



Khaos v.0.1.0.0

Cross-platform library

Generated by Doxygen 1.9.2

1 File Index	1
1.1 File List	1
2 File Documentation	3
2.1 khaos/Architecture.h File Reference	3
2.1.1 Detailed Description	4
2.1.2 Macro Definition Documentation	4
2.1.2.1 ARCHITECTURE_IS	4
2.1.2.2 KHAOS_ARCH_aarch64	6
2.1.2.3 KHAOS_ARCH_Alpha	6
2.1.2.4 KHAOS_ARCH_AlphaEV4	6
2.1.2.5 KHAOS_ARCH_AlphaEV5	7
2.1.2.6 KHAOS_ARCH_AlphaEV6	7
2.1.2.7 KHAOS_ARCH_ARM	7
2.1.2.8 KHAOS_ARCH_ARM4	7
2.1.2.9 KHAOS_ARCH_ARM5	7
2.1.2.10 KHAOS_ARCH_ARM6	7
2.1.2.11 KHAOS_ARCH_ARM7	8
2.1.2.12 KHAOS_ARCH_ARM8	8
2.1.2.13 KHAOS_ARCH_Blackfin	8
2.1.2.14 KHAOS_ARCH_Convex	8
2.1.2.15 KHAOS_ARCH_Convex1	8
2.1.2.16 KHAOS_ARCH_Convex2	8
2.1.2.17 KHAOS_ARCH_Convex3_2	9
2.1.2.18 KHAOS_ARCH_Convex3_4	9
2.1.2.19 KHAOS_ARCH_Convex3_8	9
2.1.2.20 KHAOS_ARCH_E2K	9
2.1.2.21 KHAOS_ARCH_Epiphany	9
2.1.2.22 KHAOS_ARCH_i386	9
2.1.2.23 KHAOS_ARCH_i486	10
2.1.2.24 KHAOS_ARCH_i586	10
2.1.2.25 KHAOS_ARCH_i686	10
2.1.2.26 KHAOS_ARCH_IA64	10
2.1.2.27 KHAOS_ARCH_m68k	10
2.1.2.28 KHAOS_ARCH_m68k0	10
2.1.2.29 KHAOS_ARCH_m68k1	11
2.1.2.30 KHAOS_ARCH_m68k2	11
2.1.2.31 KHAOS_ARCH_m68k3	11
2.1.2.32 KHAOS_ARCH_m68k4	11
2.1.2.33 KHAOS_ARCH_m68k6	11
2.1.2.34 KHAOS_ARCH_MIPS	11
2.1.2.35 KHAOS_ARCH_MIPS1	12

2.1.2.36 KHAOS_ARCH_MIPS2	12
2.1.2.37 KHAOS_ARCH_MIPS3	12
2.1.2.38 KHAOS_ARCH_MIPS4	12
2.1.2.39 KHAOS_ARCH_PARISC	12
2.1.2.40 KHAOS_ARCH_PARISC1	12
2.1.2.41 KHAOS_ARCH_PARISC1_1	13
2.1.2.42 KHAOS_ARCH_PARISC2	13
2.1.2.43 KHAOS_ARCH_PowerPC	13
2.1.2.44 KHAOS_ARCH_PowerPC6_1	13
2.1.2.45 KHAOS_ARCH_PowerPC6_3	13
2.1.2.46 KHAOS_ARCH_PowerPC6_4	13
2.1.2.47 KHAOS_ARCH_ptx	14
2.1.2.48 KHAOS_ARCH_Pyramid	14
2.1.2.49 KHAOS_ARCH_RISCV	14
2.1.2.50 KHAOS_ARCH_RS6000	14
2.1.2.51 KHAOS_ARCH_sh	14
2.1.2.52 KHAOS_ARCH_sh1	14
2.1.2.53 KHAOS_ARCH_sh2	15
2.1.2.54 KHAOS_ARCH_sh3	15
2.1.2.55 KHAOS_ARCH_sh4	15
2.1.2.56 KHAOS_ARCH_sh5	15
2.1.2.57 KHAOS_ARCH_SPARC	15
2.1.2.58 KHAOS_ARCH_SPARC8	15
2.1.2.59 KHAOS_ARCH_SPARC9	16
2.1.2.60 KHAOS_ARCH_SYS370	16
2.1.2.61 KHAOS_ARCH_SYS390	16
2.1.2.62 KHAOS_ARCH_VERSION	16
2.1.2.63 KHAOS_ARCH_VERSION_TWEAK	16
2.1.2.64 KHAOS_ARCH_X86_32	16
2.1.2.65 KHAOS_ARCH_x86_64	17
2.1.2.66 KHAOS_ARCH_z	17
2.2 Architecture.h	17
2.3 khaos/C.h File Reference	22
2.3.1 Detailed Description	22
2.3.2 Macro Definition Documentation	22
2.3.2.1 KHAOS_LANGUAGE_Extensions	22
2.3.2.2 KHAOS_LANGUAGE_ISO	23
2.4 C.h	23
2.5 khaos/Compiler.h File Reference	23
2.5.1 Detailed Description	26
2.5.2 Macro Definition Documentation	26
2.5.2.1 COMPILER_IS	26

2.5.3 Detected compilers are :	26
2.5.3.1 ACC	26
2.5.3.2 Altium MicroBlaze C	27
2.5.3.3 Altium C-to-Hardware	27
2.5.3.4 Amsterdam Compiler Kit	27
2.5.3.5 ARM Compiler	28
2.5.3.6 Aztec C	28
2.5.3.7 Borland	28
2.5.3.8 CC65	28
2.5.3.9 Clang	29
2.5.3.10 Comeau C++	29
2.5.3.11 Compaq C/C++	29
2.5.3.12 Convex C	30
2.5.3.13 CompCert	30
2.5.3.14 Coverity	30
2.5.3.15 Cray C	30
2.5.3.16 Diab C/C++	31
2.5.3.17 DICE C	31
2.5.3.18 Digital mars	31
2.5.3.19 Dignus Systems/C++	32
2.5.3.20 DJGPP	32
2.5.3.21 EDG C++ Frontend	32
2.5.3.22 EKOPath	32
2.5.3.23 Emscripten	33
2.5.3.24 Fujitsu C++	33
2.5.3.25 Gnu GCC C/C++	33
2.5.3.26 GCC XML	34
2.5.3.27 Green Hills	34
2.5.3.28 HP ANSI C	34
2.5.3.29 HP aC++ compiler	34
2.5.3.30 IAR C/C++	35
2.5.3.31 IBM XL C/C++ (Clang-based versions)	35
2.5.3.32 IBM XL C/C++ (legacy)	35
2.5.3.33 IBM z/OS XL C/C++	36
2.5.3.34 ImageCraft C	36
2.5.3.35 Intel C/C++	36
2.5.3.36 Kai C++ compiler	37
2.5.3.37 KEIL CARM	37
2.5.3.38 KEIL C166	37
2.5.3.39 KEIL C51	38
2.5.3.40 LCC	38
2.5.3.41 LLVM	38

2.5.3.42 MetaWare High C/C++	38
2.5.3.43 Metrowerks CodeWarrior	39
2.5.3.44 Microsoft Visual C/C++	39
2.5.3.45 Microtec C/C++	39
2.5.3.46 Microway NDP C	40
2.5.3.47 SGI MIPSpro	40
2.5.3.48 Miracle C	40
2.5.3.49 MPW C++	41
2.5.3.50 Norcroft C	41
2.5.3.51 NWCC	41
2.5.3.52 Open64	41
2.5.3.53 Oracle Pro*C Precompiler	42
2.5.3.54 Oracle Solaris Studio	42
2.5.3.55 Pacific C	42
2.5.3.56 Palm C/C++	42
2.5.3.57 Pelles C	43
2.5.3.58 Portland Group C/C++	43
2.5.3.59 Renesas C/C++	43
2.5.3.60 SAS/C	43
2.5.3.61 SCO OpenServer	44
2.5.3.62 Small Device C Compiler	44
2.5.3.63 SN Compiler	44
2.5.3.64 Stratus VOS C	45
2.5.3.65 Symantec C++	45
2.5.3.66 TenDRA C/C++	45
2.5.3.67 Texas Instruments C/C++ Compiler	45
2.5.3.68 THINK C	46
2.5.3.69 Tiny C	46
2.5.3.70 Turbo C/C++	46
2.5.3.71 Ultimate C/C++	46
2.5.3.72 USL C	47
2.5.3.73 VBCC	47
2.5.3.74 Watcom C++	47
2.5.3.75 Zortech C++	47
2.5.3.76 NVCC	48
2.5.3.77 COMPILER_IS_GREATER_EQUAL	48
2.5.3.78 COMPILER_IS_LOWER	48
2.5.3.79 KHAOS_CLANG_EMULATED_VERSION	49
2.5.3.80 KHAOS_CLANG_EMULATED_VERSION_TWEAK	49
2.5.3.81 KHAOS_COMPILER_ACC	49
2.5.3.82 KHAOS_COMPILER_Altium	49
2.5.3.83 KHAOS_COMPILER_AltiumMicroblaze	49

2.5.3.84 KHAOS_COMPILER_Amsterdam	49
2.5.3.85 KHAOS_COMPILER_AppleClang	50
2.5.3.86 KHAOS_COMPILER_ARM	50
2.5.3.87 KHAOS_COMPILER_Aztec	50
2.5.3.88 KHAOS_COMPILER_Borland	50
2.5.3.89 KHAOS_COMPILER_CC65	50
2.5.3.90 KHAOS_COMPILER_Clang	50
2.5.3.91 KHAOS_COMPILER_ClangEmulated	51
2.5.3.92 KHAOS_COMPILER_Comeau	51
2.5.3.93 KHAOS_COMPILER_Compacq	51
2.5.3.94 KHAOS_COMPILER_Compcert	51
2.5.3.95 KHAOS_COMPILER_Convex	52
2.5.3.96 KHAOS_COMPILER_Coverity	52
2.5.3.97 KHAOS_COMPILER_CrayC	52
2.5.3.98 KHAOS_COMPILER_Diab	52
2.5.3.99 KHAOS_COMPILER_DICE	52
2.5.3.100 KHAOS_COMPILER_DigitalMars	52
2.5.3.101 KHAOS_COMPILER_Dignus	53
2.5.3.102 KHAOS_COMPILER_DJGPP	53
2.5.3.103 KHAOS_COMPILER_EDG	53
2.5.3.104 KHAOS_COMPILER_EKOPath	53
2.5.3.105 KHAOS_COMPILER_Emscripten	53
2.5.3.106 KHAOS_COMPILER_Fujitsu	53
2.5.3.107 KHAOS_COMPILER_GCC	54
2.5.3.108 KHAOS_COMPILER_GCCEmulated	54
2.5.3.109 KHAOS_COMPILER_GCCXML	54
2.5.3.110 KHAOS_COMPILER_GreenHills	54
2.5.3.111 KHAOS_COMPILER_HPCC	54
2.5.3.112 KHAOS_COMPILER_HPCC	54
2.5.3.113 KHAOS_COMPILER_IAR	55
2.5.3.114 KHAOS_COMPILER_IBM	55
2.5.3.115 KHAOS_COMPILER_IBMC370	55
2.5.3.116 KHAOS_COMPILER_IBMClang	55
2.5.3.117 KHAOS_COMPILER_IBMLegacy	55
2.5.3.118 KHAOS_COMPILER_IBMMVS	55
2.5.3.119 KHAOS_COMPILER_IBMOS390	56
2.5.3.120 KHAOS_COMPILER_IBMZOS	56
2.5.3.121 KHAOS_COMPILER_ImageCraft	56
2.5.3.122 KHAOS_COMPILER_Intel	56
2.5.3.123 KHAOS_COMPILER_Kai	56
2.5.3.124 KHAOS_COMPILER_KeilC166	56
2.5.3.125 KHAOS_COMPILER_KeilC51	57

