
Ukelele Tutorial

A guided introduction to using Ukelele to
create keyboard layouts



John Brownie

Ukelele 3.1

Ukelele Tutorial4

Organisation of the tutorial.....4

Creating a keyboard layout.....5

Designing the Keyboard Layout.....5

Getting Started6

Editing the keyboard layout.....12

 Creating a dead key13

 Adding output to keys.....15

 Finishing the dead key.....19

 Completing the keyboard layout21

 Installing the keyboard layout.....21

 Enabling and using the keyboard layout23

More Complex Keyboard Layouts27

Introduction.....27

Procedure28

Comments, Questions and Answers35

 The Capacities of Ukelele.....35

 Available Keyboards.....35

 The Character Palette.....35

 Why have we neglected the accent tilde?36

 By sacrificing our numbers, did we do the right thing?36

 What if we have no number pads?.....36

 Erasing36

 Ukelele Errors36

 A Few No-Nos.....37

 Once we have completed37

 Bridging the difference between the electronic and the physical keyboard37

More Advanced Topics38

Designing the Keyboard Layout.....38

Getting Started38

Creating the Dead Keys42

 The first dead key.....42

 The second dead key45

 The third dead key.....47

 The fourth and fifth dead keys.....50

Unlinking Keys	50
----------------------	----

Ukelele Tutorial

Ukelele is designed to make the creation of custom keyboard layouts for OS X (formerly Mac OS X, and soon to be known as macOS). There are lots of options and commands to make this as flexible as possible, but this means that there are some rather complex features to understand in order to get the most out of Ukelele. Everything is documented in the manual, but this tutorial is aimed at getting you into working with keyboard layouts.

Organisation of the tutorial

The tutorial comes in three parts. The first works through creating a fairly simple keyboard layout. The second, courtesy of Peter Thomas, works through a somewhat more complex keyboard layout. The third part introduces two-level dead keys by producing another complete keyboard layout. It also and explains the concept of linked keys by working through a case where you need to know about it, and shows what needs to be done.

Part 1

Creating a keyboard layout

The Yolngu keyboard layout

This tutorial takes you through the steps to create a keyboard layout in Ukelele. Some basic steps are omitted, such as launching the program.

The keyboard layout that is developed here was created the Australian Society for Indigenous Languages (AuSIL) for the Yolngu language, spoken in Arnhem Land, in the Northern Territory, Australia, and is also useful for other Australian Aboriginal languages, such as Pitjantjatjara. It is not a particularly complex keyboard layout, but it should illustrate the usage of Ukelele in a real situation.

Designing the Keyboard Layout

The first thing that should be done is to plan the keyboard layout. This needs to be done with several things in mind, and should be completed before you start creating the keyboard layout with Ukelele.

The following is the description of the principles that went into making this keyboard layout:

1. We want all keys available for both Aboriginal languages and English without losing functionality.
2. High frequency sounds (like /ŋ/) should not be on the outermost sphere of keys.
3. High frequency sounds should not require complex keying sequences (such as SHIFT + CONTROL, followed by a sequence of two-letter keystrokes) .
4. Combinations such as <shift> should only have the function of getting upper case letters (plus a few standard computer commands).
5. Speakers of Aboriginal languages should have the full range of Unicode typefaces available to them.
6. There are three dead keys in this keyboard layout. The first is the semicolon key, which produces modified letters. The other two are the less than (<) and greater than (>) keys, which are used for producing typographer's quotes (‘ ’ “ ”). The terminology used by the developers of the keyboard is that a dead key is followed by a base key to produce an output glyph, or character.

The dead key combinations are listed in the following table:

Dead Key	Base Key	Resulting Glyph	Unicode Code Point
;	j	ŋ (eng)	U+014B

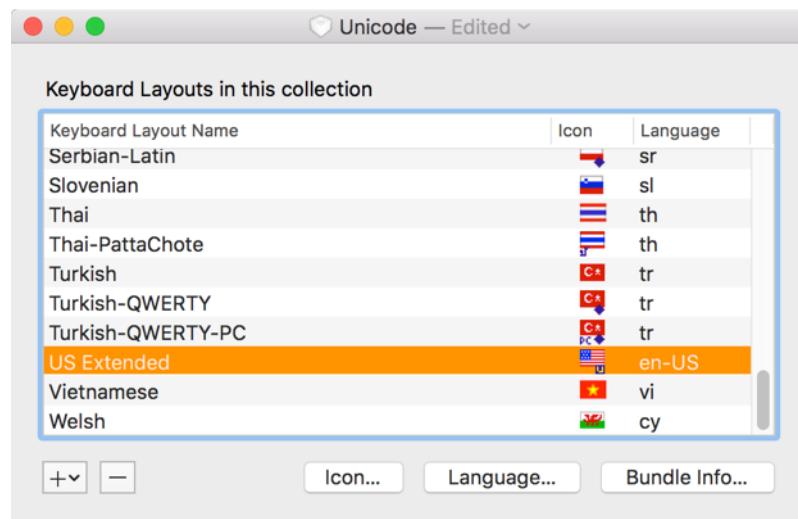
Dead Key	Base Key	Resulting Glyph	Unicode Code Point
;	J	Ñ (capital eng)	U+014A
;	a	ä	U+00E4
;	A	Ä	U+00C4
;	d	đ	U+1E0F
;	D	Đ	U+1E0E
;	I	Í	U+1E3B
;	L	Ł	U+1E3A
;	n	ń	U+1E49
;	N	Ń	U+1E48
;	r	ŕ	U+1E5F
;	R	Ŕ	U+1E5E
;	t	Ń	U+1E6F
;	T	Ń	U+1E6E
;	;	;	U+003B
;	(space)	;	U+003B
<	(space)	‘	U+2018
<	<	“	U+201C
<	{	<	U+003C
>	(space)	’	U+2019
>	>	”	U+201D
>	}	>	U+003E

Getting Started

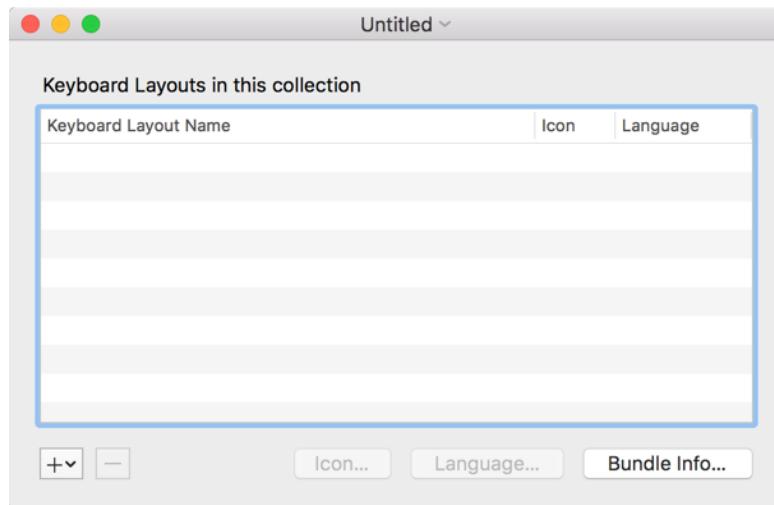
Since the keyboard layout that we'd like to create is very similar to the standard English QWERTY keyboard layout, we will begin by basing our keyboard layout on the US Extended layout, which is provided with Ukelele.

Open up the Resources folder on the installation disk image, then open the Standard Keyboards folder. You'll see an item there called Unicode. Unfortunately, you can't just double-click that to open it in Ukelele. Instead, open Ukelele and choose Open... from the file menu, then choose the Unicode document. Alternatively, drag the Unicode document onto the Ukelele icon, which will also open it.

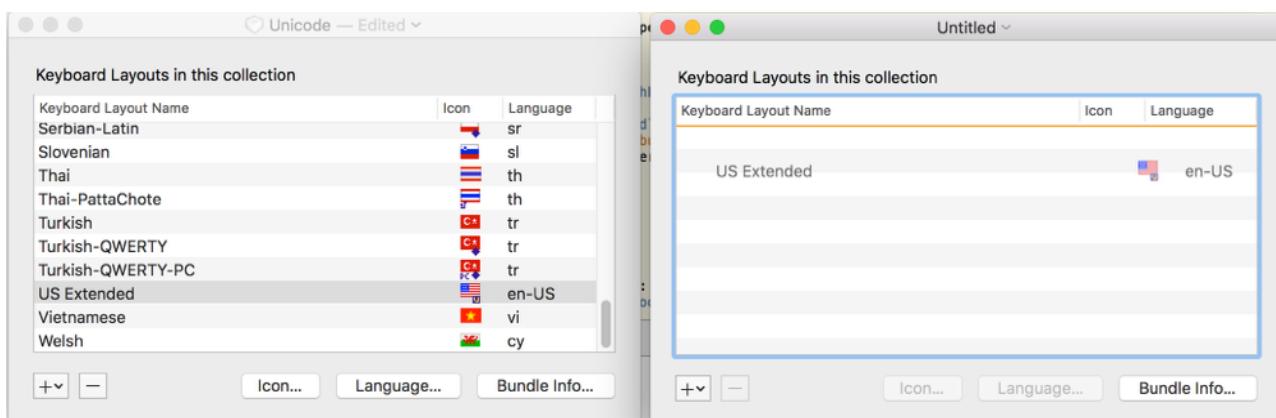
Scroll down until you find the US Extended entry, and select it. You should now have a window that looks like this:



Now, choose New Keyboard Layout Collection from the File menu, which will create an empty keyboard layout collection, which will look like this:



What you need to do next is to drag US Extended from the Unicode window to your new, Untitled window:



Once you have done this, your window should look like this:

Untitled — Edited

Keyboard Layouts in this collection

Keyboard Layout Name	Icon	Language
US Extended	USA flag icon	en-US

+ - Icon... Language... Bundle Info...

Now we're going to change the name of the keyboard layout from US Extended to Yolngu. We can do this a number of ways, but for this we'll right-click on the entry, and pick Set Keyboard Name and Script... from the contextual menu that appears:

Untitled — Edited

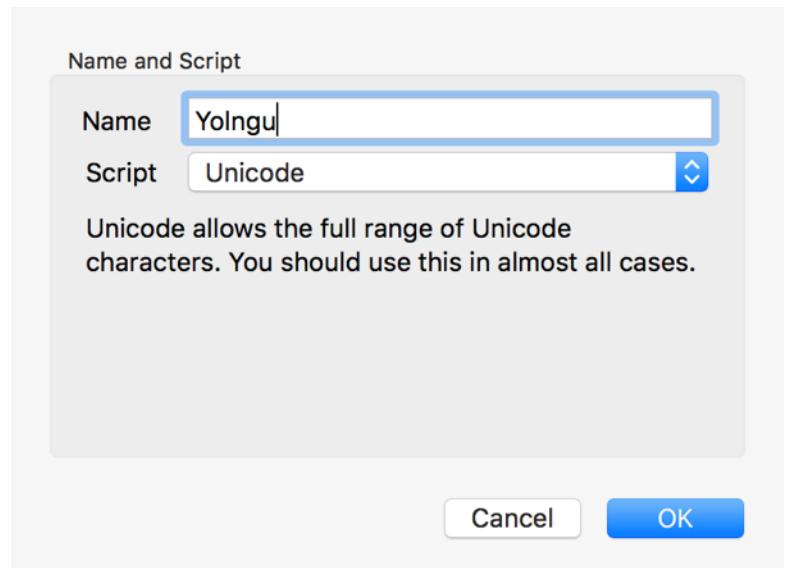
Keyboard Layouts in this collection

Keyboard Layout Name	Icon	Language
US Extended	USA flag icon	en-US

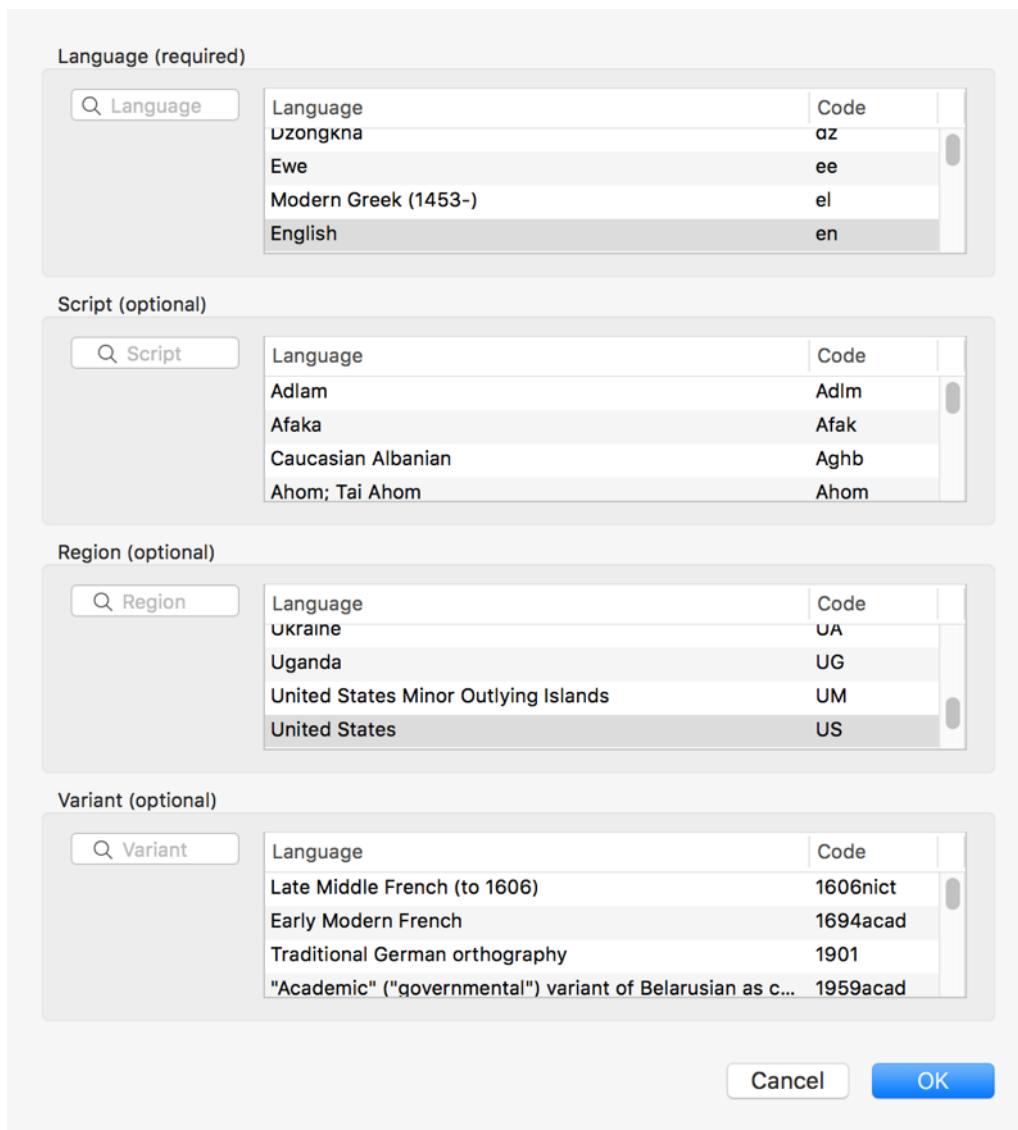
A context menu is open over the "US Extended" row, with the "Set Keyboard Name and Script..." option highlighted.

