

# Language Revision History

The following identifies the set of Swift Evolution (SE) proposals that have been implemented in each release of Swift since the language became open source, with each providing a link to the corresponding SE content on the [swift.org](https://swift.org) website.

## Swift 5.9.2

- SE-0407: [Member Macro Conformances](#)

## Swift 5.9

- SE-0402: [Generalize conformance macros as extension macros](#)
- SE-0401: [Remove Actor Isolation Inference caused by Property Wrappers](#)
- SE-0400: [Init Accessors](#)
- SE-0399: [Tuple of value pack expansion](#)
- SE-0398: [Allow Generic Types to Abstract Over Packs](#)
- SE-0397: [Freestanding Declaration Macros](#)
- SE-0396: [Conform Never to Codable](#)
- SE-0395: [Observation](#)
- SE-0394: [Package Manager Support for Custom Macros](#)
- SE-0393: [Value and Type Parameter Packs](#)
- SE-0392: [Custom Actor Executors](#)
- SE-0391: [Package Registry Publish](#)
- SE-0390: [Noncopyable structs and enums](#)
- SE-0389: [Attached Macros](#)
- SE-0388: [Convenience Async\[Throwing\]Stream.makeStream Methods](#)
- SE-0386: [New access modifier: package](#)
- SE-0384: [Importing Forward Declared Objective-C Interfaces and Protocols](#)
- SE-0382: [Expression Macros](#)
- SE-0381: [DiscardingTaskGroups](#)
- SE-0380: [if and switch Expressions](#)
- SE-0377: [borrowing and consuming Parameter Ownership Modifiers](#)
- SE-0374: [Add sleep\(for:\) to Clock](#)
- SE-0366: [consume Operator to End the Lifetime of a Variable Binding](#)

## Swift 5.8

- SE-0378: [Package Registry Authentication](#)
- SE-0376: [Function Back Deployment](#)
- SE-0375: [Opening existential arguments to optional parameters](#)
- SE-0373: [Lift all limitations on variables in result builders](#)
- SE-0372: [Document Sorting as Stable](#)
- SE-0370: [Pointer Family Initialization Improvements and Better Buffer Slices](#)
- SE-0369: [Add CustomDebugStringConvertible conformance to AnyKeyPath](#)
- SE-0368: [StaticBight](#)
- SE-0367: [Conditional compilation for attributes](#)

## Language Revision History

- SE-0365: [Allow implicit self for weak self captures, after self is unwrapped](#)
- SE-0362: [Piecemeal adoption of upcoming language improvements](#)
- SE-0274: [Concise magic file names](#)

### Swift 5.7

- SE-0363: [Unicode for String Processing](#)
- SE-0361: [Opaque result types with limited availability](#)
- SE-0358: [Primary Associated Types in the Standard Library](#)
- SE-0357: [Regex-powered string processing algorithms](#)
- SE-0356: [Swift Snippets](#)
- SE-0355: [Regex Syntax and Run-time Construction](#)
- SE-0354: [Regex Literals](#)
- SE-0353: [Constrained Existential Types](#)
- SE-0352: [Implicitly Opened Existentials](#)
- SE-0351: [Regex builder DSL](#)
- SE-0350: [Regex Type and Overview](#)
- SE-0349: [Unaligned Loads and Stores from Raw Memory](#)
- SE-0348: [buildPartialBlock for result builders](#)
- SE-0347: [Type inference from default expressions](#)
- SE-0346: [Lightweight same-type requirements for primary associated types](#)
- SE-0345: [if let shorthand for shadowing an existing optional variable](#)
- SE-0344: [Distributed Actor Runtime](#)
- SE-0343: [Concurrency in Top-level Code](#)
- SE-0341: [Opaque Parameter Declarations](#)
- SE-0340: [Unavailable From Async Attribute](#)
- SE-0339: [Module Aliasing For Disambiguation](#)
- SE-0338: [Clarify the Execution of Non-Actor-Isolated Async Functions](#)
- SE-0336: [Distributed Actor Isolation](#)
- SE-0334: [Pointer API Usability Improvements](#)
- SE-0333: [Expand usability of withMemoryRebound](#)
- SE-0329: [Clock, Instant, and Duration](#)
- SE-0328: [Structural opaque result types](#)
- SE-0326: [Enable multi-statement closure parameter/result type inference](#)
- SE-0309: [Unlock existentials for all protocols](#)
- SE-0302: [Sendable and @Sendable closures](#)
- SE-0292: [Package Registry Service](#)