2.5.3.126 KHAOS_COMPILER_KeilCarm	57
2.5.3.127 KHAOS_COMPILER_LCC	57
2.5.3.128 KHAOS_COMPILER_LLVM	57
2.5.3.129 KHAOS_COMPILER_LLVMEmulated	57
2.5.3.130 KHAOS_COMPILER_Metaware	57
2.5.3.131 KHAOS_COMPILER_Metrowerks	58
2.5.3.132 KHAOS_COMPILER_Microtec	58
2.5.3.133 KHAOS_COMPILER_Microway	58
2.5.3.134 KHAOS_COMPILER_MIPSPRO	58
2.5.3.135 KHAOS_COMPILER_Miracle	58
2.5.3.136 KHAOS_COMPILER_MPW	58
2.5.3.137 KHAOS_COMPILER_MSVC	59
2.5.3.138 KHAOS_COMPILER_Norcroft	59
2.5.3.139 KHAOS_COMPILER_NVCC	59
2.5.3.140 KHAOS_COMPILER_NWCC	59
2.5.3.141 KHAOS_COMPILER_Open64	59
2.5.3.142 KHAOS_COMPILER_Oracle	59
2.5.3.143 KHAOS_COMPILER_Pacific	60
2.5.3.144 KHAOS_COMPILER_Palm	60
2.5.3.145 KHAOS_COMPILER_Pelles	60
2.5.3.146 KHAOS_COMPILER_Portland	60
2.5.3.147 KHAOS_COMPILER_Renesas	60
2.5.3.148 KHAOS_COMPILER_SASC	60
2.5.3.149 KHAOS_COMPILER_SCO	61
2.5.3.150 KHAOS_COMPILER_SDCC	61
2.5.3.151 KHAOS_COMPILER_SNC	61
2.5.3.152 KHAOS_COMPILER_Solaris	61
2.5.3.153 KHAOS_COMPILER_Stratus	61
2.5.3.154 KHAOS_COMPILER_StratusANSI	61
2.5.3.155 KHAOS_COMPILER_StratusKR	62
2.5.3.156 KHAOS_COMPILER_Symantec	62
2.5.3.157 KHAOS_COMPILER_Tendra	62
2.5.3.158 KHAOS_COMPILER_Think	62
2.5.3.159 KHAOS_COMPILER_TI	62
2.5.3.160 KHAOS_COMPILER_Tiny	62
2.5.3.161 KHAOS_COMPILER_Turbo	63
2.5.3.162 KHAOS_COMPILER_Ultimate	63
2.5.3.163 KHAOS_COMPILER_Unknown	63
2.5.3.164 KHAOS_COMPILER_USLC	63
2.5.3.165 KHAOS_COMPILER_VBCC	63
2.5.3.166 KHAOS_COMPILER_VERSION	63
2.5.3.167 KHAOS_COMPILER_VERSION_TWEAK	64

2.5.3.168 KHAOS_COMPILER_Watcom	64
2.5.3.169 KHAOS_COMPILER_Zortech	64
2.5.3.170 KHAOS_GCC_EMULATED_VERSION	64
2.5.3.171 KHAOS_GCC_EMULATED_VERSION_TWEAK	64
2.5.3.172 KHAOS_LLVM_EMULATED_VERSION	64
2.5.3.173 KHAOS_LLVM_EMULATED_VERSION_TWEAK	65
2.5.3.174 KHAOS_NVCC_VERSION	65
2.5.3.175 KHAOS_NVCC_VERSION_TWEAK	65
2.6 Compiler.h	65
2.7 khaos/CStandards.h File Reference	73
2.7.1 Detailed Description	74
2.7.2 Macro Definition Documentation	74
2.7.2.1 KHAOS_STANDARD_C11	74
2.7.2.2 KHAOS_STANDARD_C17	74
2.7.2.3 KHAOS_STANDARD_C90	75
2.7.2.4 KHAOS_STANDARD_C95	75
2.7.2.5 KHAOS_STANDARD_C99	75
2.8 CStandards.h	75
2.9 khaos/CXX.h File Reference	75
2.9.1 Macro Definition Documentation	76
2.9.1.1 KHAOS_CXX_STANDARD	76
2.9.1.2 KHAOS_LANGUAGE_Embedded	76
2.9.1.3 KHAOS_LANGUAGE_Extensions	76
2.9.1.4 KHAOS_LANGUAGE_ISO	76
2.10 CXX.h	77
2.11 khaos/CXXStandards.h File Reference	77
2.11.1 Detailed Description	78
2.11.2 Macro Definition Documentation	78
2.11.2.1 KHAOS_STANDARD_CXX11	78
2.11.2.2 KHAOS_STANDARD_CXX14	78
2.11.2.3 KHAOS_STANDARD_CXX17	78
2.11.2.4 KHAOS_STANDARD_CXX20	78
2.11.2.5 KHAOS_STANDARD_CXX97	79
2.12 CXXStandards.h	79
2.13 khaos/Features.h File Reference	79
2.13.1 Macro Definition Documentation	79
2.13.1.1 __has_cpp_attribute	79
2.13.1.2 __has_cpp_attribute_carries_dependency	80
2.13.1.3 __has_feature	80
2.13.1.4 __has_feature_def	80
2.13.1.5 __has_include	80
2.14 Features.h	81

2.15 khaos/Language.h File Reference	81
2.15.1 Macro Definition Documentation	82
2.15.1.1 KHAOS_LANGUAGE_Assembler	82
2.15.1.2 KHAOS_LANGUAGE_C	82
2.15.1.3 KHAOS_LANGUAGE_ClangCuda	82
2.15.1.4 KHAOS_LANGUAGE_Cuda	82
2.15.1.5 KHAOS_LANGUAGE_CXX	82
2.15.1.6 KHAOS_LANGUAGE_Fortran	83
2.15.1.7 KHAOS_LANGUAGE_ObjC	83
2.15.1.8 KHAOS_LANGUAGE_ObjCPP	83
2.15.1.9 LANGUAGE_IS	83
2.15.1.10 LANGUAGE_IS_EQUAL	84
2.15.1.11 LANGUAGE_IS_GREATER	84
2.15.1.12 LANGUAGE_IS_GREATER_EQUAL	85
2.15.1.13 LANGUAGE_IS_LOWER	85
2.15.1.14 LANGUAGE_IS_LOWER_EQUAL	86
2.16 Language.h	86
2.17 khaos/Library.h File Reference	87
2.17.1 Detailed Description	88
2.17.2 Macro Definition Documentation	88
2.17.2.1 C_LIBRARY_IS	88
2.17.2.2 CXX_LIBRARY_IS	89
2.17.2.3 KHAOS_LIBRARYC_Bionic	90
2.17.2.4 KHAOS_LIBRARYC_CloudABI	90
2.17.2.5 KHAOS_LIBRARYC_GNU	90
2.17.2.6 KHAOS_LIBRARYC_klibc	90
2.17.2.7 KHAOS_LIBRARYC_uClibc	90
2.17.2.8 KHAOS_LIBRARYC_VERSION	90
2.17.2.9 KHAOS_LIBRARYC_VERSION_TWEAK	91
2.17.2.10 KHAOS_LIBRARYC_VMC	91
2.17.2.11 KHAOS_LIBRARYC_ZOS	91
2.17.2.12 KHAOS_LIBRARYCXX_Comeau	91
2.17.2.13 KHAOS_LIBRARYCXX_CXX	91
2.17.2.14 KHAOS_LIBRARYCXX_Dinkumware	91
2.17.2.15 KHAOS_LIBRARYCXX_GNU	92
2.17.2.16 KHAOS_LIBRARYCXX_IBM	92
2.17.2.17 KHAOS_LIBRARYCXX_Metrowerks	92
2.17.2.18 KHAOS_LIBRARYCXX_MFC	92
2.17.2.19 KHAOS_LIBRARYCXX_Modena	92
2.17.2.20 KHAOS_LIBRARYCXX_Roguewave	92
2.17.2.21 KHAOS_LIBRARYCXX_SGI	93
2.17.2.22 KHAOS_LIBRARYCXX_STLport	93

2.17.2.23 KHAOS_LIBRARYCXX_VERSION	93
2.17.2.24 KHAOS_LIBRARYCXX_VERSION_TWEAK	93
2.18 Library.h	93
2.19 khaos/OS.h File Reference	96
2.19.1 Detailed Description	97
2.19.2 Macro Definition Documentation	98
2.19.2.1 KHAOS_OS_AIX	98
2.19.2.2 KHAOS_OS_AIX3	98
2.19.2.3 KHAOS_OS_AIX3p2	98
2.19.2.4 KHAOS_OS_AIX4p1	98
2.19.2.5 KHAOS_OS_AIX4p3	98
2.19.2.6 KHAOS_OS_AmigaOS	99
2.19.2.7 KHAOS_OS_BEOS	99
2.19.2.8 KHAOS_OS_BSD	99
2.19.2.9 KHAOS_OS_BSD4p2	99
2.19.2.10 KHAOS_OS_BSD4p3	99
2.19.2.11 KHAOS_OS_BSD4p4	99
2.19.2.12 KHAOS_OS_BSDi	100
2.19.2.13 KHAOS_OS_CYGWIN	100
2.19.2.14 KHAOS_OS_DragonFly	100
2.19.2.15 KHAOS_OS_FreeBSD	100
2.19.2.16 KHAOS_OS_FreeBSD4p10	100
2.19.2.17 KHAOS_OS_FreeBSD4p11	100
2.19.2.18 KHAOS_OS_HAIKU	101
2.19.2.19 KHAOS_OS_HPUX	101
2.19.2.20 KHAOS_OS_iOS	101
2.19.2.21 KHAOS_OS_IRIX	101
2.19.2.22 KHAOS_OS_Linux	101
2.19.2.23 KHAOS_OS_MacOS	101
2.19.2.24 KHAOS_OS_NetBSD	102
2.19.2.25 KHAOS_OS_NetBSD0p8	102
2.19.2.26 KHAOS_OS_NetBSD0p9	102
2.19.2.27 KHAOS_OS_NetBSD1	102
2.19.2.28 KHAOS_OS_OpenBSD	102
2.19.2.29 KHAOS_OS_OpenBSD2_0	102
2.19.2.30 KHAOS_OS_OpenBSD2_1	103
2.19.2.31 KHAOS_OS_OpenBSD2_2	103
2.19.2.32 KHAOS_OS_OpenBSD2_3	103
2.19.2.33 KHAOS_OS_OpenBSD2_4	103
2.19.2.34 KHAOS_OS_OpenBSD2_5	103
2.19.2.35 KHAOS_OS_OpenBSD2_6	103
2.19.2.36 KHAOS_OS_OpenBSD2_7	104