+ - Icon... Language... Bundle Info...

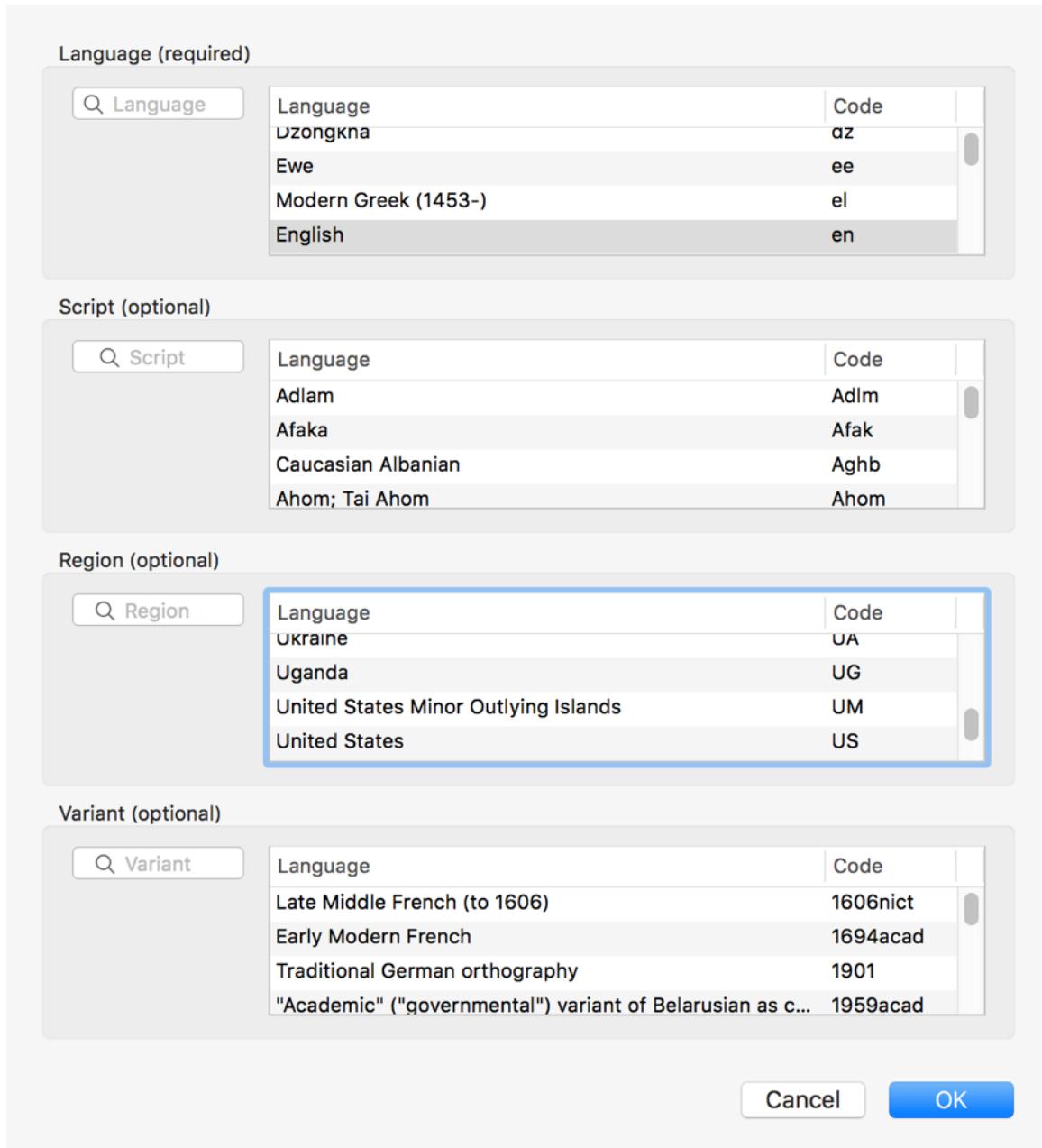
This brings up a dialog, and you should change the name to Yolngu, but leave the script as Unicode:



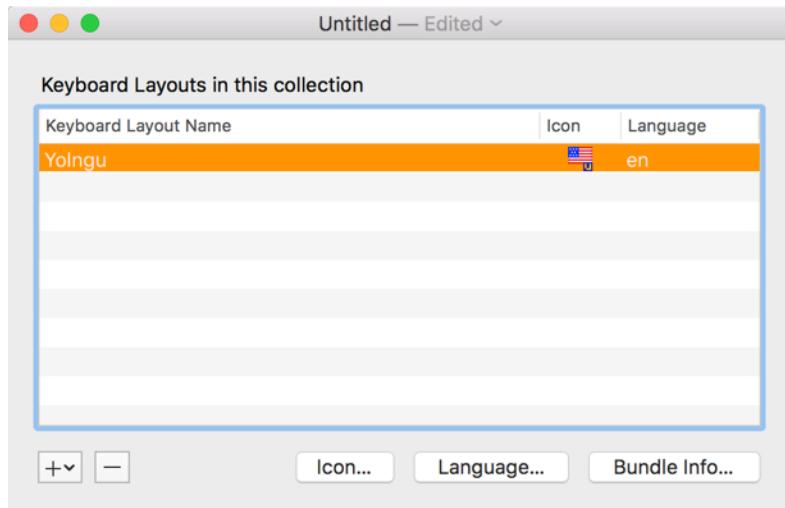
Click OK, and we are done with that. We're also going to change the language from US English to just plain English, so click on the Language... button, which brings up this dialog:



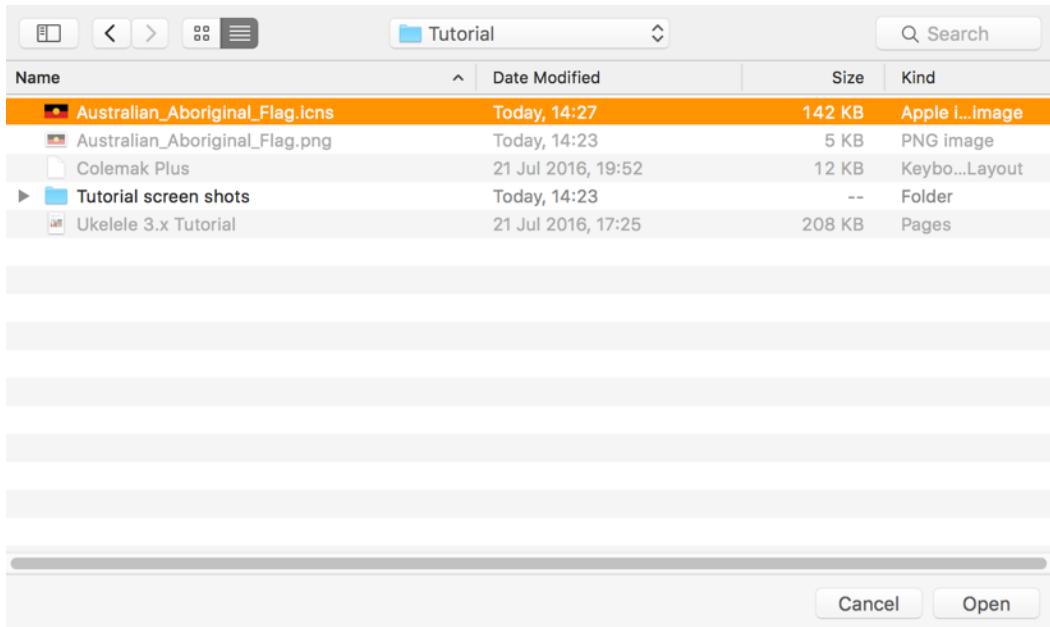
You'll see that English is selected in the Language section, which is what we want, so we don't need to change anything there. In the Region section, United States is selected, and we don't want that. To get rid of it, simply hold down the command key and click on United States, which will leave nothing selected:



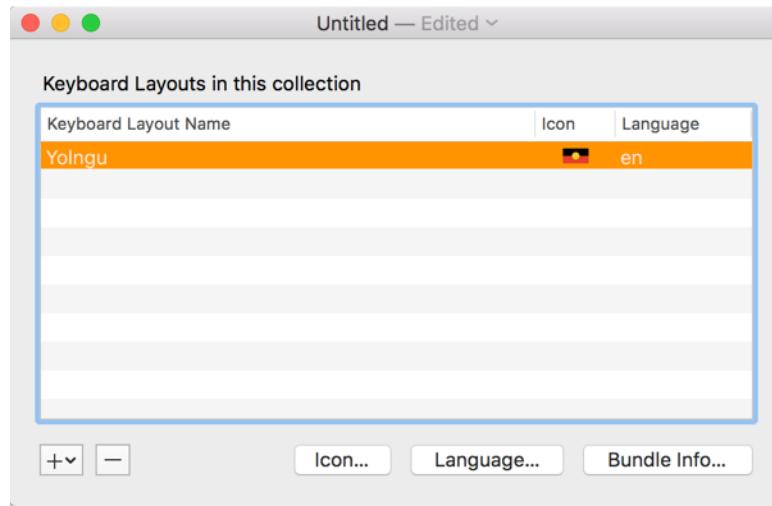
Click OK, and we're back to our window:



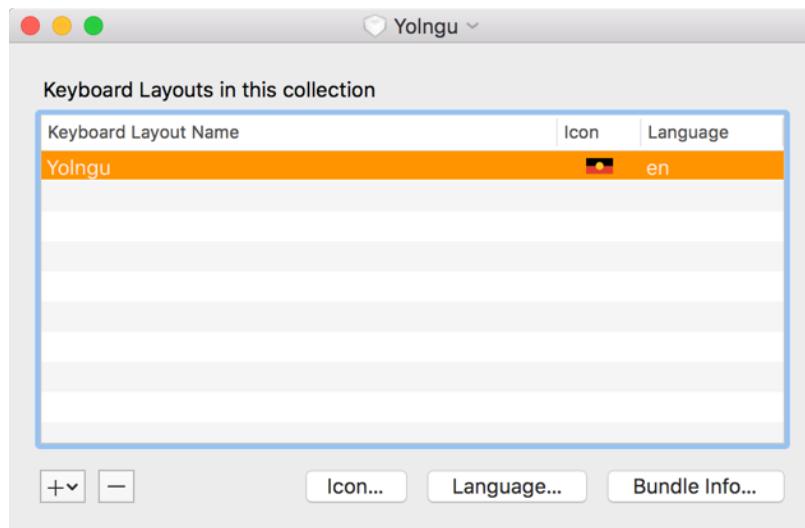
Next, we'll replace the US flag with an Australian Aboriginal flag. I pulled the Aboriginal flag from Wikipedia, then used Image2Icon to convert it to the correct format, with a .icns extension. Click on the Icon... button, and select the flag icon:



Click Open, and you will then have as follows:



Finally, for this part of the process, we'll save the keyboard layout collection with the name Yolngu (you can see that it is still currently untitled). Choose Save from the File menu or ⌘S, save it where you like with the name Yolngu, and you should see this:



Editing the keyboard layout

If you double-click the keyboard layout, you will get the keyboard window to appear:



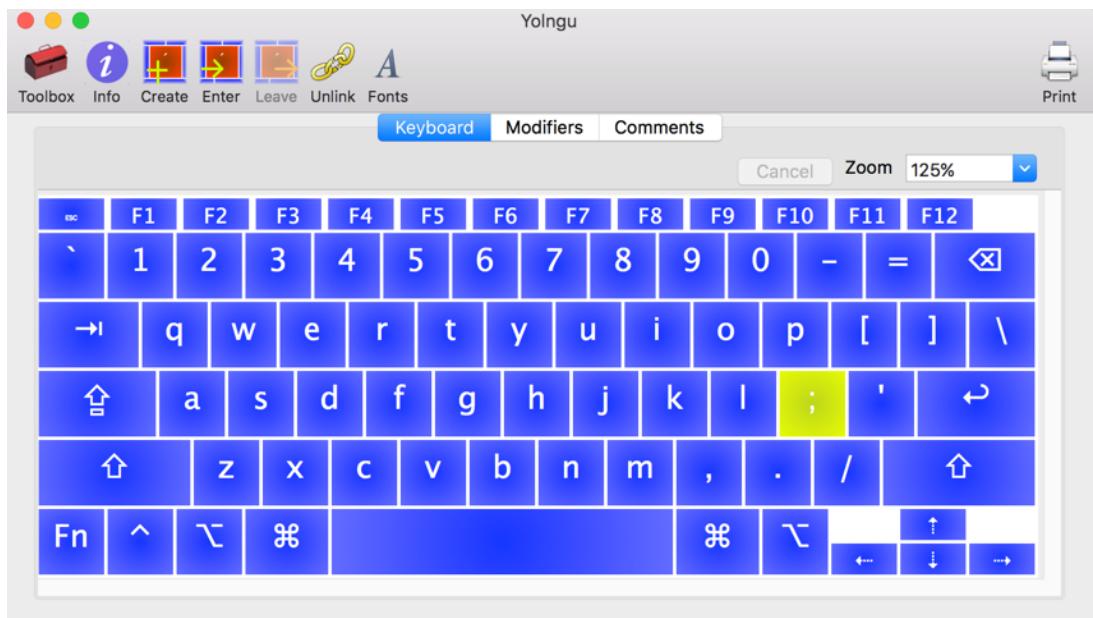
Note that your computer may show differently. Ukelele tries to match what you see on screen with the hardware keyboard that you are using. It can only get Apple keyboards correctly, and not always reliably, due to a variety of factors. The screen shots here are of a MacBook Pro. You can change the appearance by selecting keyboard type from the View menu. To get yours to look the same as the screen shots, you need to choose “Aluminium Wireless” with ANSI coding.

