Keyman Developer Tutorial

Testing a Desktop Keyboard

Session 3

This session we will test a desktop keyboard for the Dagbani language of Ghana.

1. Start **Keyman Developer**.

2, In the **Project** menu, point to **Recent Projects**, click **DagbaniTutorial.kpj**.

3. In the **Project - Keyboard** dialog box, click **Keyboards**. Then click **dagbanitutorial.kmn**. The **Details** page appears.

4. Click **Layout**. The **Layout** page appears.

5. In the **Keyboard** menu, set **Include Debug information**.

6. In the **Keyboard** menu, click **Compile Keyboard**. Or we could use the shortcut **F7**.

   If the keyboard has compiled successfully, we will see two green messages saying **Success:** … in the **Messages** pane. If the messages are red, the compilation has failed. The error should indicate on what line in the code the error occurs. We will need to identify the error and then correct it and then recompile the keyboard.

7. After a successful compilation, we are ready to test the keyboard. In **Keyboard** menu, click **Test Keyboard**.

   Below the **Code** tab is a testing box where we can test the keyboard by typing the various character sequences to ensure that they will produce the desired results. Enter each character sequences we have defined in the code to verify all is working properly. Do not forget to check the capital sequences as well.

   To change the font and its size, right-click in the testing box, and then click **Font** in the popup menu. Use the **Font** dialog box, to change the characteristics of the font to our preferences and then click **OK**.

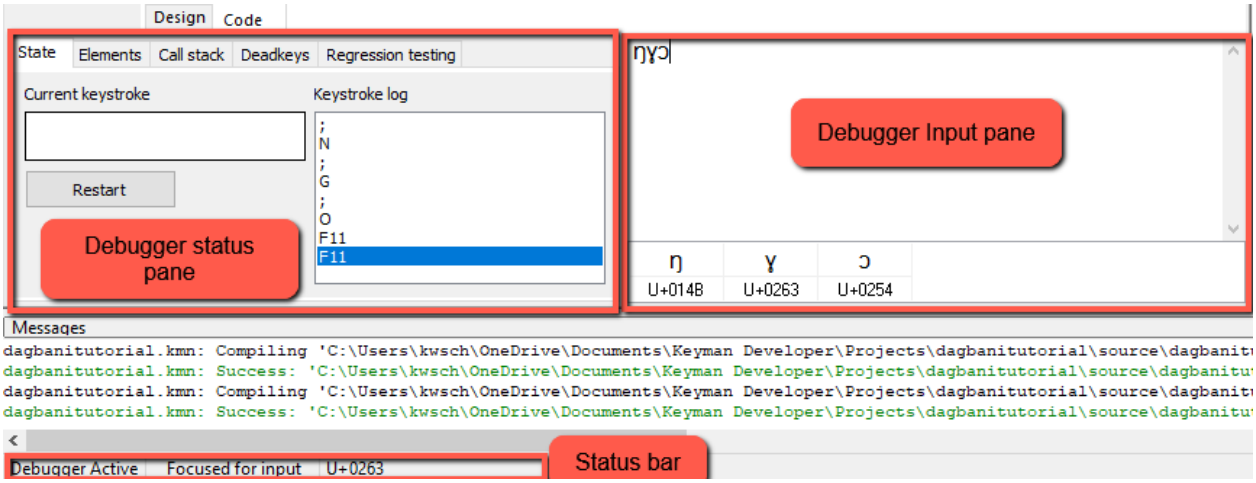
8. When we have finished our test, in the **Debug** menu, click **Stop debugging**.

   If one or more of the character sequences are not working, we need to go back to the code to fix it and then repeat the process of compiling and testing.

Debugging Errors.

In this section we will look at the debugger. When we have troubled sorting out our errors, the debugger can be helpful. To start the debugger, in the **Debug** menu, click **Start debugging** or press **F5**.

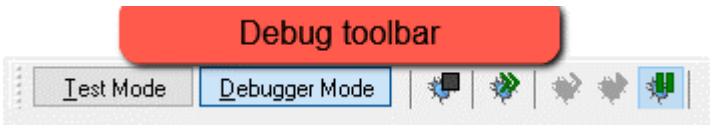
1. the **Debug** window is shown at the bottom of the **Layout** pane when debugger is activated. There are several user interface areas: The debugger input pane, the debugger status pane and the status bar which shows the current debugger status. Below is a picture of the Debug window.

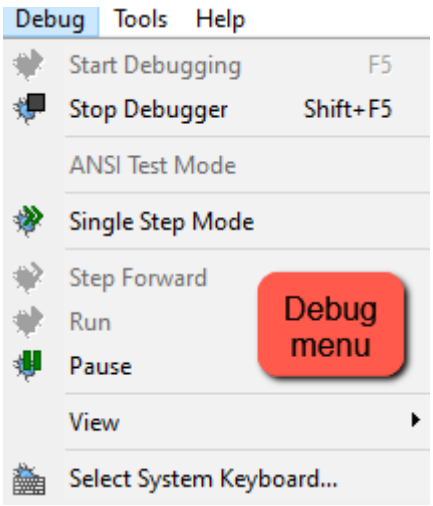


Debugger input window - The debugger input window is used for typing input to test the keyboard. In the top half of this window, type the characters that will produce the special characters. Testing the keyboard will displayed the same as in use, with one exception: dead-keys will be shown visually with an OBJ symbol.

The lower half of the window shows a list of the characters that are generated by the keyboard, along with their Unicode values.

2. The debug toolbar which is usually docked under the menu and the debug menu control the debugger.





Below is a short explanation of the debugger commands found in the Debug menu and toolbar.

Set/Clear breakpoint - A breakpoint can be put on a line of code to ask Keyman Developer to stop and show what is happening in the keyboard layout when that line is reached. We placed a breakpoint by clicking to the left of the line number. Typically, we would put a breakpoint on a rule. So, we could debug an unexpected behavior on that rule.

Start Debugging – This starts the debugger.

Stop Debugger – This stops the debugger.

Single Step Mode - When this option is active, we will step through every group and rule that our keyboard applies when a keystroke is pressed. The **Step Forward** and **Run** controls are only relevant in Single Step Mode.

Step Forward - Move to the next step in our keyboard code.

Run - Stop processing the current keystroke event in single step mode (unless a breakpoint is hit). The next keystroke event will drop into single step mode again.

Pause - Use the **Pause** command to pause the debugger. When the debugger is paused, it will not accept any input. Ordinary shortcut keys (**Shift+F5**, **Alt+Tab**, etc.) will function as usual. Press **Pause** command again to resume debugging.

3. There are five tabs in the **Debugger status** pane. The **State** tab shows the internal state of the keyboard interpreter. This tab shows the current keystroke state, and the sequence of keystrokes that were typed to arrive at this state. Clearing the text in the debug window will also clear the keystroke log; as will clicking **Restart**. The **Restart** button is disabled while stepping through an event.

4. The **Elements** tab shows the elements that make up rule currently being processed: the context, the key and what the output will be. If the rule uses stores, the contents of the store will be shown in the right-hand column, with the matched letter in red.

5. The **Call Stack** tab shows all the lines that have been processed to this point are shown in a list. We can double-click on any entry in the list to display the line in the keyboard source.

6. The **Deadkeys** tab lists all the dead-keys that are currently in the context. We can select one from the list to see it highlighted in the debug input box. This information can also be seen in the character grid in the lower half of the debugger input window.

7. The **Regression testing** tab records a sequence of keystrokes and outputs the generated characters of the keyboard to test for the same behavior when we make changes to the keyboard.

8. Try using the debugger in our project to get familiar with its features.