2.19.2.37 KHAOS_OS_OpenBSD2_8	104
2.19.2.38 KHAOS_OS_OpenBSD2_9	104
2.19.2.39 KHAOS_OS_OpenBSD3_0	104
2.19.2.40 KHAOS_OS_OpenBSD3_1	104
2.19.2.41 KHAOS_OS_OpenBSD3_2	104
2.19.2.42 KHAOS_OS_OpenBSD3_3	105
2.19.2.43 KHAOS_OS_OpenBSD3_4	105
2.19.2.44 KHAOS_OS_OpenBSD3_5	105
2.19.2.45 KHAOS_OS_OpenBSD3_6	105
2.19.2.46 KHAOS_OS_OpenBSD3_7	105
2.19.2.47 KHAOS_OS_OpenBSD3_8	105
2.19.2.48 KHAOS_OS_OpenBSD3_9	106
2.19.2.49 KHAOS_OS_OpenBSD4_0	106
2.19.2.50 KHAOS_OS_OpenBSD4_1	106
2.19.2.51 KHAOS_OS_OpenBSD4_2	106
2.19.2.52 KHAOS_OS_OpenBSD4_3	106
2.19.2.53 KHAOS_OS_OpenBSD4_4	106
2.19.2.54 KHAOS_OS_OpenBSD4_5	107
2.19.2.55 KHAOS_OS_OpenBSD4_6	107
2.19.2.56 KHAOS_OS_OpenBSD4_7	107
2.19.2.57 KHAOS_OS_OpenBSD4_8	107
2.19.2.58 KHAOS_OS_OpenBSD4_9	107
2.19.2.59 KHAOS_OS_OpenBSD5_0	107
2.19.2.60 KHAOS_OS_OpenBSD5_1	108
2.19.2.61 KHAOS_OS_OpenBSD5_2	108
2.19.2.62 KHAOS_OS_OpenBSD5_3	108
2.19.2.63 KHAOS_OS_OpenBSD5_4	108
2.19.2.64 KHAOS_OS_OpenBSD5_5	108
2.19.2.65 KHAOS_OS_OpenBSD5_6	108
2.19.2.66 KHAOS_OS_OpenBSD5_7	109
2.19.2.67 KHAOS_OS_OpenBSD5_8	109
2.19.2.68 KHAOS_OS_OpenBSD5_9	109
2.19.2.69 KHAOS_OS_OpenBSD6_0	109
2.19.2.70 KHAOS_OS_OpenBSD6_1	109
2.19.2.71 KHAOS_OS_OpenBSD6_2	109
2.19.2.72 KHAOS_OS_OpenBSD6_3	110
2.19.2.73 KHAOS_OS_OpenBSD6_4	110
2.19.2.74 KHAOS_OS_OpenBSD6_5	110
2.19.2.75 KHAOS_OS_OpenBSD6_6	110
2.19.2.76 KHAOS_OS_OpenBSD6_7	110
2.19.2.77 KHAOS_OS_OpenBSD6_8	110
2.19.2.78 KHAOS_OS_OpenBSD6_9	111

2.19.2.79 KHAOS_OS_OS400	111
2.19.2.80 KHAOS_OS_QNX	111
2.19.2.81 KHAOS_OS_QNX4	111
2.19.2.82 KHAOS_OS_Solaris	111
2.19.2.83 KHAOS_OS_SVR4	111
2.19.2.84 KHAOS_OS_UNIX	112
2.19.2.85 KHAOS_OS_VERSION	112
2.19.2.86 KHAOS_OS_VERSION_TWEAK	112
2.19.2.87 KHAOS_OS_VMS	112
2.19.2.88 KHAOS_OS_Windows	112
2.19.2.89 OS_IS	112
2.20 OS.h	114
2.21 khaos/Platform.h File Reference	120
2.21.1 Detailed Description	120
2.21.2 Macro Definition Documentation	121
2.21.2.1 KHAOS_MINGW_VERSION	121
2.21.2.2 KHAOS_MINGW_VERSION_TWEAK	121
2.21.2.3 KHAOS_PLATFORM_Android	121
2.21.2.4 KHAOS_PLATFORM_CloudABI	121
2.21.2.5 KHAOS_PLATFORM_iOSIphone	121
2.21.2.6 KHAOS_PLATFORM_iOSSimulator	122
2.21.2.7 KHAOS_PLATFORM_iOSTarget	122
2.21.2.8 KHAOS_PLATFORM_MINGW	122
2.21.2.9 KHAOS_PLATFORM_MINGW32	122
2.21.2.10 KHAOS_PLATFORM_MINGW32Emulated	122
2.21.2.11 KHAOS_PLATFORM_MINGW64	122
2.21.2.12 KHAOS_PLATFORM_MINGW64Emulated	123
2.21.2.13 KHAOS_PLATFORM_MINGWEmulated	123
2.21.2.14 KHAOS_PLATFORM_UWP	123
2.21.2.15 KHAOS_PLATFORM_VERSION	123
2.21.2.16 KHAOS_PLATFORM_VERSION_TWEAK	123
2.21.2.17 KHAOS_PLATFORM_WindowsDesktop	123
2.21.2.18 KHAOS_PLATFORM_WindowsPhone	124
2.21.2.19 KHAOS_PLATFORM_WindowsServer	124
2.21.2.20 KHAOS_PLATFORM_WindowsStore	124
2.21.2.21 KHAOS_PLATFORM_WindowsSystem	124
2.21.2.22 PLATFORM_IS	124
2.22 Platform.h	125
2.23 khaos/Pragma.h File Reference	127
2.23.1 Detailed Description	127
2.24 Pragma.h	127
2.25 khaos/Unix.h File Reference	128

2.25.1 Detailed Description	128
2.25.2 Macro Definition Documentation	128
2.25.2.1 KHAOS_UNIX_LSB_VERSION	129
2.25.2.2 KHAOS_UNIX_LSB_VERSION_TWEAK	129
2.25.2.3 KHAOS_UNIX_STANDARD_LSB	129
2.25.2.4 KHAOS_UNIX_STANDARD_POSIX2	129
2.25.2.5 KHAOS_UNIX_STANDARD_POSIX_1988	129
2.25.2.6 KHAOS_UNIX_STANDARD_POSIX_1990	129
2.25.2.7 KHAOS_UNIX_STANDARD_POSIX_1993	130
2.25.2.8 KHAOS_UNIX_STANDARD_POSIX_1995	130
2.25.2.9 KHAOS_UNIX_STANDARD_POSIX_2001	130
2.25.2.10 KHAOS_UNIX_STANDARD_POSIX_2008	130
2.25.2.11 KHAOS_UNIX_STANDARD_SUS	130
2.25.2.12 KHAOS_UNIX_STANDARD_SUSv2	130
2.25.2.13 KHAOS_UNIX_STANDARD_SUSv3	131
2.25.2.14 KHAOS_UNIX_STANDARD_SUSv4	131
2.25.2.15 KHAOS_UNIX_STANDARD_XPG3	131
2.25.2.16 KHAOS_UNIX_STANDARD_XPG4	131
2.25.2.17 UNIX_STANDARD	131
2.26 Unix.h	132
2.27 kaos/VersionMacros.h File Reference	133
2.27.1 Detailed Description	133
2.27.2 Macro Definition Documentation	134
2.27.2.1 KHAOS_GET_VERSION_MAJOR	134
2.27.2.2 KHAOS_GET_VERSION_MINOR	134
2.27.2.3 KHAOS_GET_VERSION_PATCH	134
2.27.2.4 KHAOS_GET_VERSION_TWEAK	135
2.27.2.5 KHAOS_SET_VERSION	135
2.27.2.6 KHAOS_SET_VERSION_TWEAK	136
2.28 VersionMacros.h	136
2.29 kaos/Warning.h File Reference	137
2.29.1 Detailed Description	137
2.29.2 Macro Definition Documentation	137
2.29.2.1 DISABLE_WARNING	137
2.29.2.2 WARNING_POP	138
2.29.2.3 WARNING_PUSH	138
2.30 Warning.h	138

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

khaos/ Architecture.h	
Detect the architecture	3
khaos/ C.h	
Detect the C standard and define some useful macros	22
khaos/ Compiler.h	
Detect the compiler, convert the vendor version number to standardized one if available at compilation time and store it in KHAOS_COMPILER_VERSION	23
khaos/ CStandards.h	
Define the values of the official C standard	73
khaos/ CXX.h	75
khaos/ CXXStandards.h	
Define the version of the C++ standard	77
khaos/ Features.h	79
khaos/ Language.h	81
khaos/ Library.h	
Detect the library and define some useful macros	87
khaos/ OS.h	
Detect the OS	96
khaos/ Platform.h	
Detect the platform	120
khaos/ Pragma.h	
Cross-compiler pragma macro	127
khaos/ Unix.h	
Define the values of the official POSIX,X/Open and LSB standards	128
khaos/ VersionMacros.h	
Define macros to set, get and compare versions	133
khaos/ Warning.h	
Cross-compiler warning push and pop macros	137

Chapter 2

File Documentation

2.1 khaos/Architecture.h File Reference

Detect the architecture.

```
#include "VersionMacros.h"
```

Macros

- #define [ARCHITECTURE_IS](#)(architecture)
Detect the architecture.
- #define [KHAOS_ARCH_Alpha](#) 0
- #define [KHAOS_ARCH_AlphaEV4](#) 0
- #define [KHAOS_ARCH_AlphaEV5](#) 0
- #define [KHAOS_ARCH_AlphaEV6](#) 0
- #define [KHAOS_ARCH_ARM](#) 0
- #define [KHAOS_ARCH_ARM7](#) 0
- #define [KHAOS_ARCH_ARM6](#) 0
- #define [KHAOS_ARCH_ARM5](#) 0
- #define [KHAOS_ARCH_ARM4](#) 0
- #define [KHAOS_ARCH_ARM8](#) 0
- #define [KHAOS_ARCH_aarch64](#) 0
- #define [KHAOS_ARCH_Blackfin](#) 0
- #define [KHAOS_ARCH_Convex](#) 0
- #define [KHAOS_ARCH_Convex1](#) 0
- #define [KHAOS_ARCH_Convex2](#) 0
- #define [KHAOS_ARCH_Convex3_2](#) 0
- #define [KHAOS_ARCH_Convex3_4](#) 0
- #define [KHAOS_ARCH_Convex3_8](#) 0
- #define [KHAOS_ARCH_E2K](#) 0
- #define [KHAOS_ARCH_IA64](#) 0
- #define [KHAOS_ARCH_m68k](#) 0
- #define [KHAOS_ARCH_m68k6](#) 0
- #define [KHAOS_ARCH_m68k4](#) 0
- #define [KHAOS_ARCH_m68k3](#) 0
- #define [KHAOS_ARCH_m68k2](#) 0