### Swift 5.6

- SE-0337: [Incremental migration to concurrency checking](#)
- SE-0335: [Introduce existential any](#)
- SE-0332: [Package Manager Command Plugins](#)
- SE-0331: [Remove Sendable conformance from unsafe pointer types](#)

## *Language Revision History*

- SE-0325: [Additional Package Plugin APIs](#)
- SE-0324: [Relax diagnostics for pointer arguments to C functions](#)
- SE-0322: [Temporary uninitialized buffers](#)
- SE-0320: [Allow coddling of non String / Int keyed Dictionary into a KeyedContainer](#)
- SE-0315: [Type placeholders \(formerly, “Placeholder types”\)](#)
- SE-0305: [Package Manager Binary Target Improvements](#)
- SE-0303: [Package Manager Extensible Build Tools](#)
- SE-0290: [Unavailability Condition](#)

### Swift 5.5.2

- SE-0323: [Asynchronous Main Semantics](#)

### Swift 5.5

- SE-0319: [Conform Never to Identifiable](#)
- SE-0317: [async let bindings](#)
- SE-0316: [Global actors](#)
- SE-0314: [AsyncStream and AsyncThrowingStream](#)
- SE-0313: [Improved control over actor isolation](#)
- SE-0311: [Task Local Values](#)
- SE-0310: [Effectful Read-only Properties](#)
- SE-0308: [if for postfix member expressions](#)
- SE-0307: [Allow interchangeable use of CGFloat and Double types](#)
- SE-0306: [Actors](#)
- SE-0304: [Structured concurrency](#)
- SE-0300: [Continuations for interfacing async tasks with synchronous code](#)
- SE-0299: [Extending Static Member Lookup in Generic Contexts](#)
- SE-0298: [Async/Await: Sequences](#)
- SE-0297: [Concurrency Interoperability with Objective-C](#)
- SE-0296: [Async/await](#)
- SE-0295: [Codable synthesis for enums with associated values](#)
- SE-0293: [Extend Property Wrappers to Function and Closure Parameters](#)
- SE-0291: [Package Collections](#)

### Swift 5.4

- SE-0294: [Declaring executable targets in Package Manifests](#)
- SE-0289: [Result builders](#)
- SE-0287: [Extend implicit member syntax to cover chains of member references](#)
- SE-0284: [Allow Multiple Variadic Parameters in Functions, Subscripts, and Initializers](#)

### Swift 5.3

- SE-0286: [Forward-scan matching for trailing closures](#)
- SE-0285: [Ease the transition to concise magic file strings](#)

## Language Revision History

- SE-0282: [Clarify the Swift memory consistency model \\*](#)
- SE-0281: [@main: Type-Based Program Entry Points](#)
- SE-0280: [Enum cases as protocol witnesses](#)
- SE-0279: [Multiple Trailing Closures](#)
- SE-0278: [Package Manager Localized Resources](#)
- SE-0277: [Float16](#)
- SE-0276: [Multi-Pattern Catch Clauses](#)
- SE-0273: [Package Manager Conditional Target Dependencies](#)
- SE-0272: [Package Manager Binary Dependencies](#)
- SE-0271: [Package Manager Resources](#)
- SE-0269: [Increase availability of implicit self in @escaping closures when reference cycles are unlikely to occur](#)
- SE-0268: [Refine didSet Semantics](#)
- SE-0267: [where clauses on contextually generic declarations](#)
- SE-0266: [Synthesized Comparable conformance for enum types](#)
- SE-0263: [Add a String Initializer with Access to Uninitialized Storage](#)

