iOS消除对应的警告!

WildDylan (/u/81c4380481c1) + 关注
2015.05.11 21:19* 字数 160 阅读 49435 评论 38 喜欢 153
(/u/81c4380481c1)

在iOS开发过程中, 我们可能会碰到一些系统方法弃用, weak、循环引用、不能执行之类的警告。 有代码洁癖的孩子们很想消除他们, 今天就让我们来一次Fuck 警告！！

首先学会基本的语句

```
#pragma clang diagnostic push
#pragma clang diagnostic ignored "-Wdeprecated-declarations"
```

这里写出现警告的代码

```
#pragma clang diagnostic pop
```

这样就消除了方法弃用的警告!

同理, 大家可以在下边搜索到对应的警告, 这样 就可以把前边的字串填入上边的ignored的后边, 然后阔住你的代码, 就OK了

喜欢请关注、哈哈

源网址 (https://link.jianshu.com?t=http://fuckingclangwarnings.com/)

```
Semantic Warnings
Warning Message
-WCFString-literal  input conversion stopped due to an input byte that does not belo
-WNSObject-attribute    __attribute ((NSObject)) may be put on a typedef only, attri
-Wabstract-vbase-init   initializer for virtual base class %0 of abstract class %1 w
-Waddress-of-array-temporary    pointer is initialized by a temporary array, which w
-Warc-maybe-repeated-use-of-weak    "weak %select{variable|property|implicit propert
-Warc-non-pod-memaccess %select{destination for|source of}0 this %1 call is a pointe
-Warc-performSelector-leaks performSelector may cause a leak because its selector is
-Warc-repeated-use-of-weak  "weak %select{variable|property|implicit property|instan
-Warc-retain-cycles capturing %0 strongly in this block is likely to lead to a retai
-Warc-unsafe-retained-assign    assigning retained object to unsafe property object
-Warc-unsafe-retained-assign    assigning %select{array literal|dictionary literal|n
-Warc-unsafe-retained-assign    assigning retained object to %select{weak|unsafe_unr
-Warray-bounds  array index %0 is past the end of the array (which contains %1 eleme
-Warray-bounds  array index %0 is before the beginning of the array
-Warray-bounds  'static' has no effect on zero-length arrays
-Warray-bounds  array argument is too small contains %0 elements, callee requires at
-Warray-bounds-pointer-arithmetic   the pointer incremented by %0 refers past the en
-Warray-bounds-pointer-arithmetic   the pointer decremented by %0 refers before the
-Wassign-enum   integer constant not in range of enumerated type %0
-Watomic-property-with-user-defined-accessor    writable atomic property %0 cannot p
-Wattributes    unknown attribute %0 ignored
-Wauto-var-id   'auto' deduced as 'id' in declaration of %0
-Wavailability  unknown platform %0 in availability macro
-Wavailability  overriding method %select{introduced after|deprecated before|obsolet
-Wavailability  availability does not match previous declaration
-Wavailability  overriding method cannot be unavailable on %0 when its overridden me
-Wavailability  feature cannot be %select{introduced|deprecated|obsoleted}0 in %1 ve
-Wbad-function-cast cast from function call of type %0 to non-matching type %1
-Wbitfield-constant-conversion  implicit truncation from %2 to bitfield changes valu
-Wbitwise-op-parentheses    '&' within '|'
-Wbool-conversion   "initialization of pointer of type %0 to null from a constant bo
-Wbridge-cast   %0 cannot bridge to %1
-Wbridge-cast   %0 bridges to %1, not %2
-Wbuiltin-requires-header   declaration of built-in function '%0' requires inclusion
-Wbuiltin-requires-header   declaration of built-in function '%0' requires inclusion
-Wbuiltin-requires-header   declaration of built-in function '%0' requires inclusion
-Wc++-compat    %select{|empty }0%select{struct|union}1 has size 0 in C, %select{siz
-Wc++11-compat  explicit instantiation cannot be 'inline'
-Wc++11-compat  explicit instantiation of %0 must occur at global scope
-Wc++11-compat  explicit instantiation of %0 not in a namespace enclosing %1
-Wc++11-compat  explicit instantiation of %q0 must occur in namespace %1
-Wc++11-narrowing   constant expression evaluates to %0 which cannot be narrowed to
-Wc++11-narrowing   type %0 cannot be narrowed to %1 in initializer list in C++11
-Wc++11-narrowing   non-constant-expression cannot be narrowed from type %0 to %1 in
-Wc++98-c++11-compat    type definition in a constexpr %select{function|constructor}
-Wc++98-c++11-compat    use of this statement in a constexpr %select{function|constru
-Wc++98-c++11-compat    init-captures.def warn_cxx11_compat_init_capture : Warning "
-Wc++98-c++11-compat    variable declaration in a constexpr %select{function|constru
-Wc++98-c++11-compat    constexpr function with no return statements is incompatible
-Wc++98-c++11-compat    multiple return statements in constexpr function is incompat
-Wc++98-c++11-compat    variable templates are incompatible with C++ standards befor
-Wc++98-compat  substitution failure due to access control is incompatible with C++9
-Wc++98-compat  %select{anonymous struct|union}0 member %1 with a non-trivial %selec
-Wc++98-compat  enumeration type in nested name specifier is incompatible with C++98
-Wc++98-compat  static data member %0 in union is incompatible with C++98
-Wc++98-compat  default template arguments for a function template are incompatible
-Wc++98-compat  scalar initialized from empty initializer list is incompatible with
-Wc++98-compat  befriending %1 without '%select{struct|interface|union|class|enum}0'
-Wc++98-compat  use of null pointer as non-type template argument is incompatible wi
-Wc++98-compat  friend declaration naming a member of the declaring class is incompa
-Wc++98-compat  non-class friend type %0 is incompatible with C++98