- #define `KHAOS_ARCH_m68k1` 0
- #define `KHAOS_ARCH_m68k0` 0
- #define `KHAOS_ARCH_MIPS` 0
- #define `KHAOS_ARCH_MIPS1` 0
- #define `KHAOS_ARCH_MIPS2` 0
- #define `KHAOS_ARCH_MIPS3` 0
- #define `KHAOS_ARCH_MIPS4` 0
- #define `KHAOS_ARCH_PARISC` 0
- #define `KHAOS_ARCH_PARISC1` 0
- #define `KHAOS_ARCH_PARISC1_1` 0
- #define `KHAOS_ARCH_PARISC2` 0
- #define `KHAOS_ARCH_PowerPC` 0
- #define `KHAOS_ARCH_PowerPC6_1` 0
- #define `KHAOS_ARCH_PowerPC6_3` 0
- #define `KHAOS_ARCH_PowerPC6_4` 0
- #define `KHAOS_ARCH_ptx` 0
- #define `KHAOS_ARCH_Pyramid` 0
- #define `KHAOS_ARCH_RISCV` 0
- #define `KHAOS_ARCH_RS6000` 0
- #define `KHAOS_ARCH_SPARC` 0
- #define `KHAOS_ARCH_SPARC9` 0
- #define `KHAOS_ARCH_SPARC8` 0
- #define `KHAOS_ARCH_sh` 0
- #define `KHAOS_ARCH_sh5` 0
- #define `KHAOS_ARCH_sh4` 0
- #define `KHAOS_ARCH_sh3` 0
- #define `KHAOS_ARCH_sh2` 0
- #define `KHAOS_ARCH_sh1` 0
- #define `KHAOS_ARCH_SYS370` 0
- #define `KHAOS_ARCH_SYS390` 0
- #define `KHAOS_ARCH_z` 0
- #define `KHAOS_ARCH_X86_32` 0
- #define `KHAOS_ARCH_i686` 0
- #define `KHAOS_ARCH_i586` 0
- #define `KHAOS_ARCH_i486` 0
- #define `KHAOS_ARCH_i386` 0
- #define `KHAOS_ARCH_x86_64` 0
- #define `KHAOS_ARCH_Epiphany` 0
- #define `KHAOS_ARCH_VERSION` `KHAOS_SET_VERSION(0, 0, 0)`
- #define `KHAOS_ARCH_VERSION_TWEAK` `KHAOS_SET_VERSION_TWEAK(0)`

2.1.1 Detailed Description

Detect the architecture.

Definition in file [Architecture.h](#).

2.1.2 Macro Definition Documentation

2.1.2.1 ARCHITECTURE_IS

```
#define ARCHITECTURE_IS(  
    architecture )
```

Detect the architecture.

Parameters

in	<i>architecture</i>	The architecture name.
----	---------------------	------------------------

Note

Use this macro only if you need preprocessor architecture checking. For example :

```
#if ARCHITECTURE_IS(ARM)
//...
#else
//...
#endif
```

Values for architecture

Macro
Alpha
AlphaEV4
AlphaEV5
AlphaEV6
ARM
ARM8
ARM7
ARM6
ARM5
ARM4
aarch64
Blackfin
Convex
Convex1
Convex2
Convex3p2
Convex3p4
Convex3p8
E2K
IA64
m68k
m68k6
m68k4
m68k3
m68k2
m68k1
m68k0
MIPS
MIPS1
MIPS2
MIPS3
MIPS4
PARISC
PARISC1
PARISC1p1
PARISC2
PowerPC
PowerPC6p1

Macro
PowerPC6p3
PowerPC6p4
ptx
Pyramid
RISCV
RS6000
SPARC
SPARC9
SPARC8
sh
sh5
sh4
sh3
sh2
sh1
SYS370
SYS390
z
X86_32
i686
i586
i486
i386
x86_64
Epiphany

Definition at line 94 of file [Architecture.h](#).

2.1.2.2 KHAOS_ARCH_aarch64

```
#define KHAOS_ARCH_aarch64 0
```

Definition at line 325 of file [Architecture.h](#).

2.1.2.3 KHAOS_ARCH_Alpha

```
#define KHAOS_ARCH_Alpha 0
```

Definition at line 295 of file [Architecture.h](#).

2.1.2.4 KHAOS_ARCH_AlphaEV4

```
#define KHAOS_ARCH_AlphaEV4 0
```

Definition at line 298 of file [Architecture.h](#).

2.1.2.5 KHAOS_ARCH_AlphaEV5

```
#define KHAOS_ARCH_AlphaEV5 0
```

Definition at line 301 of file [Architecture.h](#).

2.1.2.6 KHAOS_ARCH_AlphaEV6

```
#define KHAOS_ARCH_AlphaEV6 0
```

Definition at line 304 of file [Architecture.h](#).

2.1.2.7 KHAOS_ARCH_ARM

```
#define KHAOS_ARCH_ARM 0
```

Definition at line 307 of file [Architecture.h](#).

2.1.2.8 KHAOS_ARCH_ARM4

```
#define KHAOS_ARCH_ARM4 0
```

Definition at line 319 of file [Architecture.h](#).

2.1.2.9 KHAOS_ARCH_ARM5

```
#define KHAOS_ARCH_ARM5 0
```

Definition at line 316 of file [Architecture.h](#).

2.1.2.10 KHAOS_ARCH_ARM6

```
#define KHAOS_ARCH_ARM6 0
```

Definition at line 313 of file [Architecture.h](#).

2.1.2.11 KHAOS_ARCH_ARM7

```
#define KHAOS_ARCH_ARM7 0
```

Definition at line 310 of file [Architecture.h](#).

2.1.2.12 KHAOS_ARCH_ARM8

```
#define KHAOS_ARCH_ARM8 0
```

Definition at line 322 of file [Architecture.h](#).

2.1.2.13 KHAOS_ARCH_Blackfin

```
#define KHAOS_ARCH_Blackfin 0
```

Definition at line 328 of file [Architecture.h](#).

2.1.2.14 KHAOS_ARCH_Convex

```
#define KHAOS_ARCH_Convex 0
```

Definition at line 331 of file [Architecture.h](#).

2.1.2.15 KHAOS_ARCH_Convex1

```
#define KHAOS_ARCH_Convex1 0
```

Definition at line 334 of file [Architecture.h](#).

2.1.2.16 KHAOS_ARCH_Convex2

```
#define KHAOS_ARCH_Convex2 0
```

Definition at line 337 of file [Architecture.h](#).

2.1.2.17 KHAOS_ARCH_Convex3_2

```
#define KHAOS_ARCH_Convex3_2 0
```

Definition at line 340 of file [Architecture.h](#).

2.1.2.18 KHAOS_ARCH_Convex3_4

```
#define KHAOS_ARCH_Convex3_4 0
```

Definition at line 343 of file [Architecture.h](#).

2.1.2.19 KHAOS_ARCH_Convex3_8

```
#define KHAOS_ARCH_Convex3_8 0
```

Definition at line 346 of file [Architecture.h](#).

2.1.2.20 KHAOS_ARCH_E2K

```
#define KHAOS_ARCH_E2K 0
```

Definition at line 349 of file [Architecture.h](#).

2.1.2.21 KHAOS_ARCH_Epiphany

```
#define KHAOS_ARCH_Epiphany 0
```

Definition at line 481 of file [Architecture.h](#).

2.1.2.22 KHAOS_ARCH_i386

```
#define KHAOS_ARCH_i386 0
```

Definition at line 475 of file [Architecture.h](#).

2.1.2.23 KHAOS_ARCH_i486

```
#define KHAOS_ARCH_i486 0
```

Definition at line 472 of file [Architecture.h](#).

2.1.2.24 KHAOS_ARCH_i586

```
#define KHAOS_ARCH_i586 0
```

Definition at line 469 of file [Architecture.h](#).

2.1.2.25 KHAOS_ARCH_i686

```
#define KHAOS_ARCH_i686 0
```

Definition at line 466 of file [Architecture.h](#).

2.1.2.26 KHAOS_ARCH_IA64

```
#define KHAOS_ARCH_IA64 0
```

Definition at line 352 of file [Architecture.h](#).

2.1.2.27 KHAOS_ARCH_m68k

```
#define KHAOS_ARCH_m68k 0
```

Definition at line 355 of file [Architecture.h](#).

2.1.2.28 KHAOS_ARCH_m68k0

```
#define KHAOS_ARCH_m68k0 0
```

Definition at line 373 of file [Architecture.h](#).

2.1.2.29 KHAOS_ARCH_m68k1

```
#define KHAOS_ARCH_m68k1 0
```

Definition at line 370 of file [Architecture.h](#).

2.1.2.30 KHAOS_ARCH_m68k2

```
#define KHAOS_ARCH_m68k2 0
```

Definition at line 367 of file [Architecture.h](#).

2.1.2.31 KHAOS_ARCH_m68k3

```
#define KHAOS_ARCH_m68k3 0
```

Definition at line 364 of file [Architecture.h](#).

2.1.2.32 KHAOS_ARCH_m68k4

```
#define KHAOS_ARCH_m68k4 0
```

Definition at line 361 of file [Architecture.h](#).

2.1.2.33 KHAOS_ARCH_m68k6

```
#define KHAOS_ARCH_m68k6 0
```

Definition at line 358 of file [Architecture.h](#).

2.1.2.34 KHAOS_ARCH_MIPS

```
#define KHAOS_ARCH_MIPS 0
```

Definition at line 376 of file [Architecture.h](#).

2.1.2.35 KHAOS_ARCH_MIPS1

```
#define KHAOS_ARCH_MIPS1 0
```

Definition at line 379 of file [Architecture.h](#).

2.1.2.36 KHAOS_ARCH_MIPS2

```
#define KHAOS_ARCH_MIPS2 0
```

Definition at line 382 of file [Architecture.h](#).

2.1.2.37 KHAOS_ARCH_MIPS3

```
#define KHAOS_ARCH_MIPS3 0
```

Definition at line 385 of file [Architecture.h](#).

2.1.2.38 KHAOS_ARCH_MIPS4

```
#define KHAOS_ARCH_MIPS4 0
```

Definition at line 388 of file [Architecture.h](#).

2.1.2.39 KHAOS_ARCH_PARISC

```
#define KHAOS_ARCH_PARISC 0
```

Definition at line 391 of file [Architecture.h](#).

2.1.2.40 KHAOS_ARCH_PARISC1

```
#define KHAOS_ARCH_PARISC1 0
```

Definition at line 394 of file [Architecture.h](#).

2.1.2.41 KHAOS_ARCH_PARISC1_1

```
#define KHAOS_ARCH_PARISC1_1 0
```

Definition at line 397 of file [Architecture.h](#).

2.1.2.42 KHAOS_ARCH_PARISC2

```
#define KHAOS_ARCH_PARISC2 0
```

Definition at line 400 of file [Architecture.h](#).

2.1.2.43 KHAOS_ARCH_PowerPC

```
#define KHAOS_ARCH_PowerPC 0
```

Definition at line 403 of file [Architecture.h](#).

2.1.2.44 KHAOS_ARCH_PowerPC6_1

```
#define KHAOS_ARCH_PowerPC6_1 0
```

Definition at line 406 of file [Architecture.h](#).

2.1.2.45 KHAOS_ARCH_PowerPC6_3

```
#define KHAOS_ARCH_PowerPC6_3 0
```

Definition at line 409 of file [Architecture.h](#).

2.1.2.46 KHAOS_ARCH_PowerPC6_4

```
#define KHAOS_ARCH_PowerPC6_4 0
```

Definition at line 412 of file [Architecture.h](#).