By starting our keyboard layout based on one that is close to what we want, we have only a little work to do. In this tutorial, we only need to define the three dead keys, and the keyboard layout will be complete.

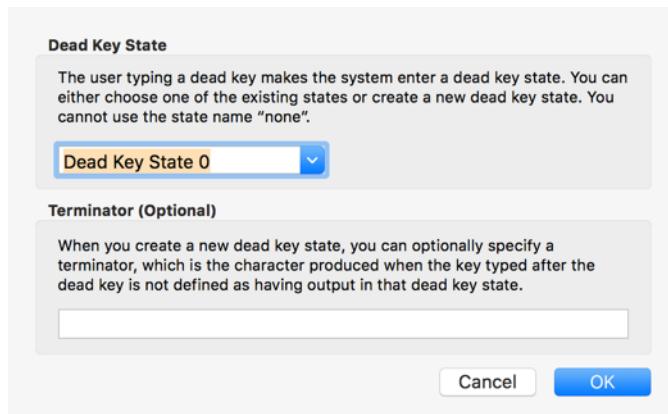
Creating a dead key

We begin with the most complex of the three. In fact, none are particularly complex, with the only difference being the number of key combinations that need to be defined. For this, the dead key is the semicolon key, with no modifiers pressed.

We begin the process by choosing by selecting the semicolon key, either by pressing the physical key, or by clicking the representation on screen. At this point the key on screen will change colour to show that it has been selected:



We now select **Create Dead Key...** from the Keyboard menu (or click the Create button on the toolbar), which brings up the following dialog:



Since this keyboard layout already has some dead keys defined, you could choose one of them in the top part of the dialog (click on the arrow if you're curious), but this is not what you want to do. We want a new dead key state. Any name will do, so we'll just accept the default at this point.

A *terminator* is optional. It is the output produced by the dead key if the next key has nothing defined. It makes sense in places like accents, to get the accent alone. Here, we don't need anything, so we'll just leave it empty, and click OK. This brings up the keyboard in the new dead key state, which has no output defined:



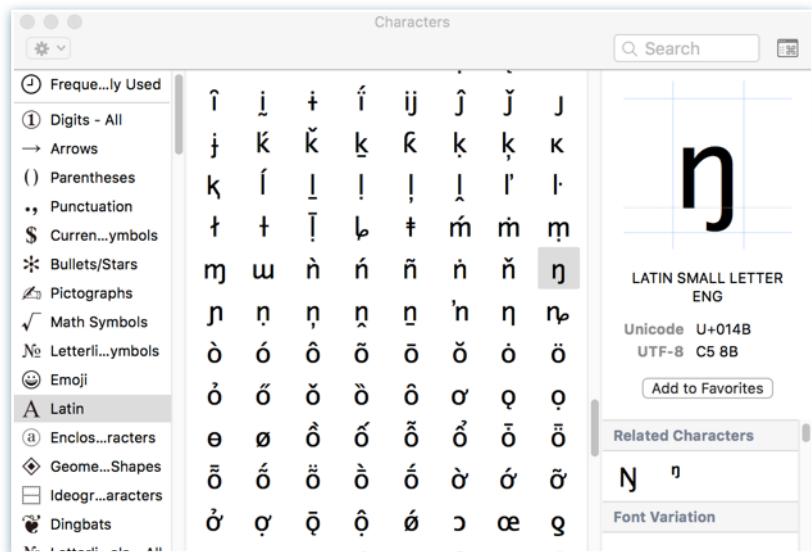
You can see that most keys appear slightly greyed out. This indicates that no output is defined in this dead key state. The output you see is what would be output when no dead key is active, and is there to help you choose the correct keys.

Adding output to keys

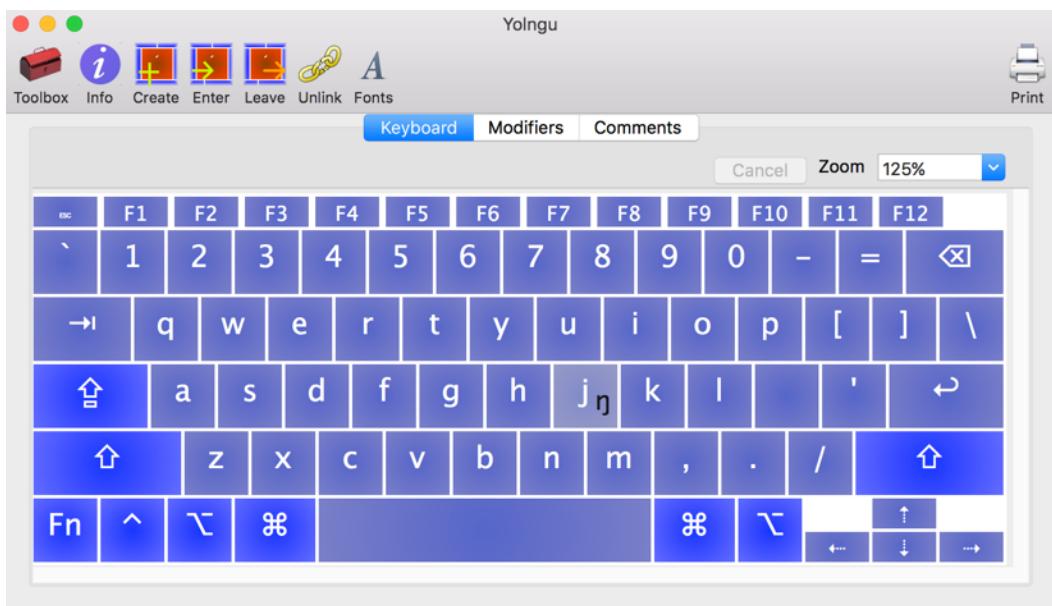
Now that we are in the dead key state, we add the output that we want. We will start with the ŋ key, which should produce Unicode code point U+014B, ŋ . There are several ways to do this. For this one, we will use drag and drop, using the Character Viewer (also known as the Character Palette in older versions of Mac OS X) as the source. In OS X 10.11, this is shown by choosing Show Emoji & Symbols from the input menu (the one with a flag on the right side of the menu bar), or Emoji & Symbols from the Edit menu in most applications.

The Character Viewer looks different in nearly every version of OS X, so details may well be different to the screen shots here, which will show the version as of OS X 10.11.6, El Capitan.

You can find the ŋ character in various ways, which depend on the version of OS X you are using. In 10.11, it is in the “Latin” section. Alternatively, you can search for “eng” and it is near the top. Again, you could also find it by selecting “Unicode” from the View popup, and working down to 014B, in the Latin Extended-A section.



Whatever way you get it, we now want to drag the η to the j key in the main window. Click and hold the η in the Character Viewer, then drag it onto the main window. As it goes over keys that can accept a drag, the key will receive the drag highlight, which lightens its colour.



Once you release the mouse, you will see that the η is now in place:



Having done the j key, we want to handle the a key, which produces ä. To show a different method of adding output, this time we will use the popover. To get it, you double-click the a key. The popover has a text entry box, and you can add the output in several ways. If you are using a US keyboard layout or similar, typing ñu a gets ä. Alternatively, you could find ä in the Character Viewer, and double-click it to enter it.



You can continue to add the output for the various keys using either method. In this way, all the output for the unshifted keys can be added, to get the following:



We still need to handle the uppercase letters that need to be produced in this dead key state. We'll begin with the j key again, which will output ߱. The drag and drop method requires holding down the shift key when you drop the ߱ onto the j key:



For the Ä, we'll use the double-click method to get the popover, but we'll use the code point for Ä, U+00C4. So, hold down the shift key and double-click the a key, then enter Ä as the new output:



The code point is entered with XML or HTML notation, either as a hexadecimal number (as we just did), or a decimal number, by omitting the x. Also, leading zeroes can be omitted, and both uppercase and lowercase can be used for the numbers, so we could just as well use Ä, Ä, or even Ä to indicate the appropriate code point.

We need to fill in the remaining output, using whichever method you prefer. However, one further way to make the process easier is to use the “Sticky Modifiers” option. Just like pressing caps lock once makes it active and pressing a second time makes it inactive, Sticky Modifiers makes all the modifier keys behave this way. So, turn on Sticky Modifiers and press the shift key. You’ll see that it is highlighted in the keyboard window as normal. This means that you don’t have to keep pressing the key to change the output for several keys that have shift pressed. This isn’t revolutionary for this case, but it does make it easier for some more complex cases.

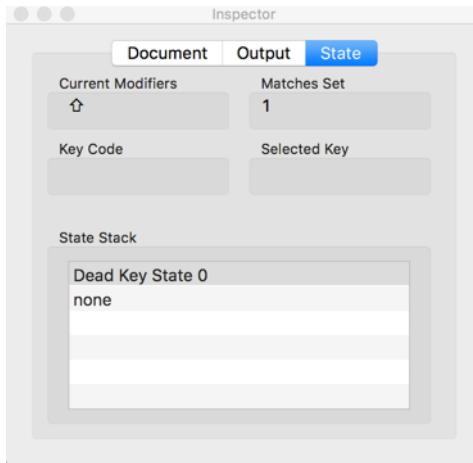
When you’re done, the keyboard layout should look like this:



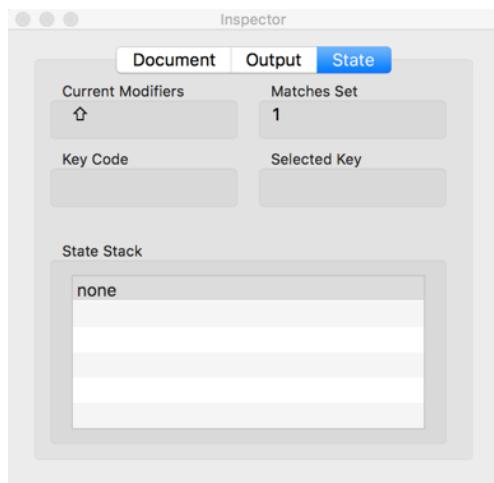
Finishing the dead key

Once you have put all the correct output for a dead key state, you need to finish the dead key. Dead key states are entered and left in a stack, so that the last one entered is the first one left. You can see the state stack in the inspector window, which you can show or hide by clicking on

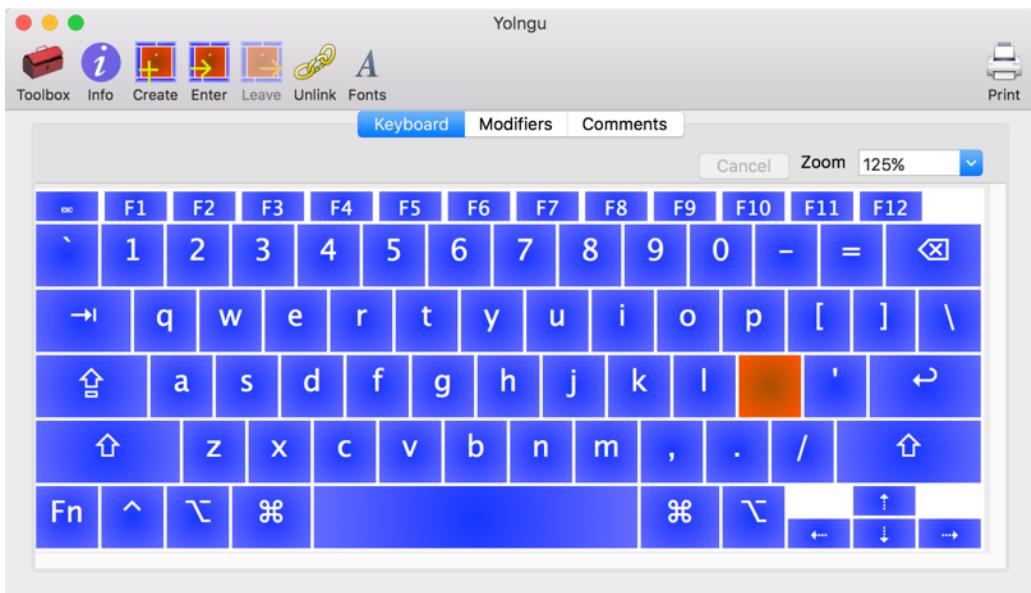
the Info button in the toolbar (or using the menu item Show Inspector in the View menu, or the keyboard shortcut ⌘⌘I), and selecting the State tab. You'll see a window like this:



Note that the top entry in the state stack is “Dead Key State 0”, which is the (default) name we used when creating the dead key. The entry below it is “none”, which is the special state which means that no dead key state is active. If you select Leave Dead Key State from the Keyboard menu, or click the Leave button on the toolbar, you’ll see that the top state is popped off the stack, and we return to state “none”.



And now the keyboard window shows the normal output, with the dead key blank and shaded red to indicate that it is a dead key:



Completing the keyboard layout

The first dead key is completed, but two others need to be created. The details of how to do these are the same as before, so we won't cover them in detail. One dead key will be the < key, which means with the shift key down, while the other will be >. Each has no terminator, and we can just use the default state names. Each has only three keys to assign output to.

Once you've completed these, the keyboard window will look like this with the shift key down:



You can close the window now, taking you back to the keyboard layout collection which contains the Yolngu keyboard layout.

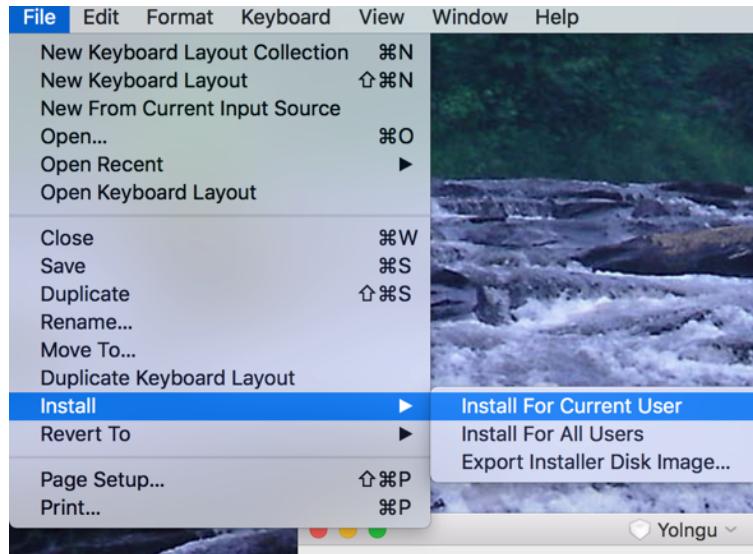
Installing the keyboard layout

Now that we have completed our keyboard layout, we need to install it, to make it available for our use. There are a number of ways to do this, each with its own reasons to do so. We will look at two different methods in this tutorial.