### Swift 5.2

- SE-0253: [Callable values of user-defined nominal types](#)
- SE-0249: [Key Path Expressions as Functions](#)
- SE-0226: [Package Manager Target Based Dependency Resolution](#)

### Swift 5.1

- SE-0261: [Identifiable Protocol](#)
- SE-0260: [Library Evolution for Stable ABIs](#)
- SE-0258: [Property Wrappers](#)
- SE-0255: [Implicit returns from single-expression functions](#)
- SE-0254: [Static and class subscripts](#)
- SE-0252: [Key Path Member Lookup](#)
- SE-0251: [SIMD additions](#)
- SE-0248: [String Gaps and Missing APIs](#)
- SE-0247: [Contiguous Strings](#)
- SE-0245: [Add an Array Initializer with Access to Uninitialized Storage](#)
- SE-0244: [Opaque Result Types](#)
- SE-0242: [Synthesize default values for the memberwise initializer](#)
- SE-0240: [Ordered Collection Diffing](#)
- SE-0068: [Expanding Swift Self to class members and value types](#)

### Swift 5.0

- SE-0241: [Deprecate String Index Encoded Offsets](#)
- SE-0239: [Add Codable conformance to Range types](#)
- SE-0238: [Package Manager Target Specific Build Settings](#)

## Language Revision History

- SE-0237: [Introduce withContiguous{Mutable}StorageAvailable methods](#)
- SE-0236: [Package Manager Platform Deployment Settings](#)
- SE-0235: [Add Result to the Standard Library](#)
- SE-0234: [Remove Sequence.SubSequence](#)
- SE-0233: [Make Numeric Refine a new AdditiveArithmetic Protocol](#)
- SE-0232: [Remove Some Customization Points from the Standard Library's Collection Hierarchy](#)
- SE-0230: [Flatten nested optionals resulting from 'try?'](#)
- SE-0229: [SIMD Vectors](#)
- SE-0228: [Fix ExpressibleByStringInterpolation](#)
- SE-0227: [Identify key path](#)
- SE-0225: [Adding isMultiple to BinaryInteger](#)
- SE-0224: [Support 'less than' operator in compilation conditions](#)
- SE-0221: [Character Properties](#)
- SE-0219: [Package Manager Dependency Mirroring](#)
- SE-0218: [Introduce compactMapValues to Dictionary](#)
- SE-0216: [Introduce user-defined dynamically "callable" types](#)
- SE-0215: [Conform Never to Equatable and Hashable](#)
- SE-0214: [Renaming the DictionaryLiteral type to KeyValuePairs](#)
- SE-0213: [Literal initialization via coercion](#)
- SE-0211: [Add Unicode Properties to Unicode.Scalar](#)
- SE-0200: [Enhancing String Literals Delimiters to Support Raw Text](#)
- SE-0192: [Handling Future Enum Cases](#)

### Swift 4.2

- SE-0212: [Compiler Version Directive](#)
- SE-0210: [Add an offset\(of:\) method to MemoryLayout](#)
- SE-0209: [Package Manager Swift Language Version API Update](#)
- SE-0208: [Package Manager System Library Targets](#)
- SE-0207: [Add an allSatisfy algorithm to Sequence](#)
- SE-0206: [Hashable Enhancements](#)
- SE-0205: [withUnsafePointer\(to: :\) and withUnsafeBytes\(of: :\) for immutable values](#)
- SE-0204: [Add last\(where:\) and lastIndex\(where:\) Methods](#)
- SE-0202: [Random Unification](#)
- SE-0201: [Package Manager Local Dependencies](#)
- SE-0199: [Adding toggle to Bool](#)
- SE-0197: [Adding in-place removeAll\(where:\) to the Standard Library](#)
- SE-0196: [Compiler Diagnostic Directives](#)
- SE-0195: [Introduce User-defined "Dynamic Member Lookup" Types](#)
- SE-0194: [Derived Collection of Enum Cases](#)
- SE-0193: [Cross-module inlining and specialization](#)
- SE-0174: [Change RangeReplaceableCollection.filter to return Self](#)