2.1.2.47 KHAOS_ARCH_ptx

```
#define KHAOS_ARCH_ptx 0
```

Definition at line 415 of file [Architecture.h](#).

2.1.2.48 KHAOS_ARCH_Pyramid

```
#define KHAOS_ARCH_Pyramid 0
```

Definition at line 418 of file [Architecture.h](#).

2.1.2.49 KHAOS_ARCH_RISCV

```
#define KHAOS_ARCH_RISCV 0
```

Definition at line 421 of file [Architecture.h](#).

2.1.2.50 KHAOS_ARCH_RS6000

```
#define KHAOS_ARCH_RS6000 0
```

Definition at line 424 of file [Architecture.h](#).

2.1.2.51 KHAOS_ARCH_sh

```
#define KHAOS_ARCH_sh 0
```

Definition at line 436 of file [Architecture.h](#).

2.1.2.52 KHAOS_ARCH_sh1

```
#define KHAOS_ARCH_sh1 0
```

Definition at line 451 of file [Architecture.h](#).

2.1.2.53 KHAOS_ARCH_sh2

```
#define KHAOS_ARCH_sh2 0
```

Definition at line 448 of file [Architecture.h](#).

2.1.2.54 KHAOS_ARCH_sh3

```
#define KHAOS_ARCH_sh3 0
```

Definition at line 445 of file [Architecture.h](#).

2.1.2.55 KHAOS_ARCH_sh4

```
#define KHAOS_ARCH_sh4 0
```

Definition at line 442 of file [Architecture.h](#).

2.1.2.56 KHAOS_ARCH_sh5

```
#define KHAOS_ARCH_sh5 0
```

Definition at line 439 of file [Architecture.h](#).

2.1.2.57 KHAOS_ARCH_SPARC

```
#define KHAOS_ARCH_SPARC 0
```

Definition at line 427 of file [Architecture.h](#).

2.1.2.58 KHAOS_ARCH_SPARC8

```
#define KHAOS_ARCH_SPARC8 0
```

Definition at line 433 of file [Architecture.h](#).

2.1.2.59 KHAOS_ARCH_SPARC9

```
#define KHAOS_ARCH_SPARC9 0
```

Definition at line 430 of file [Architecture.h](#).

2.1.2.60 KHAOS_ARCH_SYS370

```
#define KHAOS_ARCH_SYS370 0
```

Definition at line 454 of file [Architecture.h](#).

2.1.2.61 KHAOS_ARCH_SYS390

```
#define KHAOS_ARCH_SYS390 0
```

Definition at line 457 of file [Architecture.h](#).

2.1.2.62 KHAOS_ARCH_VERSION

```
#define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(0, 0, 0)
```

Definition at line 485 of file [Architecture.h](#).

2.1.2.63 KHAOS_ARCH_VERSION_TWEAK

```
#define KHAOS_ARCH_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)
```

Definition at line 489 of file [Architecture.h](#).

2.1.2.64 KHAOS_ARCH_X86_32

```
#define KHAOS_ARCH_X86_32 0
```

Definition at line 463 of file [Architecture.h](#).

2.1.2.65 KHAOS_ARCH_x86_64

```
#define KHAOS_ARCH_x86_64 0
```

Definition at line 478 of file [Architecture.h](#).

2.1.2.66 KHAOS_ARCH_z

```
#define KHAOS_ARCH_z 0
```

Definition at line 460 of file [Architecture.h](#).

2.2 Architecture.h

[Go to the documentation of this file.](#)

```
00001 #ifndef KHAOS_ARCHITECTURE_H
00002 #define KHAOS_ARCHITECTURE_H
00003
00004 #include "VersionMacros.h"
00005
00094 #define ARCHITECTURE_IS(architecture) (KHAOS_ARCH_##architecture >= 1)
00095
00096 #if defined(__alpha__) || defined(__alpha) || defined(_M_ALPHA)
00097     #define KHAOS_ARCH_Alpha 1
00098     #if defined(__alpha_ev4__)
00099         #define KHAOS_ARCH_AlphaEV4 1
00100         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(4, 0, 0)
00101     #elif defined(__alpha_ev5__)
00102         #define KHAOS_ARCH_AlphaEV5 1
00103         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(5, 0, 0)
00104     #elif defined(__alpha_ev6__)
00105         #define KHAOS_ARCH_AlphaEV6 1
00106         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(6, 0, 0)
00107     #endif
00108 #elif defined(__ARM_ARCH) || defined(__TARGET_ARCH_ARM) || defined(__TARGET_ARCH_THUMB) ||
defined(__M_ARM) || defined(__arm__) || defined(__thumb__) || defined(__ARM_ARCH_7__) ||
defined(__ARM_ARCH_7A__) || defined(__ARM_ARCH_7R__) \
00109 || defined(__ARM_ARCH_7M__) || defined(__ARM_ARCH_6K__) || defined(__ARM_ARCH_6Z__) ||
defined(__ARM_ARCH_6KZ__) || defined(__ARM_ARCH_6T2__) || defined(__ARM_ARCH_5TE__) ||
defined(__ARM_ARCH_5TEJ__) || defined(__ARM_ARCH_4T__) \
00110 || defined(__ARM_ARCH_4__)
00111     #define KHAOS_ARCH_ARM 1
00112     #if defined(__ARM_ARCH_7__) || defined(__ARM_ARCH_7A__) || defined(__ARM_ARCH_7R__) ||
defined(__ARM_ARCH_7M__)
00113         #define KHAOS_ARCH_ARM7 1
00114         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(7, 0, 0)
00115     #elif defined(__ARM_ARCH_6K__) || defined(__ARM_ARCH_6Z__) || defined(__ARM_ARCH_6KZ__) ||
defined(__ARM_ARCH_6T2__)
00116         #define KHAOS_ARCH_ARM6 1
00117         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(6, 0, 0)
00118     #elif defined(__ARM_ARCH_5TE__) || defined(__ARM_ARCH_5TEJ__)
00119         #define KHAOS_ARCH_ARM5 1
00120         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(5, 0, 0)
00121     #elif defined(__ARM_ARCH_4T__) || defined(__ARM_ARCH_4__)
00122         #define KHAOS_ARCH_ARM4 1
00123         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(4, 0, 0)
00124     #elif defined(__ARM_ARCH)
00125         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(__ARM_ARCH, 0, 0)
00126     #elif defined(__TARGET_ARCH_ARM)
00127         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(__TARGET_ARCH_ARM, 0, 0)
00128     #elif defined(__TARGET_ARCH_THUMB)
00129         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(__TARGET_ARCH_THUMB, 0, 0)
00130     #elif defined(_M_ARM)
00131         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(_M_ARM, 0, 0)
00132     #endif
00133 #elif defined(__arm64) || defined(_M_ARM64) || defined(__aarch64__) || defined(__AARCH64EL__)
00134     #define KHAOS_ARCH_ARM8 1
00135     #define KHAOS_ARCH_aarch64 1
00136     #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(8, 0, 0)
00137 #elif defined(__bfin__) || defined(__BFIN__) || defined(bfin) || defined(BFIN)
```

```

00138 #define KHAOS_ARCH_Blackfin 1
00139 #elif defined(__convex__)
00140 #define KHAOS_ARCH_Convex 1
00141 #if defined(__convex_c1__)
00142 #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(1, 0, 0)
00143 #define KHAOS_ARCH_Convex1 1
00144 #elif defined(__convex_c2__)
00145 #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(2, 0, 0)
00146 #define KHAOS_ARCH_Convex2 1
00147 #elif defined(__convex_c32__)
00148 #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(3, 2, 0)
00149 #define KHAOS_ARCH_Convex3_2 1
00150 #elif defined(__convex_c34__)
00151 #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(3, 4, 0)
00152 #define KHAOS_ARCH_Convex3_4 1
00153 #elif defined(__convex_c38__)
00154 #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(3, 8, 0)
00155 #define KHAOS_ARCH_Convex3_8 1
00156 #endif
00157 #elif defined(__e2k__)
00158 #define KHAOS_ARCH_E2K 1
00159 #if defined(__iset__)
00160 #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(__iset__, 0, 0)
00161 #endif
00162 #elif defined(__ia64__) || defined(_IA64) || defined(__IA64__) || defined(__ia64) || defined(_M_IA64)
    || defined(__itanium__)
00163 #define KHAOS_ARCH_IA64 1
00164 #elif defined(__m68k__) || defined(M68000) || defined(__MC68K__)
00165 #define KHAOS_ARCH_m68k 1
00166 #if defined(__mc68060__) || defined(mc68060) || defined(__mc68060)
00167 #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(6, 0, 0)
00168 #define KHAOS_ARCH_m68k6 1
00169 #elif defined(__mc68040__) || defined(mc68040) || defined(__mc68040)
00170 #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(4, 0, 0)
00171 #define KHAOS_ARCH_m68k4 1
00172 #elif defined(__mc68030__) || defined(mc68030) || defined(__mc68030)
00173 #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(3, 0, 0)
00174 #define KHAOS_ARCH_m68k3 1
00175 #elif defined(__mc68020__) || defined(mc68020) || defined(__mc68020)
00176 #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(2, 0, 0)
00177 #define KHAOS_ARCH_m68k2 1
00178 #elif defined(__mc68010__) || defined(mc68010) || defined(__mc68010)
00179 #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(1, 0, 0)
00180 #define KHAOS_ARCH_m68k1 1
00181 #elif defined(__mc68000__) || defined(mc68000) || defined(__mc68000)
00182 #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(0, 0, 0)
00183 #define KHAOS_ARCH_m68k0 1
00184 #endif
00185 #elif defined(__mips__) || defined(__mips) || defined(__MIPS__) || defined(mips)
00186 #define KHAOS_ARCH_MIPS 1
00187 #if defined(_MIPS_ISA_MIPS1) || defined(_R3000)
00188 #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(1, 0, 0)
00189 #define KHAOS_ARCH_MIPS1 1
00190 #elif defined(_MIPS_ISA_MIPS2) || defined(__MIPS_ISA2__) || defined(_R4000)
00191 #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(2, 0, 0)
00192 #define KHAOS_ARCH_MIPS2 1
00193 #elif defined(_MIPS_ISA_MIPS3) || defined(__MIPS_ISA3__)
00194 #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(3, 0, 0)
00195 #define KHAOS_ARCH_MIPS3 1
00196 #elif defined(_MIPS_ISA_MIPS4) || defined(__MIPS_ISA4__)
00197 #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(4, 0, 0)
00198 #define KHAOS_ARCH_MIPS4 1
00199 #elif defined(__mips)
00200 #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(__mips, 0, 0)
00201 #endif
00202 #elif defined(__hppa__) || defined(__hppa) || defined(__HPPA__)
00203 #define KHAOS_ARCH_PARISC 1
00204 #if defined(_PA_RISC1_0)
00205 #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(1, 0, 0)
00206 #define KHAOS_ARCH_PARISC1 1
00207 #elif defined(_PA_RISC1_1) || defined(__HPPA11__) || defined(__PA7100__)
00208 #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(1, 1, 0)
00209 #define KHAOS_ARCH_PARISC1_1 1
00210 #elif defined(_PA_RISC2_0) || defined(_RISC2_0__) || defined(__HPPA20__) || defined(__PA8000__)
00211 #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(2, 0, 0)
00212 #define KHAOS_ARCH_PARISC2 1
00213 #endif
00214 #elif defined(__powerpc) || defined(__powerpc__) || defined(__powerpc64__) || defined(__POWERPC__) ||
    defined(__ppc__) || defined(__ppc64__) || defined(__PPC__) || defined(__PPC64__) || defined(_M_PPC)
    || defined(_ARCH_PPC) || defined(_ARCH_PPC64) \
00215 || defined(__PPCGECKO__) || defined(__PPCBROADWAY__) || defined(_XENON) || defined(__ppc)
00216 #define KHAOS_ARCH_PowerPC 1
00217 #if defined(__ppc601__) || defined(_ARCH_601) || defined(_ARCH_601)
00218 #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(6, 1, 0)
00219 #define KHAOS_ARCH_PowerPC6_1 1
00220 #elif defined(__ppc603__) || defined(_ARCH_603) || defined(_ARCH_603)
00221 #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(6, 3, 0)