-Wc++98-compat  befriending enumeration type %0 is incompatible with C++98
-Wc++98-compat  use of non-static data member %0 in an unevaluated context is incomp
-Wc++98-compat  friend function %0 would be implicitly redefined in C++98
-Wc++98-compat  %select{class template|class template partial|variable template|vari
-Wc++98-compat  reference initialized from initializer list is incompatible with C++
-Wc++98-compat  redundant parentheses surrounding address non-type template argument
-Wc++98-compat  initialization of initializer_list object is incompatible with C++98
-Wc++98-compat  use of 'template' keyword outside of a template is incompatible with
-Wc++98-compat  non-type template argument referring to %select{function|object}0 %1
-Wc++98-compat  use of 'typename' outside of a template is incompatible with C++98
-Wc++98-compat  passing object of trivial but non-POD type %0 through variadic %sele
-Wc++98-compat  goto would jump into protected scope in C++98
-Wc++98-compat  constructor call from initializer list is incompatible with C++98
-Wc++98-compat  'auto' type specifier is incompatible with C++98
-Wc++98-compat  delegating constructors are incompatible with C++98
-Wc++98-compat  'constexpr' specifier is incompatible with C++98
-Wc++98-compat  inheriting constructors are incompatible with C++98
-Wc++98-compat  explicit conversion functions are incompatible with C++98
-Wc++98-compat  switch case would be in a protected scope in C++98
-Wc++98-compat  '%0' type specifier is incompatible with C++98
-Wc++98-compat  indirect goto might cross protected scopes in C++98
-Wc++98-compat-pedantic cast between pointer-to-function and pointer-to-object is in
-Wc++98-compat-pedantic implicit conversion from array size expression of type %0 to
```

```
-Wcast-align    cast from %0 to %1 increases required alignment from %2 to %3
-Wcast-of-sel-type  cast of type %0 to %1 is deprecated use sel_getName instead
-Wchar-subscripts   array subscript is of type 'char'
-Wconditional-uninitialized variable %0 may be uninitialized when %select{used here|
-Wconstant-logical-operand  use of logical '%0' with constant operand
-Wconstexpr-not-const   'constexpr' non-static member function will not be implicitl
-Wconsumed  state of variable '%0' must match at the entry and exit of loop
-Wconsumed  parameter '%0' not in expected state when the function returns: expected
-Wconsumed  argument not in expected state expected '%0', observed '%1'
-Wconsumed  invalid invocation of method '%0' on a temporary object while it is in t
-Wconsumed  return state set for an unconsumable type '%0'
-Wconsumed  consumed analysis attribute is attached to member of class '%0' which is
-Wconsumed  invalid invocation of method '%0' on object '%1' while it is in the '%2'
-Wconsumed  return value not in expected state expected '%0', observed '%1'
-Wconversion    implicit conversion discards imaginary component: %0 to %1
-Wconversion    non-type template argument with value '%0' converted to '%1' for uns
-Wconversion    implicit conversion loses floating-point precision: %0 to %1
-Wconversion    implicit conversion loses integer precision: %0 to %1
-Wconversion    non-type template argument value '%0' truncated to '%1' for template
-Wconversion    implicit conversion turns vector to scalar: %0 to %1
-Wconversion    implicit conversion turns floating-point number into integer: %0 to
-Wcovered-switch-default    default label in switch which covers all enumeration val
-Wcustom-atomic-properties  atomic by default property %0 has a user defined %select
-Wdangling-field    initializing pointer member %0 with the stack address of paramet
-Wdangling-field    binding reference %select{|subobject of }1member %0 to a tempora
-Wdangling-field    binding reference member %0 to stack allocated parameter %1
-Wdangling-initializer-list array backing the initializer list will be destroyed at
-Wdelete-incomplete deleting pointer to incomplete type %0 may cause undefined behav
-Wdelete-non-virtual-dtor   delete called on %0 that is abstract but has non-virtual
-Wdelete-non-virtual-dtor   delete called on %0 that has virtual functions but non-v
-Wdeprecated    access declarations are deprecated use using declarations instead
-Wdeprecated    definition of implicit copy %select{constructor|assignment operator}
-Wdeprecated    dynamic exception specifications are deprecated
-Wdeprecated-increment-bool incrementing expression of type bool is deprecated
-Wdeprecated-objc-isa-usage assignment to Objective-C's isa is deprecated in favor o
-Wdeprecated-objc-isa-usage direct access to Objective-C's isa is deprecated in favo
-Wdeprecated-objc-pointer-introspection bitmasking for introspection of Objective-C
-Wdeprecated-objc-pointer-introspection-performSelector warn_objc_pointer_masking.Te
-Wdeprecated-writable-strings   dummy warning to enable -fconst-strings
-Wdirect-ivar-access    instance variable %0 is being directly accessed
-Wdistributed-object-modifiers  conflicting distributed object modifiers on return t
-Wdistributed-object-modifiers  conflicting distributed object modifiers on paramete
-Wdivision-by-zero  division by zero is undefined
-Wdivision-by-zero  remainder by zero is undefined
-Wdocumentation parameter '%0' not found in the function declaration