## Language Revision History

- SE-0143: [Conditional conformances](#)
- SE-0079: [Allow using optional binding to upgrade self from a weak to strong reference](#)
- SE-0054: [Abolish ImplicitlyUnwrappedOptional type](#)

### Swift 4.1

- SE-0198: [Playground QuickLook API Revamp](#)
- SE-0191: [Eliminate IndexDistance from Collection](#)
- SE-0190: [Target environment platform condition](#)
- SE-0189: [Restrict Cross-module Struct Initializers](#)
- SE-0188: [Make Standard Library Index Types Hashable](#)
- SE-0187: [Introduce Sequence.compactMap\( : \)](#)
- SE-0186: [Remove ownership keyword support in protocols](#)
- SE-0185: [Synthesizing Equatable and Hashable conformance](#)
- SE-0184: [Unsafe\[Mutable\]\[Raw\]\[Buffer\]Pointer: add missing methods, adjust existing labels for clarity, and remove deallocation size](#)
- SE-0157: [Support recursive constraints on associated types](#)
- SE-0075: [Adding a Build Configuration Import Test](#)

### Swift 4.0

- SE-0183: [Substring performance affordances](#)
- SE-0182: [String Newline Escaping](#)
- SE-0181: [Package Manager C/C++ Language Standard Support](#)
- SE-0180: [String Index Overhaul](#)
- SE-0179: [Swift run Command](#)
- SE-0178: [Add unicodeScalars property to Character](#)
- SE-0176: [Enforce Exclusive Access to Memory](#)
- SE-0175: [Package Manager Revised Dependency Resolution](#)
- SE-0173: [Add MutableCollection.swapAt\( : : \)](#)
- SE-0172: [One-sided Ranges](#)
- SE-0171: [Reduce with inout](#)
- SE-0170: [NSNumber bridging and Numeric types](#)
- SE-0169: [Improve Interaction Between private Declarations and Extensions](#)
- SE-0168: [Multi-Line String Literals](#)
- SE-0167: [Swift Encoders](#)
- SE-0166: [Swift Archival & Serialization](#)
- SE-0165: [Dictionary & Set Enhancements](#)
- SE-0164: [Remove final support in protocol extensions](#)
- SE-0163: [String Revision: Collection Conformance, C Interop, Transcoding](#)
- SE-0162: [Package Manager Custom Target Layouts](#)
- SE-0161: [Smart KeyPaths: Better Key-Value Coding for Swift](#)
- SE-0160: [Limiting @objc inference](#)
- SE-0158: [Package Manager Manifest API Redesign](#)

## Language Revision History

- SE-0156: [Class and Subtype existentials](#)
- SE-0154: [Provide Custom Collections for Dictionary Keys and Values](#)
- SE-0150: [Package Manager Support for branches](#)
- SE-0149: [Package Manager Support for Top of Tree development](#)
- SE-0148: [Generic Subscripts](#)
- SE-0146: [Package Manager Product Definitions](#)
- SE-0142: [Permit where clauses to constrain associated types](#)
- SE-0104: [Protocol-oriented integers](#)

### Swift 3.1

- SE-0152: [Package Manager Tools Version](#)
- SE-0151: [Package Manager Swift Language Compatibility Version](#)
- SE-0147: [Move UnsafeMutablePointer.initialize\(from:\) to UnsafeMutableBufferPointer](#)
- SE-0145: [Package Manager Version Pinning](#)
- SE-0141: [Availability by Swift version](#)
- SE-0082: [Package Manager Editable Packages](#)
- SE-0080: [Failable Numeric Conversion Initializers](#)
- SE-0045: [Add prefix\(while:\) and drop\(while:\) to the stdlib](#)