Installing for your own use

In the first method, we will install the keyboard layout for our own personal use on our computer. This is simple to do, but it can't easily be shared with another computer or another user of our computer. The next method is more useful if you want to share the keyboard layout with others.

This method is simple: From the File menu, choose Install, and Install for Current User from the submenu.

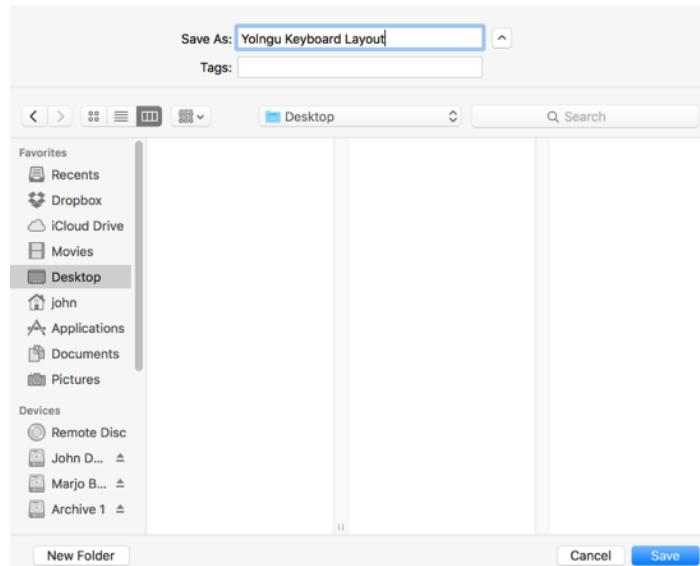


That is all that there is to installing the keyboard layout! Next, go on to the section on enabling and using the keyboard layout.

Creating an installer disk image

The second method creates a disk image which contains the keyboard layout. This can be shared with other users, who can use it to install the keyboard layout on their computer.

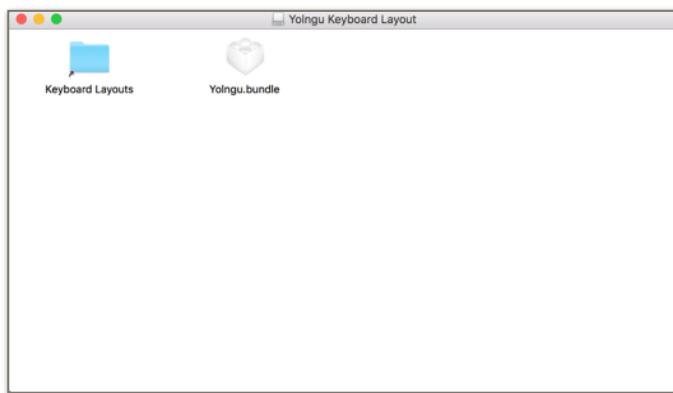
Again, it's straightforward: From the File menu, choose Install, and Export Installer Disk Image... from the submenu. This creates a standard Save dialog, and allows you to specify where you will save it, and the name for it. For now, we'll call it Yolngu Keyboard Layout, and save it to the desktop.



The save takes several seconds to complete. Once it's done, you'll find the file on your desktop:



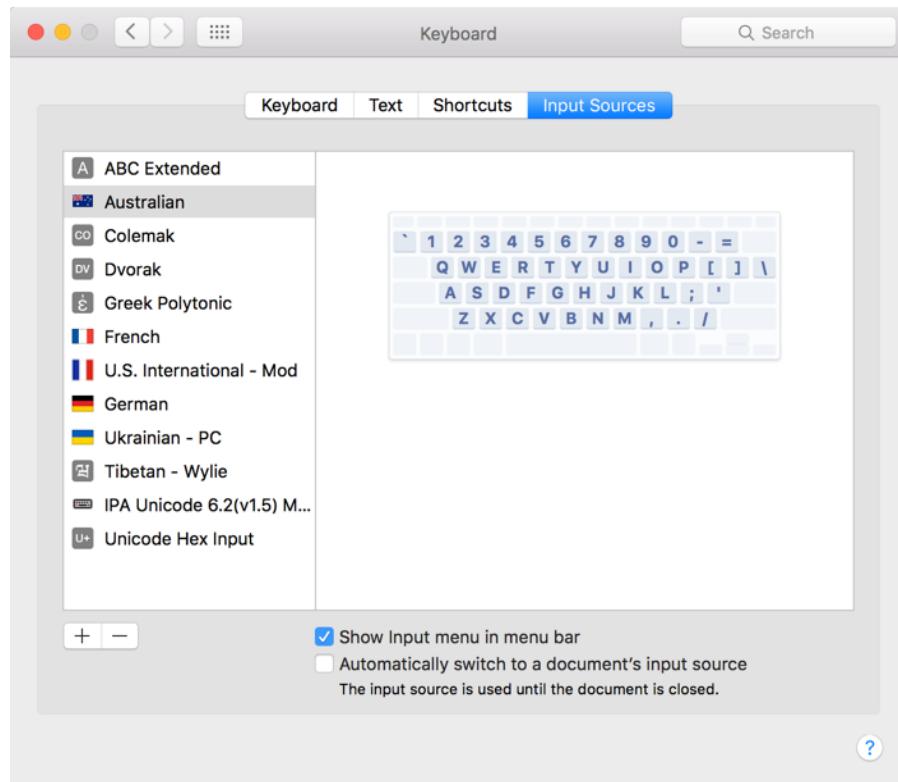
If you open the disk image, you will get a Finder window like this:



To install it on any computer, drag the Yolngu.bundle item onto the Keyboard Layouts folder, and it will install, asking for permission to do so. That's all that is needed.

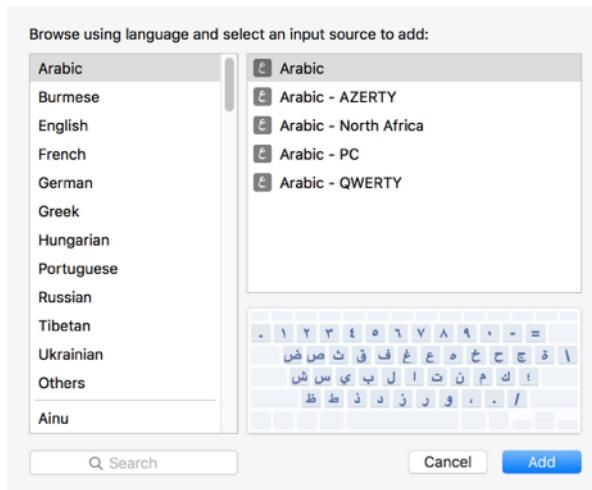
Enabling and using the keyboard layout

Once your keyboard layout is installed, it needs to be enabled. This is done in System Preferences. The exact location varies from version to version of OS X, but in 10.11 (El Capitan), it is the Input Sources tab of the Keyboard preference pane.

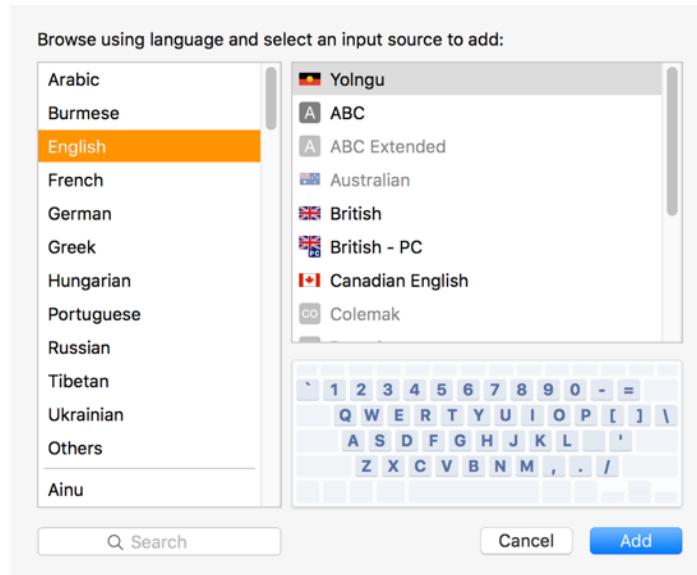


Again, appearance is different in different versions of OS X. The exact procedure that follow also differs, but we'll just use El Capitan's procedure, and others should be fairly readily followed where they differ.

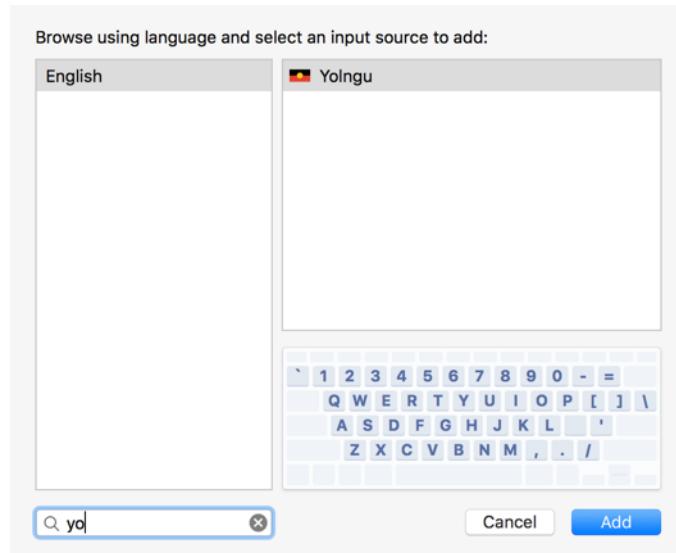
We want to add a new input source, so we click the + button in the lower left. This brings up a sheet where we need to select our keyboard layout.



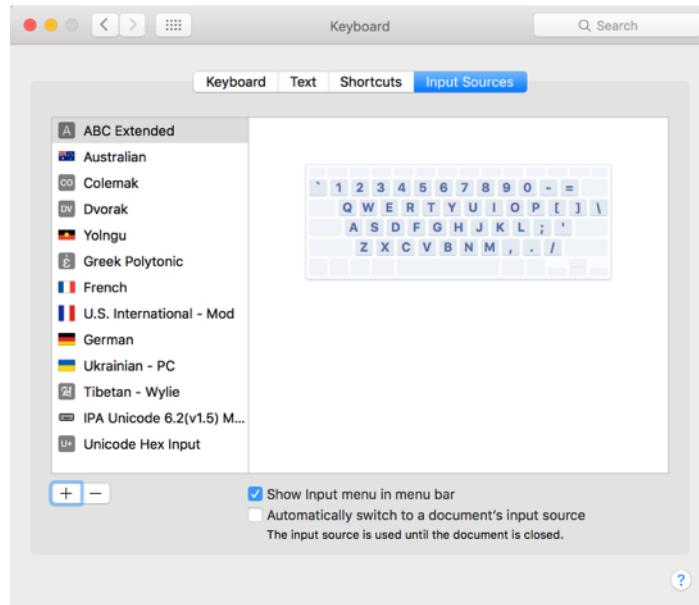
You'll notice that the left side has a list of languages, those languages you have installed first (including a catch-all "Others"), then a longer list of major languages. Since we set the language of our Yolngu keyboard layout to English, click on English on the left.



In our case, Yolngu appears at the top of the list. However, it may not be so simple, and the search field at the lower left helps. Just start typing Yolngu, and it will appear within a few keystrokes:



Now click Add, and you will see that Yolngu is now listed in the currently enabled keyboard layouts:



If you now open the input menu, you'll see that Yolngu is available to be selected:



To actually use the keyboard layout, you need to select it in the input menu. After that, it should just work! (Trouble-shooting tip: If it doesn't work in all applications, try logging out and logging in again, then activating the keyboard layout.)

And that's the end of this part of the tutorial.

Part 2

More Complex Keyboard Layouts

Provided by Peter Thomas

This part of the tutorial was written by Peter Thomas, representing a different perspective on using Ukelele. His main point is to show how to create dead keys to produce accented vowels, but without using the option key, as the standard US keyboard layout does. Again, importance is placed on the planning.

This was originally written for an earlier version of Ukelele, and has been somewhat adapted for the current version.

Introduction

Ukelele has been created for people who type in several languages, but do so on one keyboard. Ukelele is capable of producing accents on letters — grave, acute, circumflex, dieresis, tilde etc. — on ordinary (non function) keys. It does so making use of *dead keys*.

A dead key, usually representing an accent, does not respond immediately to our touch, hence its name. But as soon as we type its companion key, usually a vowel, both simultaneously appear: the accent together with the vowel.

Ukelele combines the accents grave, acute, circumflex and dieresis with all vowels. Furthermore, it combines the accent tilde with its own vowels plus some rare consonants. Such two stroke combinations of accents and vowels (plus rare consonants) will enable us to type texts in most non-English languages which use Roman script. Since the accented dead keys to be installed require space on our keyboard, we proceed carefully. Say we require the accents grave, acute, circumflex and dieresis. We place these four accented dead keys on the lower case of our keyboard layout. In doing so we have to “knock off” some key contents (symbols) that were there before. We have to find space for them elsewhere. We end up deciding which items of lesser importance we are ready to sacrifice.

Now, it happens to some of us that our fingers “look” automatically for the accustomed places of “their” accents, independently of the case we are typing in. Therefore we want to install our four accented dead keys not on one, but on all three positions (lower case, upper case and caps lock). This puts us into a bind. While the sacrifice of four key contents might be acceptable, a loss of twelve key spaces is not. What are we to do?

We shall sacrifice all *number keys* in the upper row of the lower and caps lock positions. Having the number pad at the right side of our keyboard, we don’t need numbers at its left. Thus we gain twenty free key spaces, which — after reallocating the earlier “knocked off” punctuation or symbols on new keys — leaves us with a bonus of eight free spaces. We may leave them empty or fill them with useful new content.