-Wdocumentation not a Doxygen trailing comment
-Wduplicate-enum    element %0 has been implicitly assigned %1 which another element
-Wduplicate-method-match    multiple declarations of method %0 found and ignored
-Wdynamic-class-memaccess   %select{destination for|source of|first operand of|secon
-Wempty-body    switch statement has empty body
-Wempty-body    for loop has empty body
-Wempty-body    if statement has empty body
-Wempty-body    range-based for loop has empty body
-Wempty-body    while loop has empty body
-Wenum-compare  comparison of two values with different enumeration types%diff{ ($ a
-Wenum-conversion   implicit conversion from enumeration type %0 to different enumer
-Wexit-time-destructors declaration requires an exit-time destructor
-Wexplicit-ownership-type   method parameter of type %0 with no explicit ownership
-Wextern-c-compat   %select{|empty }0%select{struct|union}1 has size 0 in C, %select
-Wextern-initializer    'extern' variable has an initializer
-Wfloat-equal   comparing floating point with == or != is unsafe
-Wformat    "data argument position '%0' exceeds the number of data arguments (%1)
-Wformat    position arguments in format strings start counting at 1 (not 0)
-Wformat    invalid position specified for %select{field width|field precision}0
-Wformat    cannot mix positional and non-positional arguments in format string
-Wformat    values of type '%0' should not be used as format arguments add an explic
-Wformat    format specifies type %0 but the argument has type %1
-Wformat    zero field width in scanf format string is unused
-Wformat    no closing ']' for '%%[' in scanf format string
-Wformat    format string should not be a wide string
-Wformat    format string contains '\\0' within the string body
-Wformat    '%select{*|.*}0' specified field %select{width|precision}0 is missing a
-Wformat    field %select{width|precision}0 should have type %1, but argument has ty
-Wformat    %select{field width|precision}0 used with '%1' conversion specifier, res
-Wformat    format string missing
-Wformat    incomplete format specifier
-Wformat    flag '%0' results in undefined behavior with '%1' conversion specifier
-Wformat    flag '%0' is ignored when flag '%1' is present
-Wformat    more '%%' conversions than data arguments
-Wformat    length modifier '%0' results in undefined behavior or no effect with '%1
-Wformat-extra-args data argument not used by format string
-Wformat-invalid-specifier  invalid conversion specifier '%0'
-Wformat-nonliteral format string is not a string literal
-Wformat-security   format string is not a string literal (potentially insecure)
-Wformat-zero-length    format string is empty
-Wgcc-compat    GCC does not allow the 'cleanup' attribute argument to be anything o
-Wglobal-constructors   declaration requires a global constructor
```

```
-Wglobal-constructors   declaration requires a global destructor
-Wgnu-conditional-omitted-operand   use of GNU ?: conditional expression extension,
-Wheader-hygiene    using namespace directive in global context in header
-Widiomatic-parentheses using the result of an assignment as a condition without pa
-Wignored-attributes    'malloc' attribute only applies to functions returning a po
-Wignored-attributes    %0 attribute only applies to %select{functions|unions|variab
-Wignored-attributes    '%0' attribute cannot be specified on a definition
-Wignored-attributes    __weak attribute cannot be specified on an automatic variabl
-Wignored-attributes    Objective-C GC does not allow weak variables on the stack
-Wignored-attributes    __weak attribute cannot be specified on a field declaration
-Wignored-attributes    attribute %0 cannot be applied to %select{functions|Objectiv
-Wignored-attributes    attribute declaration must precede definition
-Wignored-attributes    attribute %0 is ignored, place it after \"%select{class|stru
-Wignored-attributes    __declspec attribute %0 is not supported
-Wignored-attributes    attribute %0 ignored, because it cannot be applied to a type
-Wignored-attributes    attribute %0 after definition is ignored
-Wignored-attributes    %0 attribute ignored
-Wignored-attributes    'sentinel' attribute only supported for variadic %select{fun
-Wignored-attributes    'sentinel' attribute requires named arguments
-Wignored-attributes    '%0' only applies to %select{function|pointer|Objective-C ob
-Wignored-attributes    'nonnull' attribute applied to function with no pointer argu
-Wignored-attributes    %0 attribute can only be applied to instance variables or pr
-Wignored-attributes    ibaction attribute can only be applied to Objective-C instan
-Wignored-attributes    %0 calling convention ignored on variadic function
-Wignored-attributes    %0 only applies to variables with static storage duration an
-Wignored-attributes    %0 attribute argument not supported: %1
-Wignored-attributes    #pragma ms_struct can not be used with dynamic classes or st
-Wignored-attributes    transparent union definition must contain at least one field
-Wignored-attributes    first field of a transparent union cannot have %select{float
-Wignored-attributes    'gnu_inline' attribute requires function to be marked 'inlin
-Wignored-attributes    calling convention %0 ignored for this target
-Wignored-attributes    transparent_union attribute can only be applied to a union d