### Swift 3.0.1

- SE-0140: [Warn when Optional converts to Any, and bridge Optional As Its Payload Or NSNull](#)
- SE-0139: [Bridge Numeric Types to NSNumber and Cocoa Structs to NSValue](#)
- SE-0138: [UnsafeRawBufferPointer](#)

### Swift 3.0

- SE-0155: [Normalize Enum Case Representation](#)
- SE-0137: [Avoiding Lock-In to Legacy Protocol Designs](#)
- SE-0136: [Memory layout of values](#)
- SE-0135: [Package Manager Support for Differentiating Packages by Swift version](#)
- SE-0134: [Rename two UTF8-related properties on String](#)
- SE-0133: [Rename flatten\(\) to joined\(\)](#)
- SE-0131: [Add AnyHashable to the standard library](#)
- SE-0130: [Replace repeating Character and UnicodeScalar forms of String.init](#)
- SE-0129: [Package Manager Test Naming Conventions](#)
- SE-0128: [Change failable UnicodeScalar initializers to failable](#)
- SE-0127: [Clearing up stdlib Pointer and Buffer Routines](#)
- SE-0125: [Remove NonObjectiveCBase and isUniquelyReferenced](#)
- SE-0124: [Int.init\(ObjectIdentifier\) and UInt.init\(ObjectIdentifier\) should have a bitPattern: label](#)
- SE-0121: [Remove Optional Comparison Operators](#)
- SE-0120: [Revise partition Method Signature](#)

## Language Revision History

- SE-0118: [Closure Parameter Names and Labels](#)
- SE-0117: [Allow distinguishing between public access and public overridability](#)
- SE-0116: [Import Objective-C id as Swift Any type](#)
- SE-0115: [Rename Literal Syntax Protocols](#)
- SE-0114: [Updating Buffer “Value” Names to “Header” Names](#)
- SE-0113: [Add integral rounding functions to FloatingPoint](#)
- SE-0112: [Improved NSError Bridging](#)
- SE-0111: [Remove type system significance of function argument labels](#)
- SE-0109: [Remove the Boolean protocol](#)
- SE-0107: [UnsafeRawPointer API](#)
- SE-0106: [Add a macOS Alias for the OSX Platform Configuration Test](#)
- SE-0103: [Make non-escaping closures the default](#)
- SE-0102: [Remove @noreturn attribute and introduce an empty Never type](#)
- SE-0101: [Reconfiguring sizeof and related functions into a unified MemoryLayout struct](#)
- SE-0099: [Restructuring Condition Clauses](#)
- SE-0096: [Converting dynamicType from a property to an operator](#)
- SE-0095: [Replace protocol<P1,P2> syntax with P1 & P2 syntax](#)
- SE-0094: [Add sequence\(first:next:\) and sequence\(state:next:\) to the stdlib](#)
- SE-0093: [Adding a public base property to slices](#)
- SE-0092: [Typealiases in protocols and protocol extensions](#)
- SE-0091: [Improving operator requirements in protocols](#)
- SE-0089: [Renaming String.init<T>\( :T\)](#)
- SE-0088: [Modernize libdispatch for Swift 3 naming conventions](#)
- SE-0086: [Drop NS Prefix in Swift Foundation](#)
- SE-0085: [Package Manager Command Names](#)
- SE-0081: [Move where clause to end of declaration](#)
- SE-0077: [Improved operator declarations](#)
- SE-0076: [Add overrides taking an UnsafePointer source to non-destructive copying methods on UnsafeMutablePointer](#)
- SE-0072: [Fully eliminate implicit bridging conversions from Swift](#)
- SE-0071: [Allow \(most\) keywords in member references](#)
- SE-0070: [Make Optional Requirements Objective-C-only](#)
- SE-0069: [Mutability and Foundation Value Types](#)
- SE-0067: [Enhanced Floating Point Protocols](#)
- SE-0066: [Standardize function type argument syntax to require parentheses](#)
- SE-0065: [A New Model for Collections and Indices](#)
- SE-0064: [Referencing the Objective-C selector of property getters and setters](#)
- SE-0063: [SwiftPM System Module Search Paths](#)
- SE-0062: [Referencing Objective-C key-paths](#)
- SE-0061: [Add Generic Result and Error Handling to autoreleasepool\(\)](#)
- SE-0060: [Enforcing order of Defaulted parameters](#)
- SE-0059: [Update API Naming Guidelines and Rewrite Set APIs Accordingly](#)