Now, this procedure might be acceptable to those among us who have a keypad with numbers. What about those who have none? A solution will be found for them for them also.

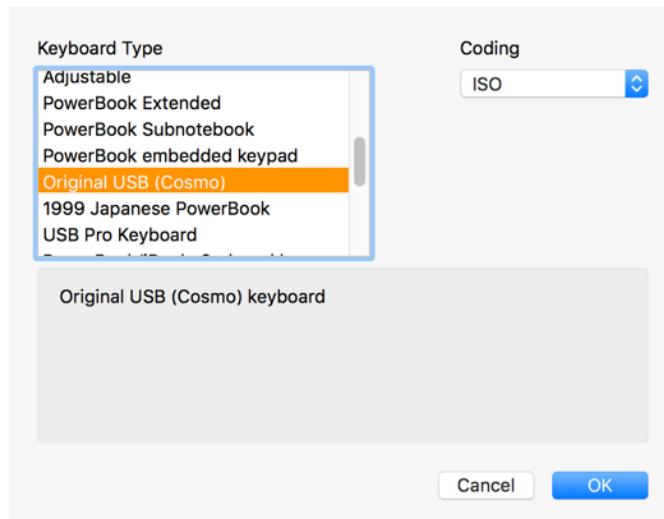
This is the introduction so far. It demonstrates the importance of *planning* the complete layout of your new keyboard on its three positions (lower case, upper case, caps lock). Inside the installer disk image of the Ukelele application we find an impressive choice of keyboards. Beginning with the one of our mother tongue, or with one we are familiar with, might be fine for some. Others may prefer to start with an ordinary QWERTY keyboard layout. Should we make errors in the planning stage, or later, we won't worry. Ukelele will give us ample opportunities to correct our mistakes.

Procedure

From the Internet we have downloaded the latest version of Ukelele and have put it into the Applications folder of our computer. We have launched Ukelele, and opened the Roman keyboard layout collection in the Standard Keyboards folder in the Resources folder on the installation disk image. We have created a new collection, and dragged the U.S. keyboard layout from the Roman collection to our new collection.

We intend to transform — on the levels lower case, upper case and caps lock — the keys of this US keylayout [] ; and ' into the dead keys ` `` ^ and `` . The keyboard position is lower case. We shall go through the following steps.

1. On the menu bar choose View → Keyboard Type → Select Original USB and ISO, then click OK.



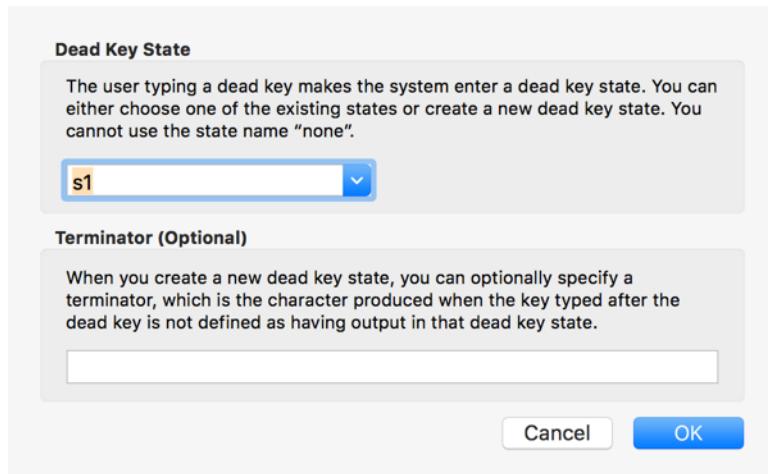
2. Choose Keyboard → Set Keyboard Name and Script. Set the name to "My US", ensure that the script is Unicode, and click OK.



3. We make sure we are still in the lower case and we press the [key.



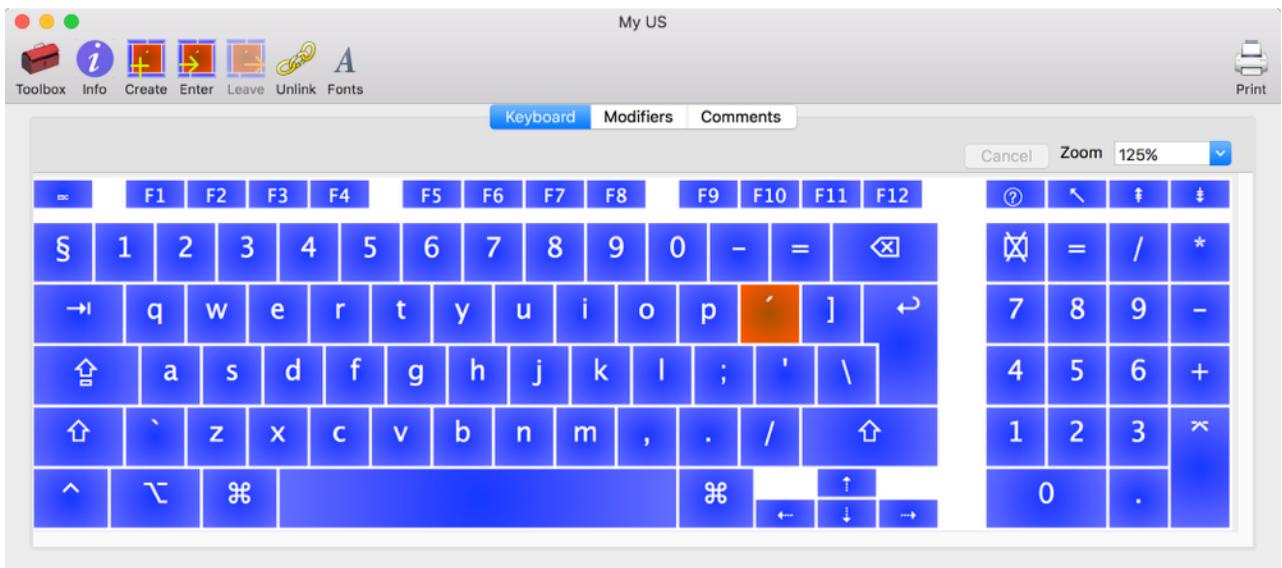
4. Choose Keyboard → Create dead key (or click the Create button on the toolbar).
5. In the dialog, we choose s1 (which stands for the acute accent) in the top part, leave the bottom part (the terminator) blank, and click OK.



6. We should now see the correct output (in this case the acute accent) on the vowels of the original keyboard, which remains in its lower case.



- Choose Keyboard → Leave Dead Key State (or click the Leave button on the toolbar). At this point, at the place of the former [key in the lower case of the original keyboard, we see the acute accent shaded in red. This means that the key we have just finished working on has become a dead key.



8. In order to complete our task we shall have to repeat the points 3 to 7 another eleven times: 3 times to do the rest of the lower case, 4 times to do the upper case and 4 times to do the caps lock. In step 5, the so-called “names” of the dead key states are the letter s followed by a number. Each number corresponds to an accent:

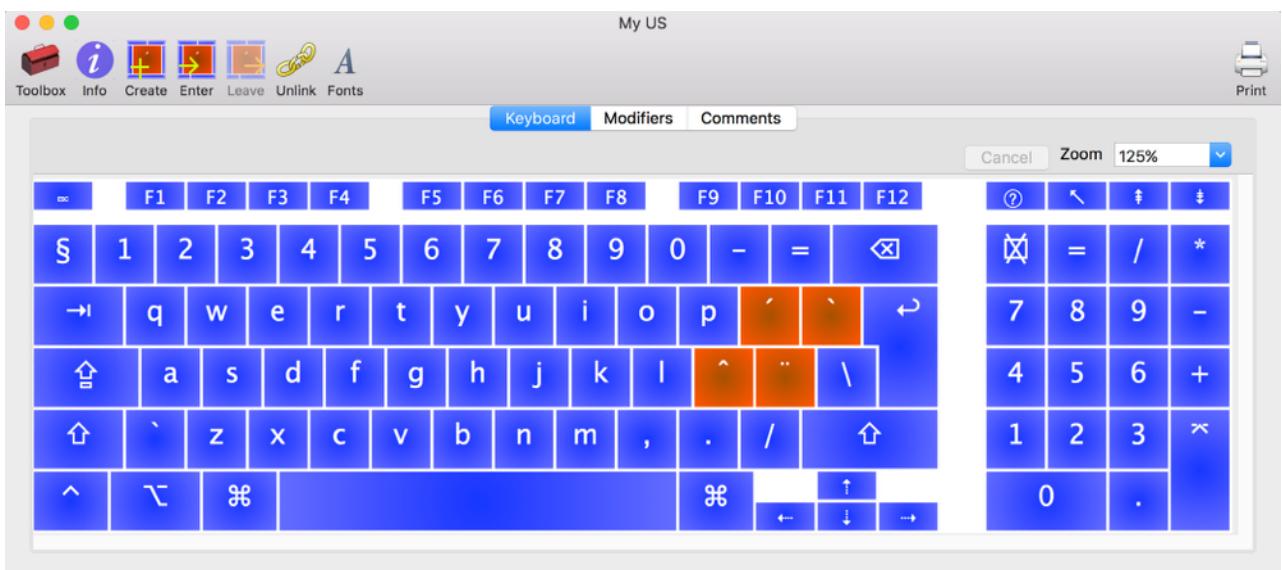
s1 → Acute

s2 → Grave

s3 → Circumflex

s4 → Dieresis

s5 → Tilde



9. We now have to restore the characters that we removed to make our dead keys. There are eight characters that we have lost: open and closing square brackets ([]), opening and closing curly brackets ({}), semicolon (;), colon (:), and single and double quotes (""). We will put these on the keys that originally had the digits 1 to 8. One way to do

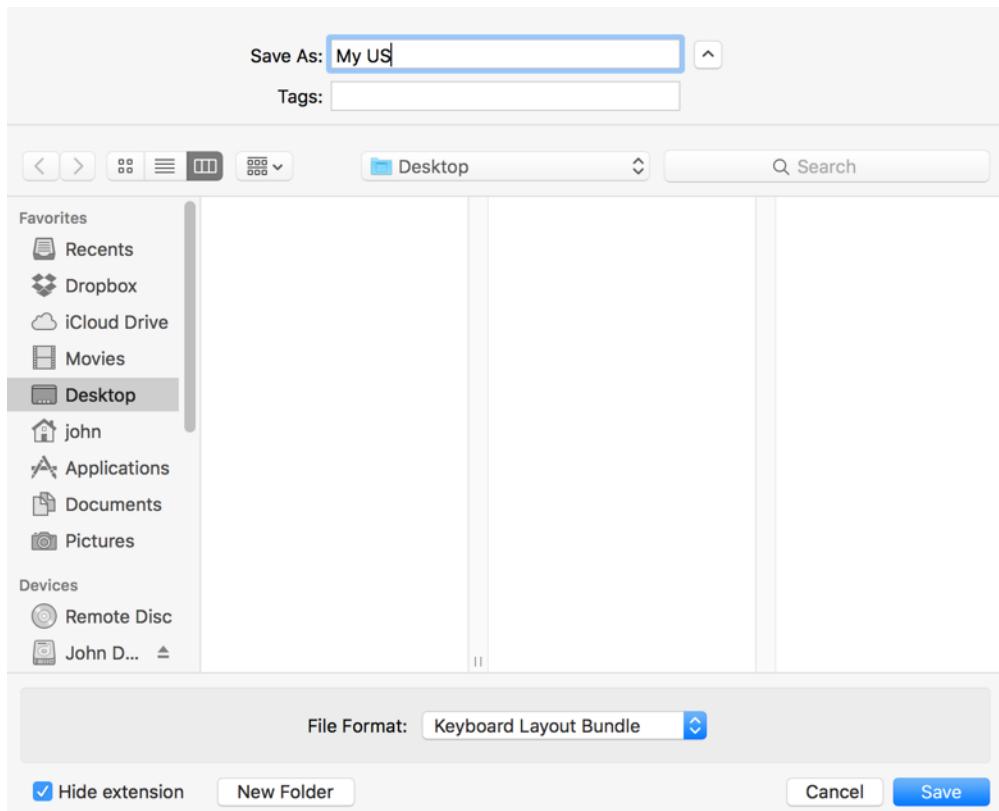
this is to double-click the “1” key, and then enter the opening square bracket as the new text in the popover that appears, and then repeat this for all the other characters.



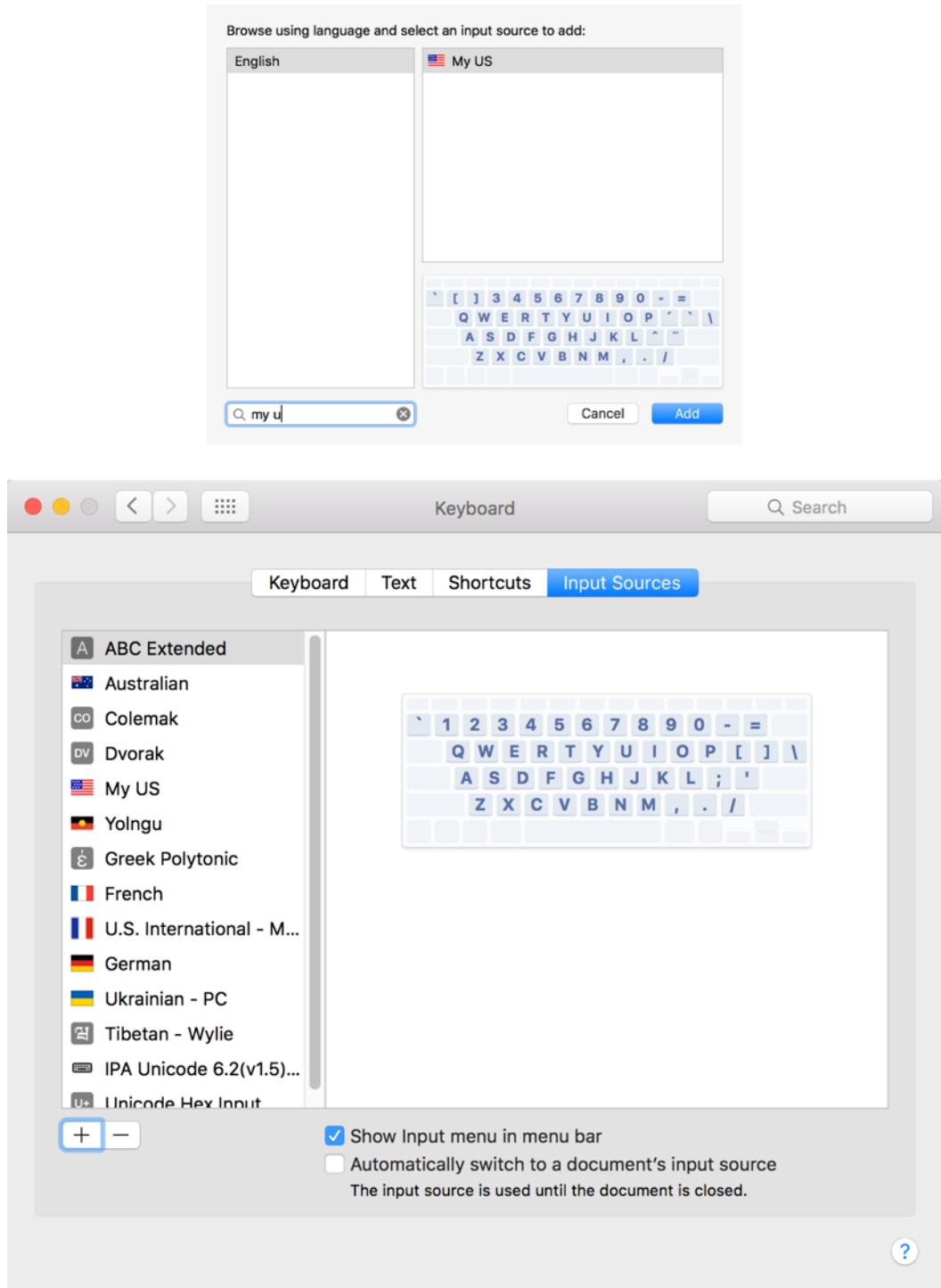
10. Alternatively, you can select the] character in some other program or the character palette, and then drag it onto the “2” key, and repeat with the other characters.