-Wignored-attributes    %select{alignment|size}0 of field %1 (%2 bits) does not matc
-Wignored-attributes    attribute %0 is already applied
-Wignored-attributes    %0 attribute ignored for field of type %1
-Wignored-attributes    %0 attribute ignored when parsing type
-Wignored-attributes    %0 attribute only applies to %select{functions|methods|prope
-Wignored-attributes    %0 attribute only applies to %select{Objective-C object|poin
-Wignored-attributes    attribute %0 is already applied with different parameters
-Wignored-attributes    unknown visibility %0
-Wignored-qualifiers    "'%0' type qualifier%s1 on return type %plural{1:has|:have}1
-Wignored-qualifiers    ARC %select{unused|__unsafe_unretained|__strong|__weak|__aut
-Wimplicit-atomic-properties    property is assumed atomic by default
-Wimplicit-atomic-properties    property is assumed atomic when auto-synthesizing th
-Wimplicit-fallthrough  fallthrough annotation in unreachable code
-Wimplicit-fallthrough  unannotated fall-through between switch labels
-Wimplicit-fallthrough  fallthrough annotation does not directly precede switch labe
-Wimplicit-function-declaration implicit declaration of function %0
-Wimplicit-function-declaration use of unknown builtin %0
-Wimplicit-retain-self  "block implicitly retains 'self' explicitly mention 'self' t
-Wincompatible-library-redeclaration    incompatible redeclaration of library functi
-Wincomplete-implementation method definition for %0 not found
-Winherited-variadic-ctor   inheriting constructor does not inherit ellipsis
-Winitializer-overrides subobject initialization overrides initialization of other f
-Winitializer-overrides initializer overrides prior initialization of this subobject
-Wint-to-pointer-cast   cast to %1 from smaller integer type %0
-Wint-to-void-pointer-cast  cast to %1 from smaller integer type %0
-Winvalid-iboutlet  IBOutletCollection properties should be copy/strong and not assi
-Winvalid-iboutlet  %select{instance variable|property}2 with %0 attribute must be a
-Winvalid-noreturn  function %0 declared 'noreturn' should not return
-Winvalid-noreturn  function declared 'noreturn' should not return
-Wlarge-by-value-copy   return value of %0 is a large (%1 bytes) pass-by-value objec
-Wlarge-by-value-copy   %0 is a large (%1 bytes) pass-by-value argument pass it by r
-Wliteral-conversion    implicit conversion from %0 to %1 changes value from %2 to %
-Wliteral-range magnitude of floating-point constant too large for type %0 maximum i
-Wliteral-range magnitude of floating-point constant too small for type %0 minimum i
-Wlogical-not-parentheses   logical not is only applied to the left hand side of thi
-Wlogical-op-parentheses    '&&' within '||'
-Wloop-analysis variable%select{s| %1|s %1 and %2|s %1, %2, and %3|s %1, %2, %3, and
-Wloop-analysis variable %0 is %select{decremented|incremented}1 both in the loop he
-Wmethod-signatures conflicting parameter types in implementation of %0: %1 vs %2
-Wmethod-signatures conflicting return type in implementation of %0: %1 vs %2
-Wmicrosoft extra qualification on member %0
-Wmismatched-method-attributes  attributes on method implementation and its declarat
-Wmismatched-parameter-types    conflicting parameter types in implementation of %0%
-Wmismatched-return-types   conflicting return type in implementation of %0%diff{: s
-Wmissing-braces    suggest braces around initialization of subobject
-Wmissing-declarations  '%0' ignored on this declaration
-Wmissing-field-initializers    missing field '%0' initializer
-Wmissing-method-return-type    method has no return type specified defaults to 'id'
-Wmissing-noreturn  %select{function|method}0 %1 could be declared with attribute 'n
-Wmissing-noreturn  block could be declared with attribute 'noreturn'
-Wmissing-prototypes    no previous prototype for function %0
-Wmissing-variable-declarations no previous extern declaration for non-static variab
-Wmultiple-move-vbase   defaulted move assignment operator of %0 will move assign vi
-Wnested-anon-types anonymous types declared in an anonymous union/struct are an ext
-Wno-typedef-redefinition   Redefinition of typedef '%0' is a C11 feature
```

```
-Wnon-literal-null-conversion   "expression which evaluates to zero treated as a nul
-Wnon-pod-varargs   second argument to 'va_arg' is of ARC ownership-qualified type %
-Wnon-pod-varargs   cannot pass %select{non-POD|non-trivial}0 object of type %1 to v
-Wnon-pod-varargs   second argument to 'va_arg' is of non-POD type %0
-Wnon-pod-varargs   cannot pass object of %select{non-POD|non-trivial}0 type %1 thro
-Wnon-virtual-dtor  %0 has virtual functions but non-virtual destructor
-Wnonnull   null passed to a callee which requires a non-null argument
-Wnull-arithmetic   use of NULL in arithmetic operation
-Wnull-arithmetic   comparison between NULL and non-pointer %select{(%1 and NULL)|(N
-Wnull-dereference  indirection of non-volatile null pointer will be deleted, not tr
-Wobjc-autosynthesis-property-ivar-name-match   autosynthesized property %0 will use
-Wobjc-forward-class-redefinition   redefinition of forward class %0 of a typedef na
-Wobjc-interface-ivars  declaration of instance variables in the interface is deprec
-Wobjc-literal-compare  direct comparison of %select{an array literal|a dictionary l