```



```

00222     #define KHAOS_ARCH_PowerPC6_3 1
00223     #elif defined(__ppc604__) || defined(__ppc604__) || defined(_ARCH_604)
00224     #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(6, 4, 0)
00225     #define KHAOS_ARCH_PowerPC6_4 1
00226     #endif
00227 #elif defined(__CUDA_ARCH__)
00228     #define KHAOS_ARCH_ptx 1
00229     #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(__CUDA_ARCH__ / 100, __CUDA_ARCH__ / 10, 0)
00230 #elif defined(pyr)
00231     #define KHAOS_ARCH_Pyramid 1
00232 #elif defined(__riscv)
00233     #define KHAOS_ARCH_RISCV 1
00234 #elif defined(__THW_RS6000) || defined(_IBMR2) || defined(_POWER) || defined(_ARCH_PWR) ||
    defined(_ARCH_PWR2)
00235     #define KHAOS_ARCH_RS6000 1
00236 #elif defined(__sparc__) || defined(__sparc)
00237     #define KHAOS_ARCH_SPARC 1
00238     #if defined(__sparcv9) || defined(__sparc_v9__)
00239         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(9, 0, 0)
00240         #define KHAOS_ARCH_SPARC9 1
00241     #elif defined(__sparcv8) || defined(__sparc_v8__)
00242         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(8, 0, 0)
00243         #define KHAOS_ARCH_SPARC8 1
00244     #endif
00245 #elif defined(__sh__)
00246     #define KHAOS_ARCH_sh 1
00247     #if defined(__SH5__)
00248         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(5, 0, 0)
00249         #define KHAOS_ARCH_SH5 1
00250     #elif defined(__SH4__)
00251         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(4, 0, 0)
00252         #define KHAOS_ARCH_SH4 1
00253     #elif defined(__sh3__) || defined(__SH3__)
00254         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(3, 0, 0)
00255         #define KHAOS_ARCH_SH3 1
00256     #elif defined(__sh2__)
00257         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(2, 0, 0)
00258         #define KHAOS_ARCH_SH2 1
00259     #elif defined(__sh1__)
00260         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(1, 0, 0)
00261         #define KHAOS_ARCH_SH1 1
00262     #endif
00263 #elif defined(__370__) || defined(__THW_370__)
00264     #define KHAOS_ARCH_SYS370 1
00265 #elif defined(__s390__) || defined(__s390x__)
00266     #define KHAOS_ARCH_SYS390 1
00267 #elif defined(__SYSC_ZARCH__)
00268     #define KHAOS_ARCH_z 1
00269 #elif defined(i386) || defined(__i386__) || defined(__i486__) || defined(__i586__) ||
    defined(__i686__) || defined(__i386) || defined(_M_IX86) || defined(_X86_) || defined(__THW_INTEL__)
    || defined(__I86__) || defined(__INTEL__)
00270     #define KHAOS_ARCH_X86_32 1
00271     #if defined(__i686__)
00272         #define KHAOS_ARCH_i686 1
00273         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(6, 0, 0)
00274     #elif defined(__i586__)
00275         #define KHAOS_ARCH_i586 1
00276         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(5, 0, 0)
00277     #elif defined(__i486__)
00278         #define KHAOS_ARCH_i486 1
00279         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(4, 0, 0)
00280     #elif defined(__i386__)
00281         #define KHAOS_ARCH_i386 1
00282         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(3, 0, 0)
00283     #elif defined(__I86__)
00284         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(__I86__, 0, 0)
00285     #elif defined(_M_IX86)
00286         #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(_M_IX86 / 100, 0, 0)
00287     #endif
00288 #elif defined(__x86_64) || defined(__x86_64__) || defined(__amd64__) || defined(__amd64) ||
    defined(_M_X64)
00289     #define KHAOS_ARCH_x86_64 1
00290 #elif defined(__epiphany__)
00291     #define KHAOS_ARCH_Epiphany 1
00292 #endif
00293
00294 #if !defined(KHAOS_ARCH_Alpha)
00295     #define KHAOS_ARCH_Alpha 0
00296 #endif
00297 #if !defined(KHAOS_ARCH_AlphaEV4)
00298     #define KHAOS_ARCH_AlphaEV4 0
00299 #endif
00300 #if !defined(KHAOS_ARCH_AlphaEV5)
00301     #define KHAOS_ARCH_AlphaEV5 0
00302 #endif
00303 #if !defined(KHAOS_ARCH_AlphaEV6)
00304     #define KHAOS_ARCH_AlphaEV6 0

```

```
00305 #endif
00306 #if !defined(KHAOS_ARCH_ARM)
00307     #define KHAOS_ARCH_ARM 0
00308 #endif
00309 #if !defined(KHAOS_ARCH_ARM7)
00310     #define KHAOS_ARCH_ARM7 0
00311 #endif
00312 #if !defined(KHAOS_ARCH_ARM6)
00313     #define KHAOS_ARCH_ARM6 0
00314 #endif
00315 #if !defined(KHAOS_ARCH_ARM5)
00316     #define KHAOS_ARCH_ARM5 0
00317 #endif
00318 #if !defined(KHAOS_ARCH_ARM4)
00319     #define KHAOS_ARCH_ARM4 0
00320 #endif
00321 #if !defined(KHAOS_ARCH_ARM8)
00322     #define KHAOS_ARCH_ARM8 0
00323 #endif
00324 #if !defined(KHAOS_ARCH_aarch64)
00325     #define KHAOS_ARCH_aarch64 0
00326 #endif
00327 #if !defined(KHAOS_ARCH_Blackfin)
00328     #define KHAOS_ARCH_Blackfin 0
00329 #endif
00330 #if !defined(KHAOS_ARCH_Convex)
00331     #define KHAOS_ARCH_Convex 0
00332 #endif
00333 #if !defined(KHAOS_ARCH_Convex1)
00334     #define KHAOS_ARCH_Convex1 0
00335 #endif
00336 #if !defined(KHAOS_ARCH_Convex2)
00337     #define KHAOS_ARCH_Convex2 0
00338 #endif
00339 #if !defined(KHAOS_ARCH_Convex3_2)
00340     #define KHAOS_ARCH_Convex3_2 0
00341 #endif
00342 #if !defined(KHAOS_ARCH_Convex3_4)
00343     #define KHAOS_ARCH_Convex3_4 0
00344 #endif
00345 #if !defined(KHAOS_ARCH_Convex3_8)
00346     #define KHAOS_ARCH_Convex3_8 0
00347 #endif
00348 #if !defined(KHAOS_ARCH_E2K)
00349     #define KHAOS_ARCH_E2K 0
00350 #endif
00351 #if !defined(KHAOS_ARCH_IA64)
00352     #define KHAOS_ARCH_IA64 0
00353 #endif
00354 #if !defined(KHAOS_ARCH_m68k)
00355     #define KHAOS_ARCH_m68k 0
00356 #endif
00357 #if !defined(KHAOS_ARCH_m68k6)
00358     #define KHAOS_ARCH_m68k6 0
00359 #endif
00360 #if !defined(KHAOS_ARCH_m68k4)
00361     #define KHAOS_ARCH_m68k4 0
00362 #endif
00363 #if !defined(KHAOS_ARCH_m68k3)
00364     #define KHAOS_ARCH_m68k3 0
00365 #endif
00366 #if !defined(KHAOS_ARCH_m68k2)
00367     #define KHAOS_ARCH_m68k2 0
00368 #endif
00369 #if !defined(KHAOS_ARCH_m68k1)
00370     #define KHAOS_ARCH_m68k1 0
00371 #endif
00372 #if !defined(KHAOS_ARCH_m68k0)
00373     #define KHAOS_ARCH_m68k0 0
00374 #endif
00375 #if !defined(KHAOS_ARCH_MIPS)
00376     #define KHAOS_ARCH_MIPS 0
00377 #endif
00378 #if !defined(KHAOS_ARCH_MIPS1)
00379     #define KHAOS_ARCH_MIPS1 0
00380 #endif
00381 #if !defined(KHAOS_ARCH_MIPS2)
00382     #define KHAOS_ARCH_MIPS2 0
00383 #endif
00384 #if !defined(KHAOS_ARCH_MIPS3)
00385     #define KHAOS_ARCH_MIPS3 0
00386 #endif
00387 #if !defined(KHAOS_ARCH_MIPS4)
00388     #define KHAOS_ARCH_MIPS4 0
00389 #endif
00390 #if !defined(KHAOS_ARCH_PARISC)
00391     #define KHAOS_ARCH_PARISC 0
```

```
00392 #endif
00393 #if !defined(KHAOS_ARCH_PARISC1)
00394     #define KHAOS_ARCH_PARISC1 0
00395 #endif
00396 #if !defined(KHAOS_ARCH_PARISC1_1)
00397     #define KHAOS_ARCH_PARISC1_1 0
00398 #endif
00399 #if !defined(KHAOS_ARCH_PARISC2)
00400     #define KHAOS_ARCH_PARISC2 0
00401 #endif
00402 #if !defined(KHAOS_ARCH_PowerPC)
00403     #define KHAOS_ARCH_PowerPC 0
00404 #endif
00405 #if !defined(KHAOS_ARCH_PowerPC6_1)
00406     #define KHAOS_ARCH_PowerPC6_1 0
00407 #endif
00408 #if !defined(KHAOS_ARCH_PowerPC6_3)
00409     #define KHAOS_ARCH_PowerPC6_3 0
00410 #endif
00411 #if !defined(KHAOS_ARCH_PowerPC6_4)
00412     #define KHAOS_ARCH_PowerPC6_4 0
00413 #endif
00414 #if !defined(KHAOS_ARCH_ptx)
00415     #define KHAOS_ARCH_ptx 0
00416 #endif
00417 #if !defined(KHAOS_ARCH_Pyramid)
00418     #define KHAOS_ARCH_Pyramid 0
00419 #endif
00420 #if !defined(KHAOS_ARCH_RISCV)
00421     #define KHAOS_ARCH_RISCV 0
00422 #endif
00423 #if !defined(KHAOS_ARCH_RS6000)
00424     #define KHAOS_ARCH_RS6000 0
00425 #endif
00426 #if !defined(KHAOS_ARCH_SPARC)
00427     #define KHAOS_ARCH_SPARC 0
00428 #endif
00429 #if !defined(KHAOS_ARCH_SPARC9)
00430     #define KHAOS_ARCH_SPARC9 0
00431 #endif
00432 #if !defined(KHAOS_ARCH_SPARC8)
00433     #define KHAOS_ARCH_SPARC8 0
00434 #endif
00435 #if !defined(KHAOS_ARCH_sh)
00436     #define KHAOS_ARCH_sh 0
00437 #endif
00438 #if !defined(KHAOS_ARCH_sh5)
00439     #define KHAOS_ARCH_sh5 0
00440 #endif
00441 #if !defined(KHAOS_ARCH_sh4)
00442     #define KHAOS_ARCH_sh4 0
00443 #endif
00444 #if !defined(KHAOS_ARCH_sh3)
00445     #define KHAOS_ARCH_sh3 0
00446 #endif
00447 #if !defined(KHAOS_ARCH_sh2)
00448     #define KHAOS_ARCH_sh2 0
00449 #endif
00450 #if !defined(KHAOS_ARCH_sh1)
00451     #define KHAOS_ARCH_sh1 0
00452 #endif
00453 #if !defined(KHAOS_ARCH_SYS370)
00454     #define KHAOS_ARCH_SYS370 0
00455 #endif
00456 #if !defined(KHAOS_ARCH_SYS390)
00457     #define KHAOS_ARCH_SYS390 0
00458 #endif
00459 #if !defined(KHAOS_ARCH_z)
00460     #define KHAOS_ARCH_z 0
00461 #endif
00462 #if !defined(KHAOS_ARCH_X86_32)
00463     #define KHAOS_ARCH_X86_32 0
00464 #endif
00465 #if !defined(KHAOS_ARCH_i686)
00466     #define KHAOS_ARCH_i686 0
00467 #endif
00468 #if !defined(KHAOS_ARCH_i586)
00469     #define KHAOS_ARCH_i586 0
00470 #endif
00471 #if !defined(KHAOS_ARCH_i486)
00472     #define KHAOS_ARCH_i486 0
00473 #endif
00474 #if !defined(KHAOS_ARCH_i386)
00475     #define KHAOS_ARCH_i386 0
00476 #endif
00477 #if !defined(KHAOS_ARCH_x86_64)
00478     #define KHAOS_ARCH_x86_64 0
```

```
00479 #endif
00480 #if !defined(KHAOS_ARCH_Epiphany)
00481     #define KHAOS_ARCH_Epiphany 0
00482 #endif
00483
00484 #if !defined(KHAOS_ARCH_VERSION)
00485     #define KHAOS_ARCH_VERSION KHAOS_SET_VERSION(0, 0, 0)
00486 #endif
00487
00488 #if !defined(KHAOS_ARCH_VERSION_TWEAK)
00489     #define KHAOS_ARCH_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)
00490 #endif
00491
00492 #endif
```

