────

  ▲ **SccToCmdMappings** Mappings from scancodes to internal command. (Object) ─────

    ▲ **Tn5250** 5250 scancode to command mapping list. (Array) ─────

      ▼ **1** Object **KEY_CUR_LEFT** ───── - Remove

      ▼ **2** Object **KEY_CUR_UP** ───── - Remove

      ▼ **3** Object **KEY_CUR_RIGHT** ───── - Remove

      ▼ **4** Object **KEY_CUR_DOWN** ───── - Remove

Goto Keyboard.SccToCmdMappings.Tn5250 and add a new value to the array. Set the scan code you get at Keyboard Test for the desired function.

Note: Usually the scan codes begin with "0" and internal commands with "1" or "2".

As we mention, we are limited to the already-implemented IBM functions for 5250 emulations. In our example, the Internal Command associated (105A) corresponds to the 5250_FORWTAB function.

Warning: if you assign an existing scancode, the function applied will be the one associated to the first register found in the list. For example, if we already have the 0123 scan code assigned to the 2031 internal command, the [P1] key will execute an ENTER instead of a FORWARD_TAB.

See the information about 5250 internal commands.

Validate, save and close the file
Push it to the device and open again TellNext in Android Terminal.

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

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