-Wobjc-literal-missing-atsign   string literal must be prefixed by '@'
-Wobjc-method-access    instance method %objcinstance0 not found (return type defaul
-Wobjc-method-access    class method %objcclass0 not found (return type defaults to
-Wobjc-method-access    instance method %objcinstance0 not found (return type defaul
-Wobjc-method-access    instance method %0 is being used on 'Class' which is not in
-Wobjc-method-access    class method %objcclass0 not found (return type defaults to
-Wobjc-method-access    instance method %0 found instead of class method %1
-Wobjc-missing-property-synthesis   "auto property synthesis is synthesizing propert
-Wobjc-missing-super-calls  method possibly missing a [super %0] call
-Wobjc-noncopy-retain-block-property    "retain'ed block property does not copy the
-Wobjc-nonunified-exceptions    can not catch an exception thrown with @throw in C++
-Wobjc-property-implementation  property %0 requires method %1 to be defined - use @
-Wobjc-property-implementation  property %0 requires method %1 to be defined - use @
-Wobjc-property-implicit-mismatch   "primary property declaration is implicitly strc
-Wobjc-property-matches-cocoa-ownership-rule    property's synthesized getter follow
-Wobjc-property-no-attribute    no 'assign', 'retain', or 'copy' attribute is specif
-Wobjc-property-no-attribute    default property attribute 'assign' not appropriate
-Wobjc-property-synthesis   auto property synthesis will not synthesize property '%0
-Wobjc-property-synthesis   "auto property synthesis will not synthesize property '%
-Wobjc-protocol-method-implementation   category is implementing a method which will
-Wobjc-protocol-property-synthesis  auto property synthesis will not synthesize prop
-Wobjc-redundant-literal-use    using %0 with a literal is redundant
-Wobjc-root-class   class %0 defined without specifying a base class
-Wobjc-string-compare   direct comparison of a string literal has undefined behavior
-Wobjc-string-concatenation concatenated NSString literal for an NSArray expression
-Wover-aligned  type %0 requires %1 bytes of alignment and the default allocator onl
-Woverloaded-shift-op-parentheses   overloaded operator %select{|}0 has lower preced
-Woverloaded-virtual    %q0 hides overloaded virtual %select{function|functions}1
-Woverriding-method-mismatch    conflicting distributed object modifiers on paramete
-Woverriding-method-mismatch    conflicting parameter types in declaration of %0: %1
-Woverriding-method-mismatch    conflicting variadic declaration of method and its i
-Woverriding-method-mismatch    conflicting distributed object modifiers on return t
-Woverriding-method-mismatch    conflicting parameter types in declaration of %0%dif
-Woverriding-method-mismatch    conflicting return type in declaration of %0%diff{: 
-Woverriding-method-mismatch    conflicting return type in declaration of %0: %1 vs 
-Wpacked    packed attribute is unnecessary for %0
-Wpadded    padding %select{struct|interface|class}0 %1 with %2 %select{byte|bit}3%s
-Wpadded    padding %select{struct|interface|class}0 %1 with %2 %select{byte|bit}3%s
-Wpadded    padding size of %0 with %1 %select{byte|bit}2%select{|s}3 to alignment b
-Wparentheses   using the result of an assignment as a condition without parentheses
-Wparentheses   %0 has lower precedence than %1 %1 will be evaluated first
-Wparentheses   operator '?:' has lower precedence than '%0' '%0' will be evaluated
-Wparentheses-equality  equality comparison with extraneous parentheses
-Wpointer-arith subtraction of pointers to type %0 of zero size has undefined behavi
-Wpredefined-identifier-outside-function    predefined identifier is only valid insi
-Wprivate-extern    use of __private_extern__ on a declaration may not produce exter
-Wprotocol  method %0 in protocol not implemented
-Wprotocol-property-synthesis-ambiguity property of type %0 was selected for synthes
-Wreadonly-iboutlet-property    readonly IBOutlet property '%0' when auto-synthesize
-Wreadonly-setter-attrs property attributes '%0' and '%1' are mutually exclusive
-Wreceiver-expr receiver type %0 is not 'id' or interface pointer, consider casting
-Wreceiver-forward-class    receiver type %0 for instance message is a forward decla
-Wreceiver-is-weak  "weak %select{receiver|property|implicit property}0 may be unpre
-Wreinterpret-base-class    'reinterpret_cast' %select{from|to}3 class %0 %select{to
-Wreorder   %select{field|base class}0 %1 will be initialized after %select{field|ba
-Wrequires-super-attribute  %0 attribute cannot be applied to %select{methods in pro
-Wreturn-stack-address  returning address of local temporary object
-Wreturn-stack-address  returning address of label, which is local
-Wreturn-stack-address  address of stack memory associated with local variable %0 re
-Wreturn-stack-address  reference to stack memory associated with local variable %0
-Wreturn-stack-address  returning reference to local temporary object
-Wreturn-type   control may reach end of non-void function
-Wreturn-type   non-void %select{function|method}1 %0 should return a value, Default
-Wreturn-type   control reaches end of non-void function
-Wreturn-type-c-linkage %0 has C-linkage specified, but returns incomplete type %1 v
-Wreturn-type-c-linkage %0 has C-linkage specified, but returns user-defined type %1