2.3 khaos/C.h File Reference

Detect the C standard and define some useful macros.

```
#include "CStandards.h"
```

Macros

- #define [KHAOS_LANGUAGE_ISO](#) 1
1 if strict ISO, 0 otherwise
- #define [KHAOS_LANGUAGE_Extensions](#) 0
0 if strict ISO, 1 otherwise

2.3.1 Detailed Description

Detect the C standard and define some useful macros.

Definition in file [C.h](#).

2.3.2 Macro Definition Documentation

2.3.2.1 KHAOS_LANGUAGE_Extensions

```
#define KHAOS_LANGUAGE_Extensions 0
```

0 if strict ISO, 1 otherwise

Definition at line [33](#) of file [C.h](#).

2.3.2.2 KHAOS_LANGUAGE_ISO

```
#define KHAOS_LANGUAGE_ISO 1
```

1 if strict ISO, 0 otherwise

Definition at line 32 of file C.h.

2.4 C.h

[Go to the documentation of this file.](#)

```
00001 #ifndef KHAOS_C_H
00002 #define KHAOS_C_H
00003
00008 #include "CStandards.h"
00009
00010 #if defined(__STDC__)
00011     #if defined(__STDC_VERSION__)
00012         #define KHAOS_C_STANDARD __STDC_VERSION__
00013     #else
00014         #define KHAOS_C_STANDARD KHAOS_STANDARD_C90
00015     #endif
00016 #else
00017     /* For GCC this should never happen
00018        https://gcc.gnu.org/onlinedocs/cpp/Traditional-Mode.html#Traditional-Mode */
00018     #if defined(_MSC_VER)
00019         /*
00020          * The compiler uses C compilation mode by default when a source file has a .c extension, or when
00021          * the /Tc or /TC option is specified.
00022          * The C compiler is an C89/C90 compiler that, by default, enables Microsoft extensions to the C
00023          * language. For more information about specific extensions, see Microsoft Extensions to C and C++.
00024          * When both C compilation and the /Za option are specified, the C compiler conforms strictly to
00025          * the C89/C90 standard.
00026          * The compiler treats Microsoft extended keywords as simple identifiers, disables the other
00027          * Microsoft extensions, and automatically defines the __STDC__ predefined macro for C programs.
00028          */
00029         #define KHAOS_C_STANDARD KHAOS_STANDARD_C90
00030     #else
00031         #error "C is detected but __STDC__ is not defined !"
00032     #endif
00033 #endif
00034
00035 #if defined(__STRICT_ANSI__) || !defined(_MSC_EXTENSIONS)
00036     #define KHAOS_LANGUAGE_ISO 1
00037     #define KHAOS_LANGUAGE_Extensions 0
00038 #else
00039     #define KHAOS_LANGUAGE_ISO 0
00040     #define KHAOS_LANGUAGE_Extensions 1
00041 #endif
00042 #endif
```

2.5 khaos/Compiler.h File Reference

Detect the compiler, convert the vendor version number to standardized one if available at compilation time and store it in KHAOS_COMPILER_VERSION.

```
#include "khaos/VersionMacros.h"
```

Macros

- `#define COMPILER_IS(compiler)`
Detect the compiler.
- `#define COMPILER_IS_GREATER_EQUAL(compiler, major, minor, patch) (KHAOS_COMPILER_##compiler >= KHAOS_SET_VERSION(major, minor, patch))`
- `#define COMPILER_IS_LOWER(compiler, major, minor, patch) (KHAOS_COMPILER_##compiler < KHAOS_SET_VERSION(major, minor, patch))`
- `#define KHAOS_COMPILER_NVCC 0`
- `#define KHAOS_NVCC_VERSION KHAOS_SET_VERSION(0, 0, 0)`
- `#define KHAOS_NVCC_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0, 0, 0)`
- `#define KHAOS_COMPILER_Unknown 1`
- `#define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(0, 0, 0)`
- `#define KHAOS_COMPILER_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)`
- `#define KHAOS_COMPILER_ClangEmulated`
Detect the emulated compilers.
- `#define KHAOS_CLANG_EMULATED_VERSION KHAOS_SET_VERSION(0, 0, 0)`
- `#define KHAOS_CLANG_EMULATED_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0, 0, 0)`
- `#define KHAOS_COMPILER_GCCEmulated 0`
- `#define KHAOS_GCC_EMULATED_VERSION KHAOS_SET_VERSION(0, 0, 0)`
- `#define KHAOS_GCC_EMULATED_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)`
- `#define KHAOS_COMPILER_LLVMEmulated 0`
- `#define KHAOS_LLVM_EMULATED_VERSION KHAOS_SET_VERSION(0, 0, 0)`
- `#define KHAOS_LLVM_EMULATED_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)`
- `#define KHAOS_COMPILER_ACC 0`
- `#define KHAOS_COMPILER_AltiumMicroblaze 0`
- `#define KHAOS_COMPILER_Altium 0`
- `#define KHAOS_COMPILER_Amsterdam 0`
- `#define KHAOS_COMPILER_ARM 0`
- `#define KHAOS_COMPILER_Aztec 0`
- `#define KHAOS_COMPILER_Borland 0`
- `#define KHAOS_COMPILER_CC65 0`
- `#define KHAOS_COMPILER_Comeau 0`
- `#define KHAOS_COMPILER_Compacq 0`
- `#define KHAOS_COMPILER_Convex 0`
- `#define KHAOS_COMPILER_Compcert 0`
- `#define KHAOS_COMPILER_Coverity 0`
- `#define KHAOS_COMPILER_CrayC 0`
- `#define KHAOS_COMPILER_Diab 0`
- `#define KHAOS_COMPILER_DICE 0`
- `#define KHAOS_COMPILER_DigitalMars 0`
- `#define KHAOS_COMPILER_Dignus 0`
- `#define KHAOS_COMPILER_DJGPP 0`
- `#define KHAOS_COMPILER_EDG 0`
- `#define KHAOS_COMPILER_EKOPath 0`
- `#define KHAOS_COMPILER_Emscripten 0`
- `#define KHAOS_COMPILER_Fujitsu 0`
- `#define KHAOS_COMPILER_GCCXML 0`
- `#define KHAOS_COMPILER_GreenHills 0`
- `#define KHAOS_COMPILER_HPCC 0`
- `#define KHAOS_COMPILER_HPACC 0`
- `#define KHAOS_COMPILER_IAR 0`
- `#define KHAOS_COMPILER_IBM 0`
- `#define KHAOS_COMPILER_IBMClang 0`

- `#define KHAOS_COMPILER_IBMLegacy 0`
- `#define KHAOS_COMPILER_IBMC370 0`
- `#define KHAOS_COMPILER_IBMMVS 0`
- `#define KHAOS_COMPILER_IBMOS390 0`
- `#define KHAOS_COMPILER_IBMZOS 0`
- `#define KHAOS_COMPILER_ImageCraft 0`
- `#define KHAOS_COMPILER_Intel 0`
- `#define KHAOS_COMPILER_Kai 0`
- `#define KHAOS_COMPILER_KeilCarm 0`
- `#define KHAOS_COMPILER_KeilC166 0`
- `#define KHAOS_COMPILER_KeilC51 0`
- `#define KHAOS_COMPILER_LCC 0`
- `#define KHAOS_COMPILER_Metaware 0`
- `#define KHAOS_COMPILER_Metrowerks 0`
- `#define KHAOS_COMPILER_MSVC 0`
- `#define KHAOS_COMPILER_Microtec 0`
- `#define KHAOS_COMPILER_Microway 0`
- `#define KHAOS_COMPILER_MIPSPRO 0`
- `#define KHAOS_COMPILER_Miracle 0`
- `#define KHAOS_COMPILER_MPW 0`
- `#define KHAOS_COMPILER_Norcroft 0`
- `#define KHAOS_COMPILER_NWCC 0`
- `#define KHAOS_COMPILER_Open64 0`
- `#define KHAOS_COMPILER_Oracle 0`
- `#define KHAOS_COMPILER_Solaris 0`
- `#define KHAOS_COMPILER_Pacific 0`
- `#define KHAOS_COMPILER_Palm 0`
- `#define KHAOS_COMPILER_Pelles 0`
- `#define KHAOS_COMPILER_Portland 0`
- `#define KHAOS_COMPILER_Renesas 0`
- `#define KHAOS_COMPILER_SASC 0`
- `#define KHAOS_COMPILER_SCO 0`
- `#define KHAOS_COMPILER_SDCC 0`
- `#define KHAOS_COMPILER_SNC 0`
- `#define KHAOS_COMPILER_Stratus 0`
- `#define KHAOS_COMPILER_StratusKR 0`
- `#define KHAOS_COMPILER_StratusANSI 0`
- `#define KHAOS_COMPILER_Symantec 0`
- `#define KHAOS_COMPILER_Tendra 0`
- `#define KHAOS_COMPILER_TI 0`
- `#define KHAOS_COMPILER_Think 0`
- `#define KHAOS_COMPILER_Tiny 0`
- `#define KHAOS_COMPILER_Turbo 0`
- `#define KHAOS_COMPILER_Ultimate 0`
- `#define KHAOS_COMPILER_USLC 0`
- `#define KHAOS_COMPILER_VBCC 0`
- `#define KHAOS_COMPILER_Watcom 0`
- `#define KHAOS_COMPILER_Zortech 0`
- `#define KHAOS_COMPILER_AppleClang 0`
- `#define KHAOS_COMPILER_Clang 0`
- `#define KHAOS_COMPILER_LLVM 0`
- `#define KHAOS_COMPILER_GCC 0`

2.5.1 Detailed Description

Detect the compiler, convert the vendor version number to standardized one if available at compilation time and store it in `KHAOS_COMPILER_VERSION`.