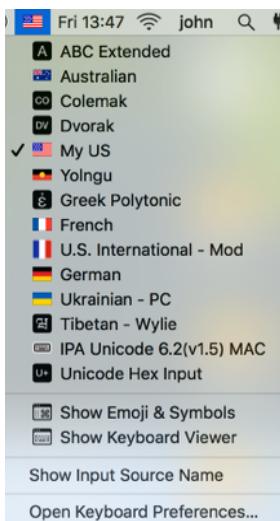
11. Now click on the keyboard collection window that we created in the first step. File → Save As → Give your keyboard a short name (ending with “.bundle” if the extension is not hidden), such as “My US”, and save it.



12. The keyboard layout needs to be installed in one of several locations. For most people, the best location is within your home directory. This can be accomplished within Ukelele by choosing File → Install → Install for Current User.
13. Now open System Preferences, and find the Keyboard pane. It has a stylised picture of a keyboard on it, and is usually up near the top. When the Keyboard pane is showing, click on the Input Sources tab. There should be a list of different keyboard layouts and other input methods that you have enabled (at least one would have been enabled when you installed Mac OS X). Below that list, click on the + button. Start typing “My US” into the search field, and you’ll see your new keyboard layout on the right. Click it and click OK. At the bottom of the preference pane, there is another check box labelled “Show Input menu in menu bar”. Check this one, too. You can now quit System Preferences.



14. There is now a flag icon towards the right hand end of the menu bar, which is the input menu. Click on this, and you should see your keyboard layout listed. Choose it from the menu. Your keyboard layout is now active, and can be used. Note that you may have to choose it again after you switch to another application. And that is it!



Comments, Questions and Answers

The Capacities of Ukelele

Theoretically, Ukelele is able to transform many ordinary (non control) keys, on all positions — lower case, upper case, caps lock, plus a few more — into dead keys. Of course such an undertaking would be senseless and counterproductive. Yet just the thought of it makes us realize the power of Ukelele.

Available Keyboards

Note that this is somewhat different in current versions of Ukelele and OS X.

Whenever we think of creating or personalizing a keyboard, we remember that the application Ukelele (Installer disk image → Resources → Standard Keyboards → Roman) contains some eighty ready made keyboards suitable for Western use. Another choice of keyboard layouts becomes available to us by clicking the colored flag next to the blue Q-Spotlight symbol at the right upper corner of the screen → Open Keyboard Preferences. Locating the keyboard which is close to what we want at the outset will save us effort and time.

Note that some of this is different if you are using Mac OS X 10.5 (Leopard). The Spotlight symbol is not blue, and the keyboards supplied with the system are not available for Ukelele to use.

The Character Palette

Note that this is somewhat different in current versions of Ukelele and OS X. In OS X 10.11 El Capitan, it is now found as Show Emoji & Symbols.

Ukelele and the Character Palette work as a team. The Palette becomes available to us by clicking the colored flag next to the blue Q-Spotlight symbol at the right upper corner of the screen → Show Character Palette. Its choice of unaccented and accented letters, numbers punctuation and other symbols is impressive. By dragging and dropping one of its items on an ordinary (non function) key of our keyboard-to-be, we “knock off” the letter, number, punctuation etc. which has been there before. We remember, after we have done with our key substitutions, to logout and login in order to have the keyboard recognized by the System.

Why have we neglected the accent tilde?

Without the tilde we cannot type Hispanic and Portuguese languages. If this is what we want, we include the tilde in the program of our dead keys. In our comments on Procedure we have not only “neglected” the tilde, but numerous other accent-plus-letter combinations, even letter-plus-letter combinations Ukelele can produce. We have limited ourselves to the use of Ukelele in typing Western languages in Roman print.

By sacrificing our numbers, did we do the right thing?

Indeed, some programs do require the numbers at the left side of our keyboard. However, this does not disturb us, since a simple keystroke (menu bar, flag next to the blue Q at the upper right corner of the screen) makes one or several other non Ukelele keyboards equipped with number pads available to us. Once we go back toTextEdit or Word we return to our Ukelele keyboard.

What if we have no number pads?

The solution proposed here will satisfy those among us who require numbers sometimes. Those who do it often will have their number pads anyway. Here is what we are going to do. After lower case, upper case and cap locks we are going to open up one more position. We press the Option Key of our keyboard-to-be. Here we are confronted with a keyboard which is filled with mostly useless items. We leave a finger of our left hand on the Option Key. With the right hand we drag and drop, from the numbers contained in the Character Palette, 1234567890 onto the top row of our new Option Key level. Perfectionists among us may repeat the same procedure keeping the option key simultaneously with the lower case key (even the upper case key) pressed.

Henceforth our ten digits will be available to us by pressing the Option Key.

Erasing

When we drag and drop something from the Character Palette on an ordinary non control key, the new key content takes the place of the old one. Occasionally, however, we have to erase a given key content, dead key or not. Some key contents may even show up at places were we expect them least; for instance on the space bar. If this happens, we double-click the unwanted symbol. A Dialog box says There is a dead key state with this key combination → There is a dead key state combination... → delete → change the output of this key combination - delete - OK. If necessary we repeat the last step.

Ukelele Errors

The example of an unwanted symbol on the space bar may make us suspect Ukelele of an erratic behavior. At this point we shall remember that even the most expensive programs, the very pillars of computerdom, behave at times strangely. Over the years, the author of Ukelele has ironed out a number of bugs. In the rare instances Ukelele insists on misbehaving, we move the application into the Trash. On our hard disk we run a version of “Disk Warrior” or “Norton” that is compatible with our System. We reinstall Ukelele (of which we have kept a copy at a safe place) and we start anew.

A Few No-Nos

We don't try to use the usual Copy and Paste because they will not work. We never make Ukelele intervene in the area of control keys and in that of the number pad at the right side of most keyboards. We only work on ordinary keys. If, by error, we disturbed a control key or a key of the number pad, we use undo to go back to before the error.

Once we have completed

Even after a careful planning and executing our keylayout we might discover — after having put it through its paces — that it needs some retouching. If this happens, we take the new Ukelele keyboard out of the System and make the necessary adjustments. After having put it back again we won't forget to logout and to login so as to have the keyboard recognized by the System.

Bridging the difference between the electronic and the physical keyboard

The electronic keyboard is the one we see on the screen; the physical keyboard is the one we put our fingers on. While Ukelele changes the electronic keyboard, the physical keyboard remains unchanged. The way of bridging this disparity will depend on us. “Hunt-and-peck” typists may glue stickers (with the appropriate writing) onto the keys of their physical keyboard whose contents have been altered. Others will place the small representation of their electronic keyboard (“Visor”?) permanently next to theTextEdit or Word page they are typing. Others again may use both methods simultaneously.

Part 3

More Advanced Topics

Multi-level dead keys, modifiers and unlinking keys

In this part of the tutorial, we will create a keyboard layout for a hypothetical language, which will demonstrate the use of two levels of dead key and editing modifier combinations, and also show a case when you would use the “unlink key” command.

Designing the Keyboard Layout

Our hypothetical language has five vowels, but has three tones (high, mid and low), represented by tone marks, with á for high tone, à for low tone, and mid tone unmarked. It also has long vowels, represented by a macron, so that ā is a long vowel. The important feature is that vowels can be marked for both tone and length, so you can get symbols such as é or ù in the language.

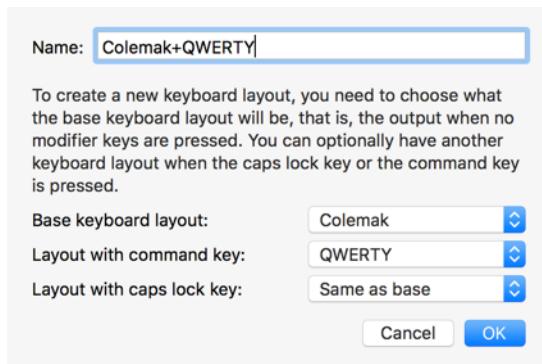
We will also make the keyboard layout based on the Colemak layout, while also using the QWERTY layout when the command key is down, so that keyboard shortcuts work with what is printed on the keys rather than the characters generated by our keyboard layout.

The keyboard layout will use three dead keys: `e will add the high tone mark, `a the low tone mark, and `m the macron. We will make it work whichever order the dead keys are typed, so that you can do `e `m e to get é, and also `m `e e to get the same character.

Getting Started

For the purposes of this exercise, we will begin with an empty keyboard layout collection. So, launch Ukelele, select File → New Keyboard Layout Collection (or type ⌘N).

We want to create a new keyboard layout which uses the Colemak layout as its basis, but with QWERTY as the key map when command is pressed. To do this, click on the + button at the lower left, and choose Standard Keyboard Layout from the menu. Next, select Colemak as the base keyboard layout, QWERTY as the keyboard layout with command pressed, and Same as base for the keyboard layout with caps lock pressed. Give it a name, Colemak+QWERTY, and click OK.



Let's save this collection while we're at this early point, again using Colemak+QWERTY as the name.

Now, double-click to open the keyboard window.



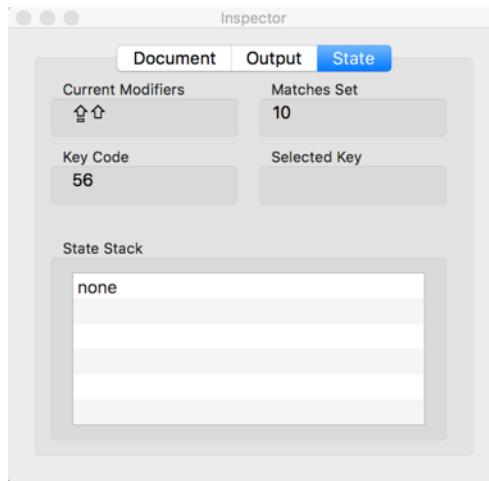
You'll see the Colemak layout here. If you're unfamiliar with it, it's a redesign of an English language keyboard that is designed to be ergonomic and easy to learn. If you experiment with different modifier combinations, you'll see that you get the Colemak layout with no modifiers, with shift, or with caps lock, but the QWERTY layout with any combination with the command key.

What may be unexpected is that shift plus caps lock has an empty layout. Why is that? The answer can be seen by shifting to the modifiers tab.

Index	Option	Shift	Command	Control	Caps Lock
0	✗	▲	⌘	△	◆
1	✗	▲	⌘	△	◆
2	✗	▲	⌘	△	◆
3	✗	▲	⌘	△	◆
4	✗	▲	⌘	△	◆
5	✗	▲	⌘	△	◆
6	✗	▲	⌘	△	◆
7	✗	▲	⌘	△	◆
8	✗	▲	⌘	△	◆
9	✗	▲	⌘	△	◆
10	✗	▲	⌘	△	◆

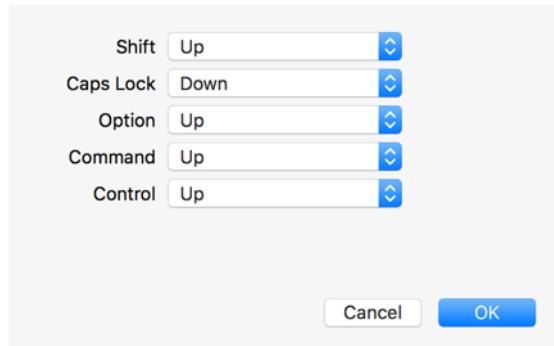
Default Index 10

If you try to find a match for caps lock plus shift, you won't find one. Row 3 has caps lock optional and shift down, but also has option down. Row 4 has caps lock, but not shift, and row 9 has caps lock and command, but not shift. So, what does it match? That will be the default index, which you can see is 10. You can also see this in the info window:

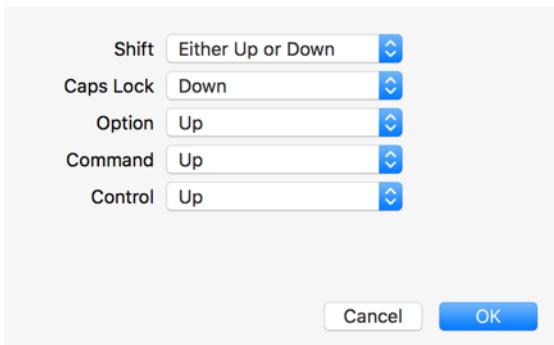


Notice that the current modifiers are caps lock and shift, and they match set 10, which is what we worked out above.