-Wsection   section does not match previous declaration
-Wselector  creating selector for nonexistent method %0
-Wselector-type-mismatch    multiple selectors named %0 found
-Wself-assign   explicitly assigning a variable of type %0 to itself
-Wself-assign-field assigning %select{field|instance variable}0 to itself
-Wsentinel  "missing sentinel in %select{function call|method dispatch|block call}0
-Wsentinel  not enough variable arguments in %0 declaration to fit a sentinel
```

```
-Wshadow    declaration shadows a %select{" "local variable|" "variable in %2|" "sta
-Wshadow-ivar   local declaration of %0 hides instance variable
-Wshift-count-negative  shift count is negative
-Wshift-count-overflow  shift count = width of type
-Wshift-op-parentheses  operator '%0' has lower precedence than '%1' '%1' will be ev
-Wshift-overflow    signed shift result (%0) requires %1 bits to represent, but %2 c
-Wshift-sign-overflow   signed shift result (%0) sets the sign bit of the shift expr
-Wshorten-64-to-32  implicit conversion loses integer precision: %0 to %1
-Wsign-compare  comparison of integers of different signs: %0 and %1
-Wsign-conversion   implicit conversion changes signedness: %0 to %1
-Wsign-conversion   operand of ? changes signedness: %0 to %1
-Wsizeof-array-argument sizeof on array function parameter will return size of %0 in
-Wsizeof-array-decay    sizeof on pointer operation will return size of %0 instead o
-Wsizeof-pointer-memaccess  '%0' call operates on objects of type %1 while the size
-Wsizeof-pointer-memaccess  argument to 'sizeof' in %0 call is the same pointer type
-Wsometimes-uninitialized   variable %0 is %select{used|captured}1 uninitialized whe
-Wstatic-local-in-inline    non-constant static local variable in inline function ma
-Wstatic-self-init  static variable %0 is suspiciously used within its own initializ
-Wstrict-selector-match multiple methods named %0 found
-Wstring-compare    result of comparison against %select{a string literal|@encode}0
-Wstring-conversion implicit conversion turns string literal into bool: %0 to %1
-Wstring-plus-char  adding %0 to a string pointer does not append to the string
-Wstring-plus-int   adding %0 to a string does not append to the string
-Wstrlcpy-strlcat-size  size argument in %0 call appears to be size of the source ex
-Wstrncat-size  the value of the size argument in 'strncat' is too large, might lead
-Wstrncat-size  size argument in 'strncat' call appears " "to be size of the source
-Wstrncat-size  the value of the size argument to 'strncat' is wrong
-Wsuper-class-method-mismatch   method parameter type %diff{$ does not match super c
-Wswitch    overflow converting case value to switch condition type (%0 to %1)
-Wswitch    case value not in enumerated type %0
-Wswitch    %0 enumeration values not handled in switch: %1, %2, %3...
-Wswitch    enumeration values %0 and %1 not handled in switch
-Wswitch    enumeration value %0 not handled in switch
-Wswitch    enumeration values %0, %1, and %2 not handled in switch
-Wswitch-enum   enumeration values %0, %1, and %2 not explicitly handled in switch
-Wswitch-enum   enumeration values %0 and %1 not explicitly handled in switch
-Wswitch-enum   %0 enumeration values not explicitly handled in switch: %1, %2, %3..
-Wswitch-enum   enumeration value %0 not explicitly handled in switch
-Wtautological-compare  comparison of %0 unsigned%select{| enum}2 expression is alwa
-Wtautological-compare  %select{self-|array }0comparison always evaluates to %select
-Wtautological-compare  comparison of unsigned%select{| enum}2 expression %0 is alwa
-Wtautological-constant-out-of-range-compare    comparison of constant %0 with expre
-Wthread-safety-analysis    locking '%0' that is already locked
-Wthread-safety-analysis    cannot call function '%0' while mutex '%1' is locked
-Wthread-safety-analysis    %select{reading|writing}2 the value pointed to by '%0' r
-Wthread-safety-analysis    unlocking '%0' that was not locked
-Wthread-safety-analysis    mutex '%0' is locked exclusively and shared in the same
-Wthread-safety-analysis    calling function '%0' requires %select{shared|exclusive}
-Wthread-safety-analysis    %select{reading|writing}2 variable '%0' requires locking
-Wthread-safety-analysis    cannot resolve lock expression
-Wthread-safety-analysis    expecting mutex '%0' to be locked at the end of function
-Wthread-safety-analysis    mutex '%0' is not locked on every path through here
-Wthread-safety-analysis    %select{reading|writing}1 the value pointed to by '%0' r
-Wthread-safety-analysis    %select{reading|writing}1 variable '%0' requires locking
-Wthread-safety-analysis    mutex '%0' is still locked at the end of function
-Wthread-safety-analysis    expecting mutex '%0' to be locked at start of each loop
-Wthread-safety-attributes  ignoring %0 attribute because its argument is invalid
-Wthread-safety-attributes  %0 attribute only applies to %select{fields and global v
-Wthread-safety-attributes  %0 attribute requires arguments that are class type or p
-Wthread-safety-attributes  %0 attribute can only be applied in a context annotated
-Wthread-safety-attributes  %0 attribute requires arguments whose type is annotated
-Wthread-safety-attributes  '%0' only applies to pointer types type here is %1