## Language Revision History

- SE-0057: [Importing Objective-C Lightweight Generics](#)
- SE-0055: [Make unsafe pointer nullability explicit using Optional](#)
- SE-0053: [Remove explicit use of let from Function Parameters](#)
- SE-0052: [Change IteratorType post-nill guarantee](#)
- SE-0049: [Move @noescape and @autoclosure to be type attributes](#)
- SE-0048: [Generic Type Aliases](#)
- SE-0047: [Defaulting non-Void functions so they warn on unused results](#)
- SE-0046: [Establish consistent label behavior across all parameters including first labels](#)
- SE-0044: [Import as member](#)
- SE-0043: [Declare variables in 'case' labels with multiple patterns](#)
- SE-0040: [Replacing Equal Signs with Colons For Attribute Arguments](#)
- SE-0039: [Modernizing Playground Literals](#)
- SE-0038: [Package Manager C Language Target Support](#)
- SE-0037: [Clarify interaction between comments & operators](#)
- SE-0036: [Requiring Leading Dot Prefixes for Enum Instance Member Implementations](#)
- SE-0035: [Limiting inout capture to @noescape contexts](#)
- SE-0034: [Disambiguating Line Control Statements from Debugging Identifiers](#)
- SE-0033: [Import Objective-C Constants as Swift Types](#)
- SE-0032: [Add first\(where: \) method to Sequence](#)
- SE-0031: [Adjusting inout Declarations for Type Decoration](#)
- SE-0029: [Remove implicit tuple splat behavior from function applications](#)
- SE-0025: [Scoped Access Level](#)
- SC-0023: [API Design Guidelines](#)
- SE-0019: [Swift Testing](#)
- SE-0017: [Change Unmanaged to use UnsafePointer](#)
- SE-0016: [Add initializers to Int and UInt to convert from UnsafePointer and UnsafeMutablePointer](#)
- SE-0008: [Add a Lazy flatMap for Sequences of Optionals](#)
- SE-0007: [Remove C-style for-loops with conditions and incrementers](#)
- SE-0006: [Apply API Guidelines to the Standard Library](#)
- SE-0005: [Better Translation of Objective-C APIs into Swift](#)
- SE-0004: [Remove the ++ and – operators](#)
- SE-0003: [Removing var from Function Parameters](#)
- SE-0002: [Removing currying func declaration syntax](#)

### Swift 2.2

- SE-0028: [Modernizing Swift's Debugging Identifiers](#)
- SE-0022: [Referencing the Objective-C selector of a method](#)
- SE-0021: [Naming Functions with Argument Labels](#)
- SE-0020: [Swift Language Version Build Configuration](#)
- SE-0015: [Tuple comparison operators](#)
- SE-0014: [Constraining AnySequence.init](#)

## *Language Revision History*

- SE-0011: [Replace typealias keyword with associatedtype for associated type declarations](#)
- SE-0001: [Allow \(most\) keywords as argument labels](#)

The following SE proposals that have been implemented in or before Swift 5.9.2, but do not have their corresponding initial Swift version identified:

- SE-0408: [Pack Iteration](#)
- SE-0264: [Standard Library Preview Package](#)
- SE-0110: [Distinguish between single-tuple and multiple-argument function types](#)