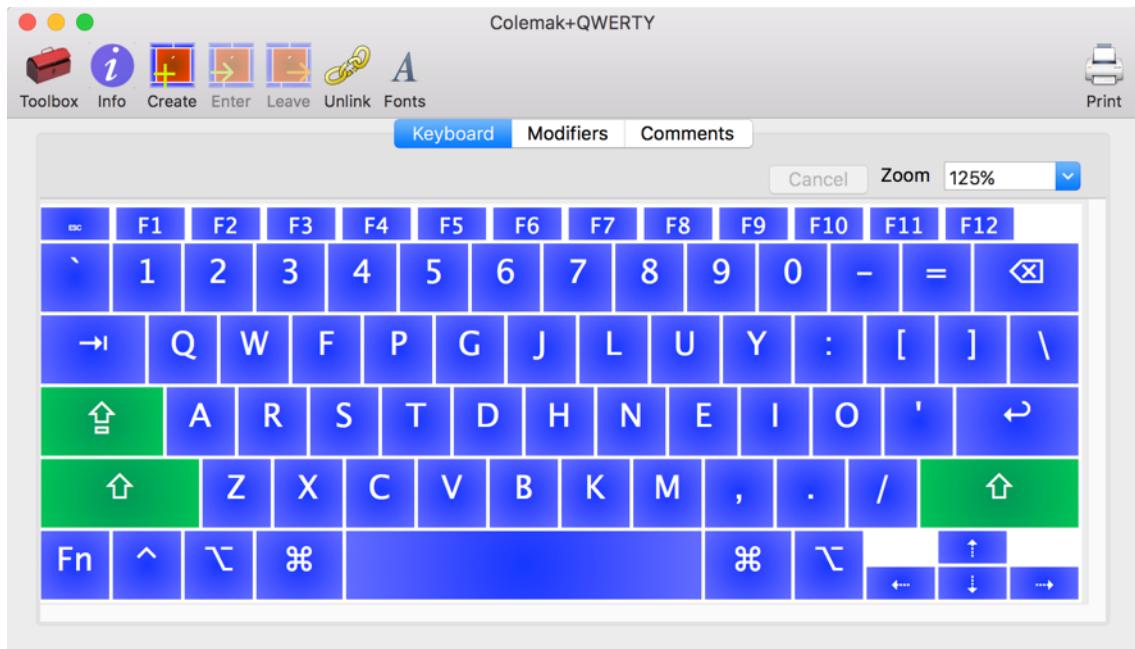
If we want shift and caps lock to be the same as caps lock, then we need to edit the modifier combinations. We double-click row 4, which has caps lock, and get the modifiers editor:



We want to change this to having shift being either up or down:



After we click OK, we will see that we get what we expected in the first place:



We could also make the same change for the command plus shift plus caps lock case (⇪⇪⌘):

Index	Option	Shift	Command	Control	Caps Lock
0	⇪	⇪	⌘	△	⇪
1	⇪	⇪	⌘	△	⇪
2	⇪	⇪	⌘	△	⇪
3	⇪	⇪	⌘	△	⇪
4	⇪	⇪	⌘	△	⇪
5	⇪	⇪	⌘	△	⇪
6	⇪	⇪	⌘	△	⇪
7	⇪	⇪	⌘	△	⇪
8	⇪	⇪	⌘	△	⇪
9	⇪	⇪	⌘	△	⇪
10	⇪	⇪	⌘	△	⇪



We have the basics sorted out, and now we'll start to work on the dead keys.

Creating the Dead Keys

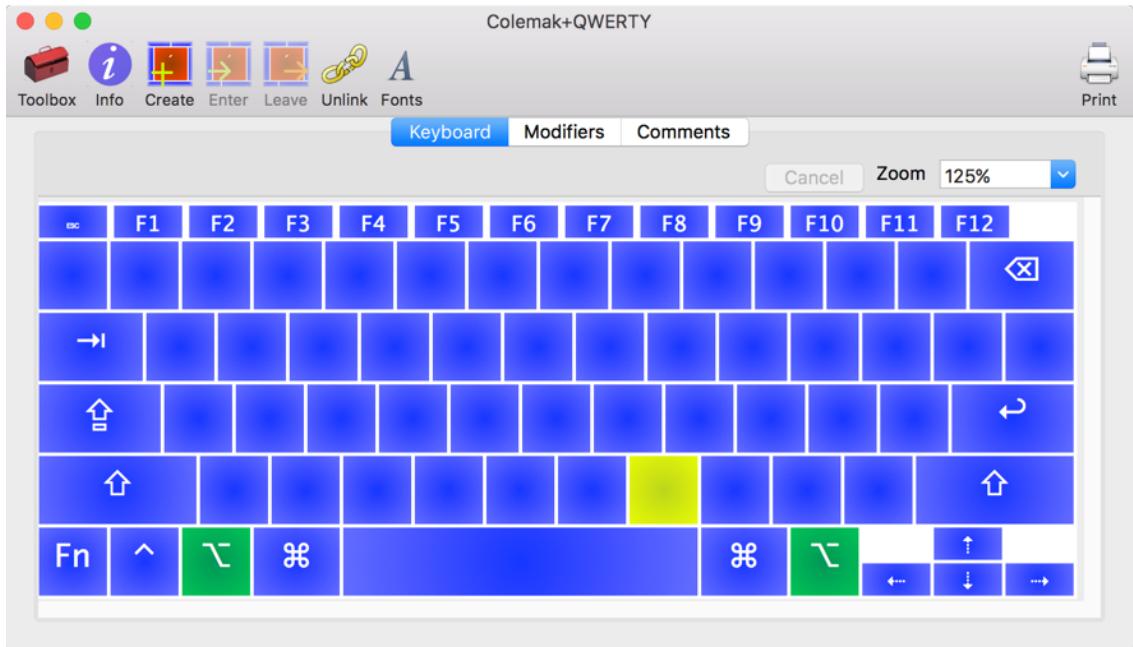
We will need to create three dead keys to satisfy the requirements we set above. One will add a macron to vowels, a second will add a high tone mark (acute accent) to vowels, and the third will add both.

The first dead key

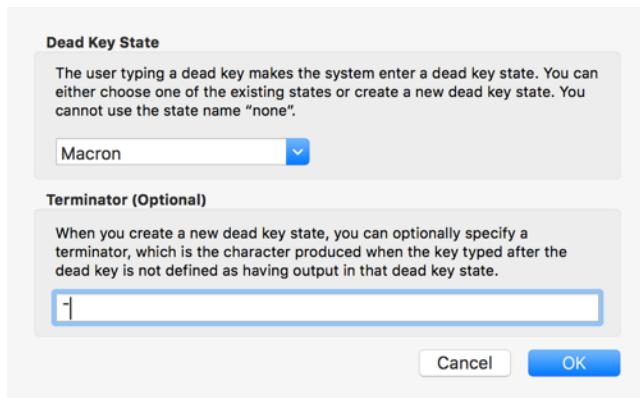
We will create the dead key that adds the macron to vowels. The actual dead key will be `⌥m`, which currently does nothing. The first thing to do is to turn on Sticky Modifiers so that we don't need to worry about missing the option key on the way. This can be done with the Toolbox, shown by clicking the Toolbox button on the toolbar (or choosing Show Toolbox from the View menu):



Next, we'll press option once to make it active, and then the m key (which is the same on both Colemak and QWERTY layouts):



Next we click on the Create button on the toolbar to get the dialog. We will make it a new dead key state (which it has to be, since there are no others at present), with the name Macron, and the terminator $\bar{ }$ (spacing modifier macron, U+00AF):



Now press the option key again (or click it on screen) to make it inactive. You'll see that most of the keys are greyed out, showing that they have no output in the current state, but showing the output when no dead key is active, to make it easier to pick the right key to change:



We want to add the new output, so we double-click on the u key, and put ū as the new output:



Continue doing this for a, e, i and o, to get this:



Now press shift (or click on screen) to activate it, and repeat for the uppercase vowels:

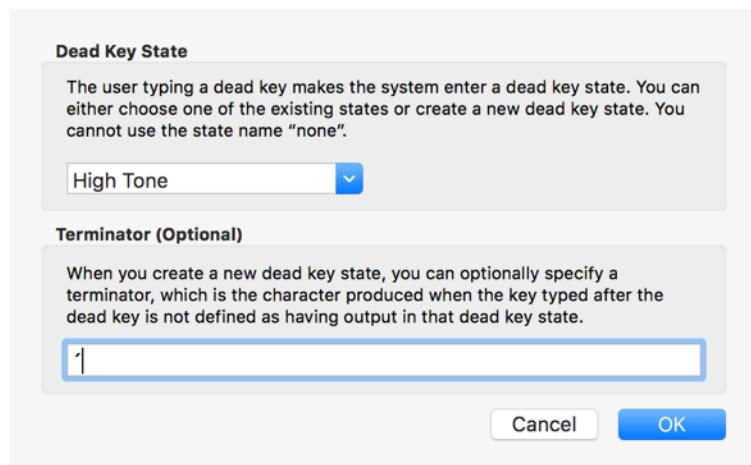


This part is complete, so we can leave this dead key state by clicking the Leave icon on the toolbar (or choosing Leave Dead Key State from the Keyboard menu).

The second dead key

Next, we will create the dead key for adding the high tone mark (acute accent) to vowels. This will be Āe . (Note that e in the Colemak layout is where k is in the QWERTY layout.) So, we need to be sure that we are in the state “none”. This can be verified by seeing that the Leave button on the toolbar is disabled, or by inspecting the state stack in the inspector.

Once again, make the option key active, and press e (the k key if you have a QWERTY layout), then click the Create button on the toolbar (or choose Create Dead Key... from the Keyboard menu). Name the new dead key state High Tone, and make the terminator ' (spacing acute accent, U+00B4):



Now press option again to deactivate it, and start entering the new output, ú for u, á for a, and so on, both in lowercase and uppercase.

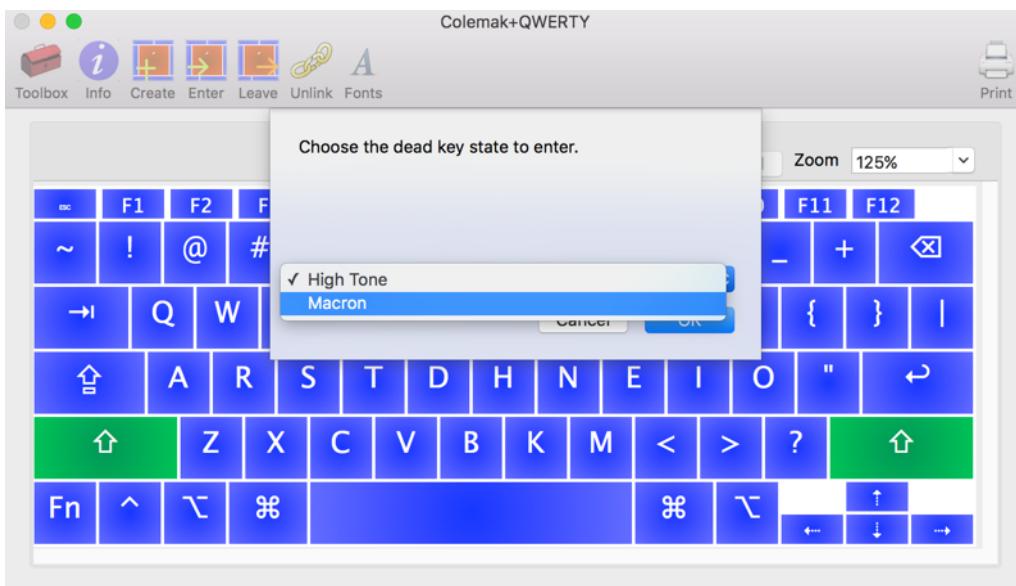


When you are done, click Leave, or choose Leave Dead Key State from the Keyboard menu to finish.

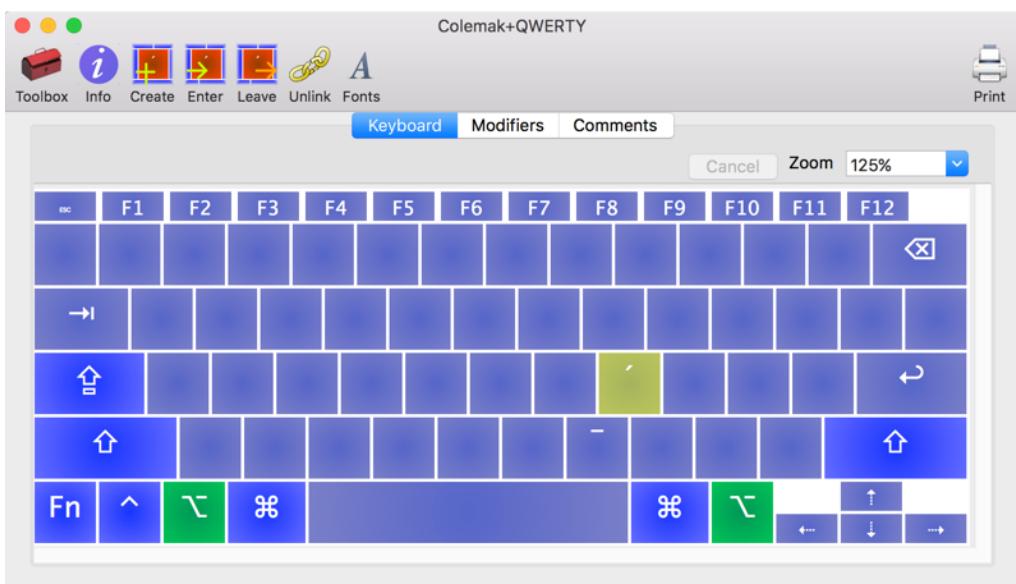
The third dead key

With the first two dead keys finished, we are ready to create the third, which has both the macron and the high tone. The idea is that we can type ̀e ́m to get it, and also ́m ̀e , that is, the two earlier dead keys in succession, in either order. So, this is going to require two stages. First, we create the dead key with one order, and then we link it to the second order.