-Wthread-safety-beta    Thread safety beta warning.
-Wthread-safety-precise %select{reading|writing}2 the value pointed to by '%0' requi
-Wthread-safety-precise %select{reading|writing}2 variable '%0' requires locking %se
-Wthread-safety-precise calling function '%0' requires %select{shared|exclusive}2 lo
-Wtype-safety   this type tag was not designed to be used with this function
-Wtype-safety   specified %0 type tag requires a null pointer
-Wtype-safety   argument type %0 doesn't match specified '%1' type tag %select{that
-Wundeclared-selector   undeclared selector %0 did you mean %1?
-Wundeclared-selector   undeclared selector %0
-Wundefined-inline  inline function %q0 is not defined
-Wundefined-internal    %select{function|variable}0 %q1 has internal linkage but is
-Wundefined-reinterpret-cast    dereference of type %1 that was reinterpret_cast fro
-Wundefined-reinterpret-cast    reinterpret_cast from %0 to %1 has undefined behavio
-Wuninitialized reference %0 is not yet bound to a value when used within its own in
-Wuninitialized field %0 is uninitialized when used here
-Wuninitialized block pointer variable %0 is uninitialized when captured by block
-Wuninitialized variable %0 is uninitialized when used within its own initialization
-Wuninitialized variable %0 is uninitialized when %select{used here|captured by bloc
-Wuninitialized reference %0 is not yet bound to a value when used here
-Wunneeded-internal-declaration %select{function|variable}0 %1 is not needed and wil
-Wunneeded-internal-declaration 'static' function %0 declared in header file should
-Wunneeded-member-function  member function %0 is not needed and will not be emitted
-Wunreachable-code  will never be executed
-Wunsequenced   multiple unsequenced modifications to %0
-Wunsequenced   unsequenced modification and access to %0
```

```
-Wunsupported-friend    dependent nested name specifier '%0' for friend template dec
-Wunsupported-friend    dependent nested name specifier '%0' for friend class declar
-Wunsupported-visibility    target does not support 'protected' visibility using 'de
-Wunused-comparison %select{equality|inequality}0 comparison result unused
-Wunused-const-variable unused variable %0
-Wunused-exception-parameter    unused exception parameter %0
-Wunused-function   unused function %0
-Wunused-label  unused label %0
-Wunused-member-function    unused member function %0
-Wunused-parameter  unused parameter %0
-Wunused-private-field  private field %0 is not used
-Wunused-property-ivar  ivar %0 which backs the property is not referenced in this p
-Wunused-result ignoring return value of function declared with warn_unused_result a
-Wunused-value  ignoring return value of function declared with %0 attribute
-Wunused-value  expression result unused should this cast be to 'void'?
-Wunused-value  expression result unused
-Wunused-variable   unused variable %0
-Wunused-volatile-lvalue    expression result unused assign into a variable to force
-Wused-but-marked-unused    %0 was marked unused but was used
-Wuser-defined-literals user-defined literal suffixes not starting with '_' are rese
-Wvarargs   second parameter of 'va_start' not last named argument
-Wvarargs   'va_start' has undefined behavior with reference types
-Wvarargs   second argument to 'va_arg' is of promotable type %0 this va_arg has und
-Wvector-conversion incompatible vector types %select{%diff{assigning to $ from $|as
-Wvexing-parse  parentheses were disambiguated as a function declaration
-Wvexing-parse  empty parentheses interpreted as a function declaration
-Wvisibility    declaration of %0 will not be visible outside of this function
-Wvisibility    redefinition of %0 will not be visible outside of this function
-Wvla   variable length array used
-Wvla-extension variable length arrays are a C99 feature
-Wweak-template-vtables explicit template instantiation %0 will emit a vtable in eve
-Wweak-vtables  %0 has no out-of-line virtual method definitions; its vtable will be
Lexer Warnings
Warning Message
-W#pragma-messages  %0
-W#warnings %0
-W#warnings %0
-Wambiguous-macro   ambiguous expansion of macro %0
-Wauto-import   treating #%select{include|import|include_next|__include_macros}0 as
-Wbackslash-newline-escape  backslash and newline separated by space
-Wc++11-compat  identifier after literal will be treated as a user-defined literal s
-Wc++11-compat  '%0' is a keyword in C++11
-Wc++98-c++11-compat    digit separators are incompatible with C++ standards before
-Wc++98-c++11-compat-pedantic   binary integer literals are incompatible with C++ st
-Wc++98-compat  raw string literals are incompatible with C++98
-Wc++98-compat  unicode literals are incompatible with C++98
-Wc++98-compat  universal character name referring to a control character is incompa
-Wc++98-compat  '::' is treated as digraph ':' (aka '[') followed by ':' in C++98
-Wc++98-compat  using this character in an identifier is incompatible with C++98
-Wc++98-compat  specifying character '%0' with a universal character name is incompa
-Wc++98-compat-pedantic variadic macros are incompatible with C++98
-Wc++98-compat-pedantic #line number greater than 32767 is incompatible with C++98
-Wc++98-compat-pedantic C++98 requires newline at end of file
-Wc++98-compat-pedantic empty macro arguments are incompatible with C++98
