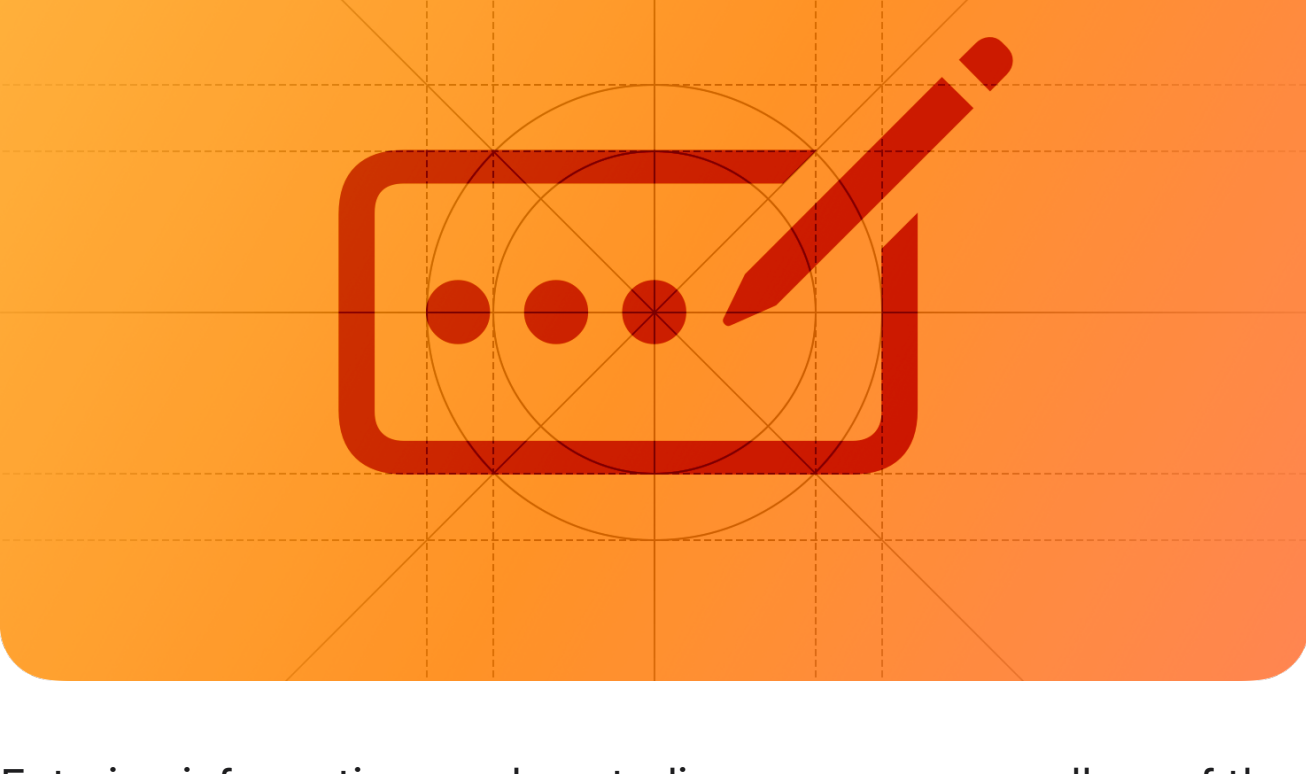


# Entering data

When you need information from people, design ways that make it easy for them to provide it without making mistakes.



Entering information can be a tedious process regardless of the interaction methods people use. Improve the experience by:

- Pre-gathering as much information as possible to minimize the amount of data that people need to supply
- Supporting all available input methods so people can choose the method that works for them

## Best practices

**Get information from the system whenever possible.** Don't ask people to enter information that you can gather automatically — such as from settings — or by getting their permission, such as their location or calendar information.

**Be clear about the data you need.** For example, you might display a prompt in a text field — like “username@company.com” — or provide an introductory label that describes the information, like “Email.” You can also prefill fields with reasonable default values, which can minimize decision making and speed data entry.

**Use a secure text-entry field when appropriate.** If your app or game needs sensitive data, use a field that obscures people's input as they enter it, typically by displaying a small filled circle symbol for each character. For developer guidance, see [SecureField](#). In tvOS, you can also configure a [digit entry view](#) to obscure the numerals people enter (for developer guidance, see [isSecureDigitEntry](#)). When you use the system-provided text field in visionOS, the system shows the entered data to the wearer, but not to anyone else; for example, a secure text field automatically blurs when people use AirPlay to stream their content.

**Never prepopulate a password field.** Always ask people to enter their password or use biometric or keychain authentication. For guidance, see [Managing accounts](#).

**When possible, offer choices instead of requiring text entry.** It's usually easier and more efficient to choose from lists of options than to type information, even when a keyboard is conveniently available. When it makes sense, consider using a picker, menu, or other selection component to give people an easy way to provide the information you need.

**As much as possible, let people provide data by dragging and dropping it or by pasting it.** Supporting these interactions can ease data entry and make your experience feel more integrated with the rest of the system.

**Dynamically validate field values.** People can get frustrated when they have to go back and correct mistakes after filling out a lengthy form. When you verify values as soon as people enter them — and provide feedback as soon as you detect a problem — you give them the opportunity to correct errors right away. For numeric data in particular, consider using a number formatter, which automatically configures a text field to accept only numeric values. You can also configure a formatter to display the value in a specific way, such as with a certain number of decimal places, as a percentage, or as currency.

**When data entry is necessary, make sure people understand that they must provide the required data before they can proceed.** For example, if you include a Next or Continue button after a set of text fields, make the button available only after people enter the data you require.

## Platform considerations

*No additional considerations for iOS, iPadOS, tvOS, visionOS, or watchOS.*

## macOS

**Consider using an expansion tooltip to show the full version of clipped or truncated text in a field.** An *expansion tooltip* behaves like a regular tooltip, appearing when the pointer rests on top of a field. Apps running in macOS — including iOS and iPadOS apps running on a Mac — can use an expansion tooltip to help people view the complete data they entered when a text field is too small to display it. For guidance, see [Offering help > macOS, visionOS](#).

## Resources

### Related

- [Text fields](#)
- [Virtual keyboards](#)
- [Keyboards](#)

### Developer documentation

[Input events](#) — SwiftUI

### Videos



What's new in UIKit

## Change log

Date	Changes
June 21, 2023	Updated to include guidance for visionOS.