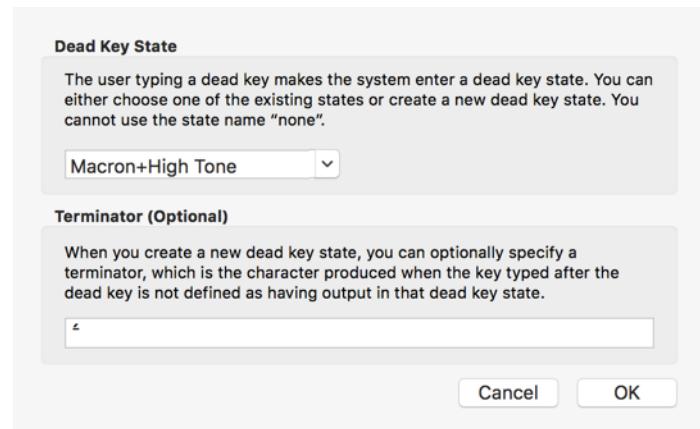
To start with, click Enter on the toolbar (or choose Enter Dead Key State... from the Keyboard menu, or press ⌘E), then choose Macron from the popup list.



Now we want to have the modifiers as just option, so we can do it either with Sticky Modifiers on or by pressing the option key while typing the e or clicking the e key on screen while Sticky Modifiers is off:



Next click on the Create button on the toolbar, and enter a new state name, Macron+High Tone, and set the terminator to ' (spacing macron, U+00AF, plus combining acute, U+0301):

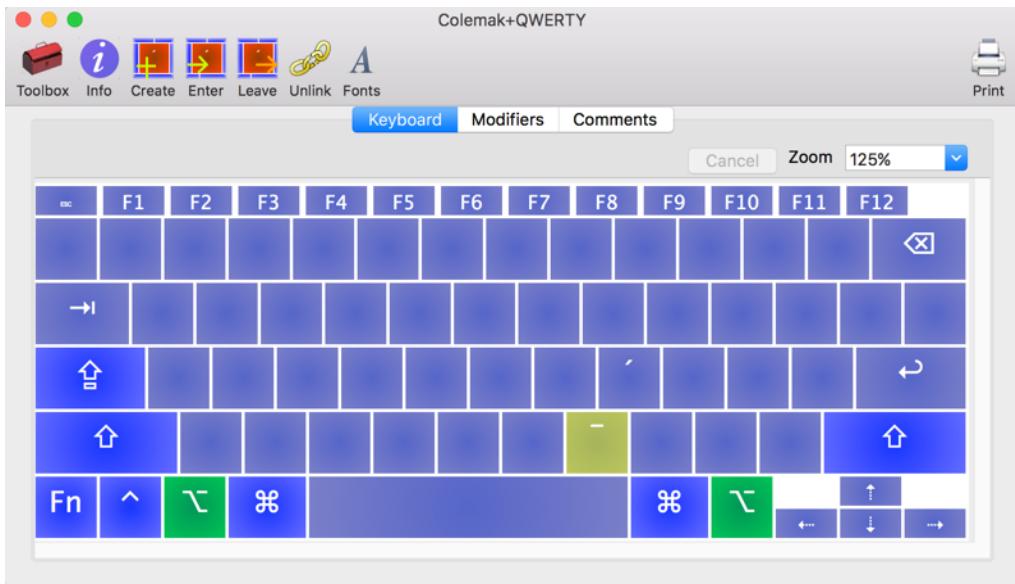


Now we add the output we want: á on the a key, ú on the u key, etc, and also in uppercase:

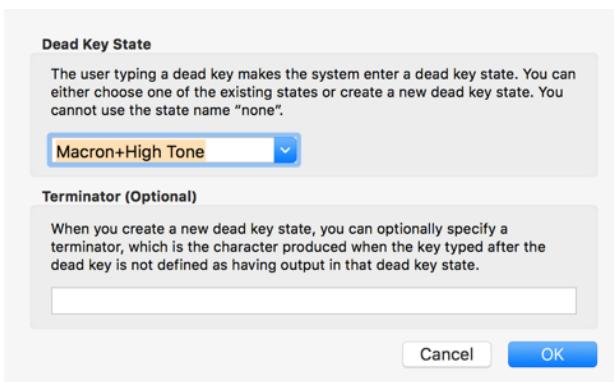


One point to note is that these combinations don't correspond to single Unicode code points, but to a sequence of a Unicode code point (e.g. U+0101 for á) plus a combining accent (U+0301), and you may have different results with different fonts and applications, as to whether the stacked diacritics display correctly.

Now click Leave twice to go back to the base state. Next, click Enter, and choose High Tone. With option active, press or click on m:



Now click on Create. We have already created the state we need, Macron+High Tone, so we select that from the combo button at the top of the dialog:



Once you click OK, you'll see that we're in the correct state again:



We can now click Leave twice to return to the base state, and we're done with creating the first three dead keys.

The fourth and fifth dead keys

The remaining dead keys are to have the low tone mark and the low tone with the macron. We need to create two more dead key states, one we'll call Low Tone, with terminator ` (spacing grave accent, U+0060), and one we'll call Low Tone+Macron, with terminator ^ (spacing macron, U+00AF, combining grave accent, U+0300). We'll make ` be the dead key for Low Tone, and then we'll need to add sequences for ` `m and the reverse, `m `.

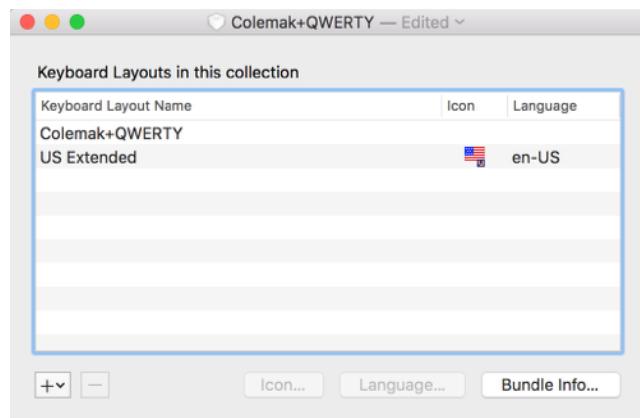
The procedure is analogous to those we've done before, so we won't spell it all out.

Once we've done this, the keyboard layout is basically complete. You could add more output for the option keys if you wished, since very little is defined, but the basics of the keyboard layout are done.

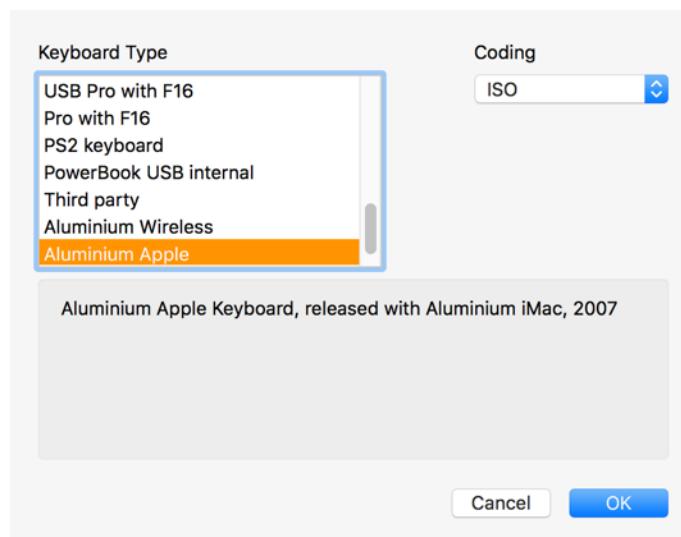
Unlinking Keys

This topic can't actually be demonstrated with the keyboard layout we've just produced, so we're going to do it with a different keyboard layout. We'll make a copy of an existing keyboard layout, and make some modifications.

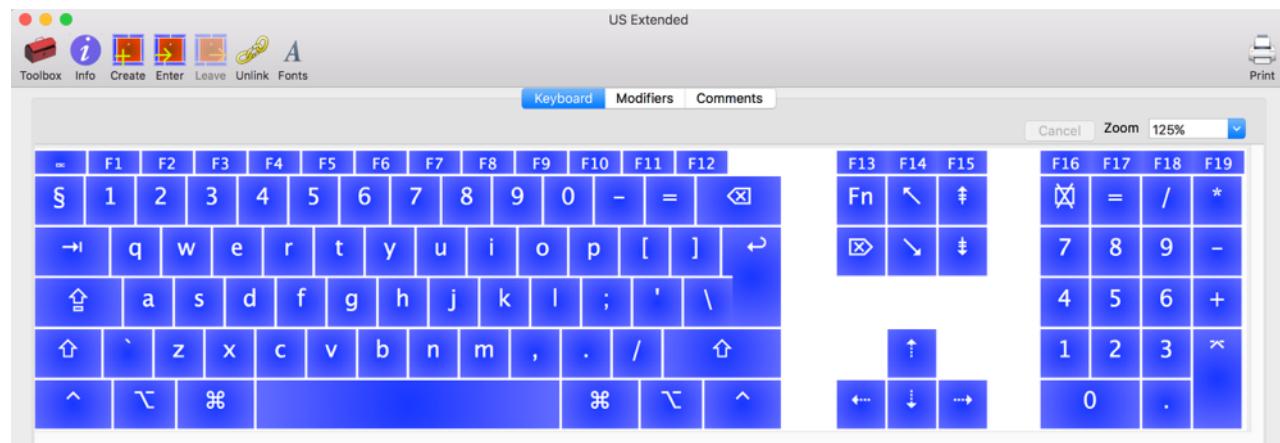
To begin with, open the Unicode keyboard layout collection, and drag the US Extended layout from there into the keyboard layout collection we created above.



Now double-click our new copy of US Extended to open it. We're going to make some changes to the numeric keypad, so we need to pick a keyboard type that has a numeric keypad so that we can see it. From the View menu, choose Keyboard Type... (or press ⌘K), and choose Aluminium Apple from the list on the left, and ISO from the Coding popup button:



This will make our keyboard window appear as follows:



Note that there's actually an error in Apple's geometry files that makes the \ and return keys appear to touch, and similarly for F13 through F15 and F16 through F19. These are cosmetic errors only, and do not affect the functioning of the keyboard layout.

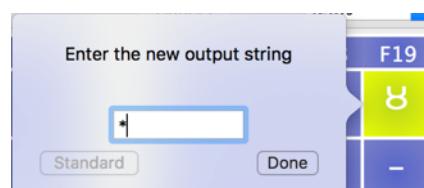
The change we're going to make is rather artificial, but demonstrates the problem of linking. If you enter the dead key state "specials" by clicking the Enter button on the toolbar and choosing "specials" from the popup button, you'll see that none of the keypad keys has any output, except the * key, which produces Ȣ (U+0222, Latin letter OU):



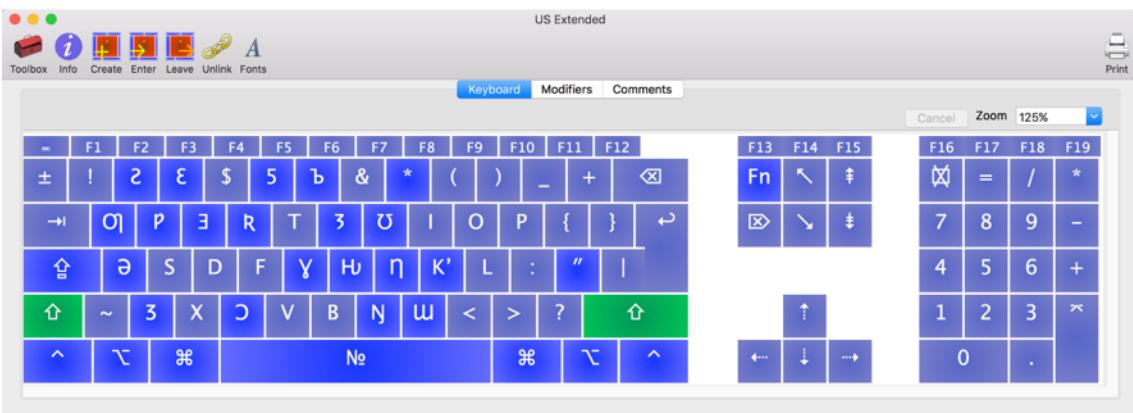
If you press shift, you will see that shift-8 also produces the same output:



We're going to make the * key on the number pad produce *. The obvious way would be simply to edit the output:



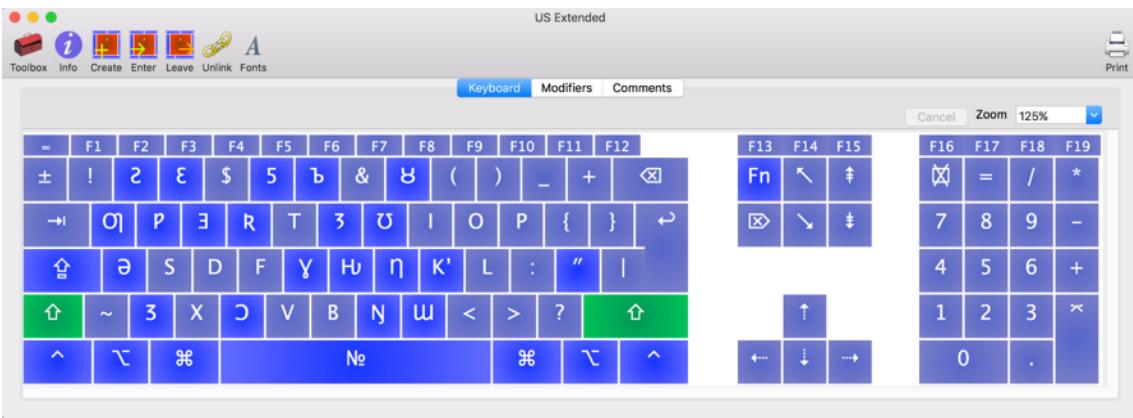
However, what actually happens is that we have changed both that key and the shift-8 output:



That's not what we want, and it's because the two keys are linked. So undo that change, and select the * key on the number pad:



And now click the Unlink button on the toolbar or choose Unlink Key... from the Keyboard menu. Nothing visible happens apart from the key being unselected. However, if you change the output now, you'll see that it is now independent of the shift-8 output:



Actually, we still haven't really got what we want. We don't actually want the * key to produce any specific output in the specials state, since no other key on the number pad does. We still need to have it unlinked, but instead of changing the output to *, we simply delete the output when we change it:



Once we've done that, the keyboard layout should look like this:



And that's how we want it, so we are done.

That completes the tutorial. There are a comprehensive manual and online help from the Help menu. Further support can be found from the Ukelele Users group on Google Groups, which can also be accessed from the Help menu.