-Wc99-compat    unicode literals are incompatible with C99
-Wc99-compat    %select{using this character in an identifier|starting an identifier
-Wcomment   '/*' within block comment
-Wcomment   escaped newline between */ characters at block comment end
-Wdisabled-macro-expansion  disabled expansion of recursive macro
-Wheader-guard  %0 is used as a header guard here, followed by #define of a differen
-Wignored-attributes    unknown attribute '%0'
-Wincomplete-module header '%0' is included in module '%1' but not listed in module
-Wincomplete-umbrella   umbrella header for module '%0' does not include header '%1'
-Winvalid-token-paste   pasting formed '%0', an invalid preprocessing token, Default
-Wmalformed-warning-check   __has_warning expected option name (e.g. \"-Wundef\")
-Wnewline-eof   no newline at end of file
-Wnull-character    null character ignored
-Wnull-character    null character(s) preserved in string literal
-Wnull-character    null character(s) preserved in character literal
-Wtrigraphs ignored trigraph would end block comment
-Wtrigraphs trigraph ignored
-Wundef %0 is not defined, evaluates to 0
-Wunicode   universal character names are only valid in C99 or C++ treating as '\\'
-Wunicode   \\%0 used with no following hex digits treating as '\\' followed by iden
-Wunicode   incomplete universal character name treating as '\\' followed by identif
-Wunicode   universal character name refers to a surrogate character
-Wunknown-pragmas   unknown pragma ignored
-Wunknown-pragmas   pragma STDC FENV_ACCESS ON is not supported, ignoring pragma
-Wunused-macros macro is not used
Parser Warnings
Warning Message
-Warc-bridge-casts-disallowed-in-nonarc '%0' casts have no effect when not using ARC
-Wattributes    unknown __declspec attribute %0 ignored
-Wavailability  'unavailable' availability overrides all other availability informat
-Wc++11-compat  use of right-shift operator ('') in template argument will require p
-Wc++11-compat  'auto' storage class specifier is redundant and incompatible with C+
-Wc++98-c++11-compat    'decltype(auto)' type specifier is incompatible with C++ sta
```

```
-Wc++98-compat  range-based for loop is incompatible with C++98
-Wc++98-compat  alias declarations are incompatible with C++98
-Wc++98-compat  in-class initialization of non-static data members is incompatible v
-Wc++98-compat  defaulted function definitions are incompatible with C++98
-Wc++98-compat  rvalue references are incompatible with C++98
-Wc++98-compat  reference qualifiers on functions are incompatible with C++98
-Wc++98-compat  inline namespaces are incompatible with C++98
-Wc++98-compat  generalized initializer lists are incompatible with C++98
-Wc++98-compat  trailing return types are incompatible with C++98
-Wc++98-compat  enumeration types with a fixed underlying type are incompatible with
-Wc++98-compat  alignof expressions are incompatible with C++98
-Wc++98-compat  '%0' keyword is incompatible with C++98
-Wc++98-compat  'decltype' type specifier is incompatible with C++98
-Wc++98-compat  deleted function definitions are incompatible with C++98
-Wc++98-compat  consecutive right angle brackets are incompatible with C++98 (use '>
-Wc++98-compat  static_assert declarations are incompatible with C++98
-Wc++98-compat  scoped enumerations are incompatible with C++98
-Wc++98-compat  lambda expressions are incompatible with C++98
-Wc++98-compat  attributes are incompatible with C++98
-Wc++98-compat  'alignas' is incompatible with C++98
-Wc++98-compat  noexcept specifications are incompatible with C++98
-Wc++98-compat  literal operators are incompatible with C++98
-Wc++98-compat  noexcept expressions are incompatible with C++98
-Wc++98-compat  'nullptr' is incompatible with C++98
-Wc++98-compat-pedantic extra '' outside of a function is incompatible with C++98
-Wc++98-compat-pedantic extern templates are incompatible with C++98
-Wc++98-compat-pedantic commas at the end of enumerator lists are incompatible with
-Wdangling-else add explicit braces to avoid dangling else
-Wdeprecated    Use of 'long' with '__vector' is deprecated
-Wdeprecated-declarations   use of C-style parameters in Objective-C method declarat
-Wdeprecated-register   'register' storage class specifier is deprecated
-Wduplicate-decl-specifier  duplicate '%0' declaration specifier
-Wextra-semi    extra ';' after member function definition
-Wextra-tokens  "extra tokens at the end of '#pragma omp %0' are ignored
-Wgcc-compat    GCC does not allow %0 attribute in this position on a function defin
-Wignored-attributes    attribute %0 ignored, because it is not attached to a declar
-Wmicrosoft-exists  dependent %select{__if_not_exists|__if_exists}0 declarations are
-Wmissing-selector-name %0 used as the name of the previous parameter rather than as
-Wsemicolon-before-method-body  semicolon before method body is ignored
-Wsource-uses-openmp    "unexpected '#pragma omp ...' in program
-Wstatic-inline-explicit-instantiation  ignoring '%select{static|inline}0' keyword o
```

小礼物走一走，来简书关注我

赞赏支持

WildDylan (/u/81c4380481c1) ♂

写了 210627 字，被 2192 人关注，获得了 1760 个喜欢

(/u/81c4380481c1)

+关注

相见不如怀念。 距离产生美。 钱花哪儿哪儿好。

喜欢 | 153

更多分享

(http://cwb.assets.jianshu.io/notes/images/1412103

下载简书 App ▸

随时随地发现和创作内容

(/apps/redirect?utm_source=note-bottom-click)