

Picker Views

On This Page

A picker view lets the user choose between certain options by spinning a wheel on the screen. Picker views are well suited for choosing things like dates and times (as the date picker does) that have a moderate number of discrete options. Other examples include picking which armor to wear in a game and picking a font for text in a word processor. The list of options should be ordered in some logical way to make scanning easier.

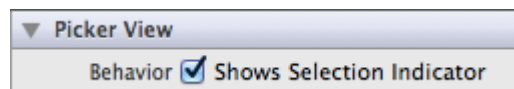


Purpose. Picker views allow users to:

- Quickly choose between a set of distinct options

Implementation. Picker views are implemented in the [UIPickerView](#) class and discussed in the [UIPickerView Class Reference](#).

Configuration. Configure picker views in Interface Builder, in the Picker View section of the Attributes Inspector. A few configurations cannot be made through the Attributes Inspector, so you must make them programmatically. You can set other configurations programmatically, too, if you prefer.



Content of Picker Views (Programmatic)

Populating a picker requires both a data source and a delegate. It is not possible to populate a picker in Interface Builder; you must do this programmatically.

On This Page

Picker views need a [delegate](#) to display data and appropriately handle user interaction. The delegate adopts the [UIPickerViewDelegate](#) protocol and provides the content for each component's row, either as an attributed string, a plain string, or a view, and it typically responds to new selections or deselections. It also implements methods to return the drawing rectangle for rows in each component—these optional methods are only needed if the delegate returns a view as part of the picker's content.

Additionally, picker views require a data source. The data source adopts the [UIPickerViewDataSource](#) protocol and implements the required methods to return the number of components (columns) and the number of rows in each component. Note that the actual contents of each row comes from the delegate, not the data source.

For information about delegates and data sources, see [Delegates and Data Sources](#).

After setting the data source and delegate, set the initial selection by calling the [selectRow:inComponent:animated:](#) without animation. Typically this is done in the [viewDidLoad](#) method of the view's view controller.

If the picker is visible, use animation when you update the selection.

You can dynamically change the rows of a component by calling the [reloadComponent:](#) method, or dynamically change the rows of all components by calling the [reloadAllComponents](#) method. When you call either of these methods, the picker view asks the delegate for new component and row data, and asks the data source for new component and row counts. Reload a picker view when a selected value in one component should change the set of values in another component. For example, changing a row value from February to March in one component should change a related component representing the days of the month.

Behavior of Picker Views

You cannot customize the picker view's selection indicator on iOS 7. The selection indicator is always shown, so toggling the Shows Selection Indicator ([showsSelectionIndicator](#)) box has no effect.

Appearance

On This Page

On iOS 7, you can customize the background color of the picker view by changing its `backgroundColor` property.

Using Auto Layout with Picker Views

You can create an Auto Layout constraint between a picker view and other user interface elements. You can create any type of constraint for a picker view besides a baseline constraint.

Picker views usually reside at the bottom of the screen in all device orientations. Select Bottom Space to Superview and set the relation equal to 0 for the date picker to pin to the bottom of the screen in all device orientations.

For general information about using Auto Layout with iOS views, see [Using Auto Layout with Views](#).

Making Picker Views Accessible

Picker views are accessible by default. The default accessibility trait for a picker view is Adjustable.

For general information about making iOS views accessible, see [Making Views Accessible](#).

Internationalizing Picker Views

To internationalize picker view, you must provide localized translations of each string in the picker.

For more information, see [Internationalization and Localization Guide](#).

Debugging Picker Views

When debugging issues with picker views, watch for this common pitfall:

Not testing loca

On This Page

Elements Similar to a Picker View

The following elements provide similar functionality to a picker view:

- **Date Picker.** Uses a picker to let the user pick a date and time. For more information, see [Date Pickers](#).
- **Stepper.** Lets the user increment and decrement a value. For more information, see [Steppers](#).