Definition in file [Compiler.h](#).

2.5.2 Macro Definition Documentation

2.5.2.1 COMPILER_IS

```
#define COMPILER_IS(  
    compiler )
```

Detect the compiler.

Parameters

in	<i>compiler</i>	The compiler name.
----	-----------------	--------------------

Note

Use this macro only if you need preprocessor compiler checking. For example :

```
#if COMPILER_IS(gcc)
//...
#else
//...
#endif
```

Format of the version numbers :

The base of the specified version number : **0X** for hexadecimal digits, and **10** for decimal digits.

The format of the vendor version number :

- **V** indicates the version digits.
- **R** indicates the revision digits.
- **P** indicates the patch digits.
- **T** indicates the tweak digits.
- **M** indicates the modification digits.
- **0** indicates an ignored digit.

2.5.3 Detected compilers are :

2.5.3.1 ACC

ACC

Macro	Type
<i>ACC</i>	detection

Note

check with ACC

2.5.3.2 Altium MicroBlaze C*Altium MicroBlaze C*

Macro	Type	Format
CMB	detection	
VERSION	Version	10VRRR
REVISION	Version	10P
BUILD	Version	10VRRRPPP

Note

check with AltiumMicroblaze

2.5.3.3 Altium C-to-Hardware*Altium C-to-hardware*

Macro	Type	Format
CHC	detection	
VERSION	Version	10VRRR
REVISION	Version	10P
BUILD	Version	10VRRRPPP

Note

check with Altium

2.5.3.4 Amsterdam Compiler Kit*Amsterdam Compiler Kit*

Macro	Type
ACK	detection

Note

check with Amsterdam

2.5.3.5 ARM Compiler

ARM Compiler

Macro	Type	Format
<code>__CC_ARM</code>	detection	
<code>__ARMCC_VERSION</code>	Version	10VRPBBB

Note

check with ARM

2.5.3.6 Aztec C

Aztec C

Macro	Type	Format
<code>AZTEC_C</code>	detection	
<code>AZTEC_C</code>	detection	
<code>__VERSION</code>	Version	10VRR

Note

check with Aztec

2.5.3.7 Borland

Borland C++

Macros	Type	Format
<code>__BORLANDC_↔</code> —	detection	
<code>__CODEGEARC_↔</code> —	detection	
<code>__BORLANDC_↔</code> —	version	0XVVRP
<code>__CODEGEARC_↔</code> —	version	0XVVRP

Note

check with Borland

2.5.3.8 CC65

CC65

Macros	Type	Format
<code>__CC65__</code>	detection	
<code>__CC65__</code>	version	0XVRP

Note

check with CC65

2.5.3.9 Clang**Clang**

Macros	Type	Format
<code>__clang__</code>	detection	
<code>__clang_major__</code>	version	V
<code>__clang_minor__</code>	version	R
<code>__clang_patchlevel__</code>	version	P
<code>__clang_version__</code>	version	V.R.P (tags/RELEASE_VRP/final)

Note

check with Clang (always defined)

check with AppleClang (only for Apple)

2.5.3.10 Comeau C++**Comeau C++**

Macros	Type	Format
<code>__COMO__</code>	detection	
<code>__COMO_VERSION__</code>	version	10VRR

Note

check with Comeau

2.5.3.11 Compaq C/C++**Compaq C/C++**

Macros	Type	Format
<code>__DECC</code>	detection	C compiler

Macros	Type	Format
__DECC_VER	version	10_VVRRTPPPP
__DECCXX	detection	C++ compiler
__DECCXX_VER	version	10_VVRRTPPPP
__VAXC		Obsolete
VAXC		Obsolete

Note

check with Compaq

2.5.3.12 Convex C

Convex C

Macro	Type
convexc	detection

Note

check with Convex

2.5.3.13 CompCert

CompCert

Macro	Type
COMPCERT	detection

Note

check with CompCert

2.5.3.14 Coverity

Coverity

Macro	Type
COVERITY	detection

Note

check with Coverity

2.5.3.15 Cray C

Cray C

Macros	Type	Format
<code>_CRAYC</code>	detection	
<code>_RELEASE</code>	version	10V
<code>_RELEASE_MINOR</code>	version	10R

Note

check with CrayC

2.5.3.16 Diab C/C++

Diab C/C++

Macros	Type	Format
<code>__DCC__</code>	detection	
<code>__VERSION_NUMBER↔</code> <code>__</code>	version	10_VRPP

Note

check with Diab

2.5.3.17 DICE C

DICE C

Macro	Type
<code>_DICE</code>	detection

Note

check with DICE

2.5.3.18 Digital mars

Digital Mars

Macros	Type	Format
<code>__DMC↔</code> <code>__</code>	detection	
<code>__DMC↔</code> <code>__</code>	version	0XVRP

Note

check with DigitalMars

2.5.3.19 Dignus Systems/C++

Dignus Systems/C++

Macros	Type	Format
__SYSC__	detection	
__SYSC_VER↔ __	version	10_VRRPP

Note

check with Dignus

2.5.3.20 DJGPP

DJGPP

Macros	Type	Format
__DJGPP__	detection	
__DJGPP__	version	10V
__DJGPP_MINOR↔ __	version	10R
__GO32__	detection	Obsolete

Note

check with DJGPP

2.5.3.21 EDG C++ Frontend

EDG C++ Frontend

Macros	Type	Format
__EDG__	detection	
__EDG_VERSION↔ __	version	10_VRR

Note

check with EDG

2.5.3.22 EKOPath

EKOPath

Macro	Type	Format
PATHCC	detection	

Macro	Type	Format
PATHCC	Version	V
PATHCC_MINOR	Version	R
PATHCC_PATCHLEVEL	Version	P

Note

check with EKOPath

2.5.3.23 Emscripten

Emscripten

Macro	Type	Format
EMSCRIPTEN	detection	
EMSCRIPTEN_major	Version	V
EMSCRIPTEN_minor	Version	R
EMSCRIPTEN_tiny	Version	P

Note

check with Emscripten

2.5.3.24 Fujitsu C++

Fujitsu C++

Macro	Type
__FCC_VERSION	detection

Note

check with Fujitsu

2.5.3.25 Gnu GCC C/C++

Gnu GCC C/C++

Macros	Type	Value
__GNUC__	detection	
__GNUC__	version	V
__GNUC_MINOR__	version	R
__GNUC_PATCHLEVEL__	version	P

Note

check with GCC

2.5.3.26 GCC XML

GCC XML

Macro	Type
GCCXML	detection

Note

check with GCCXML

2.5.3.27 Green Hills

Green Hills C/C++

Macros	Type	Value
<code>__ghs</code>	detection	
<code>__ghs__</code>	detection	
<code>__GHS_VERSION_NUMBER</code> ↔ <code>__</code>	version	10_VRP
<code>__GHS_REVISION_DATE</code> ↔ <code>__</code>	version	epoch time
<code>__ghs</code>	version	10_VRP

Note

check with GreenHills

2.5.3.28 HP ANSI C

HP ANSI C

Macro	Type
__HP_cc	detection

Note

check with HPCC

2.5.3.29 HP aC++ compiler

HP aC++

Macros	Type	Value
<code>__HP_aCC</code>	detection	
<code>__HP_aCC</code>	version	10_VVRRPP

Note

check with HPaCC

2.5.3.30 IAR C/C++

IAR C/C++

Macros	Type	Value
<code>__IAR_SYSTEMS_ICC↔</code> —	detection	
<code>__VER__</code>	version	10_VVRR

Note

check with IAR

2.5.3.31 IBM XL C/C++ (Clang-based versions)

IBM XL C/C++ (Clang-based versions)

Macros	Type	Value
<code>__ibmxl__ and __clang__</code>	detection	
<code>__ibmxl_vrm__</code>	version	0xVVRRMM00
<code>__ibmxl_version__</code>	version	V
<code>__ibmxl_release__</code>	version	R
<code>__ibmxl_modification↔</code> —	version	M
<code>__ibmxl_ptf_fix_level↔</code> —	version	F

Note

check with IBMClang or IBM

2.5.3.32 IBM XL C/C++ (legacy)

IBM XL C/C++

Macros	Type	Value
<code>__xlC__</code>	detection	
<code>__IBMC__</code>	version	0X_VRRMM

Macros	Type	Value
<code>__IBMCPP__↔</code> —	version	0X_VRRMM
<code>__xlC__</code>	version	0xVRRMMFF
<code>__xlC__</code>	version	0xVRRR
<code>__xlC_ver↔</code> —	version	0x0000MMFF

Note

check with IBM or IBMLegacy

2.5.3.33 IBM z/OS XL C/C++

IBM z/OS XL C/C++

Macros	Type	Value
<code>__IBMC__</code>	detection	
<code>__IBMCPP__</code>	detection	
<code>__IBMC__</code>	version	NVRRM N = Product (0 = C/370, 1 = MVS, 2 = OS/390, 4 = z/OS) V = Version RR = Revision M = Modification
<code>__IBMCPP__</code>	version	N = Product (0 = C/370, 1 = MVS, 2 = OS/390, 4 = z/OS) V = Version RR = Revision M = Modification
<code>__COMPILER_VER↔</code> —	version	0xNVRRMMFF N = Product (see above) V = Version RR = Revision MM = Modification FF = Fix Pack

Note

check with IBM or for platform : IBMC370 or IBMMVS or IBMOS390 or IBMZOS

2.5.3.34 ImageCraft C

ImageCraft C

Macro	Type
IMAGECRAFT	detection

Note

ImageCraft

2.5.3.35 Intel C/C++

Intel C/C++

Macros	Type	Value
<code>__INTEL_COMPILER</code>	detection	

Macros	Type	Value
__ICC	detection	Obsolete
__ECC	detection	Obsolete
__ICL	detection	Obsolete
__INTEL_COMPILER	version	VRP
__INTEL_COMPILER_BUILD_DATE	date	YYYYMMDD

Note

check with Intel

2.5.3.36 Kai C++ compiler

Kai C++ compiler

Macros	Type	Value
__KCC	detection	
__KCC_VERSION	version	10_VVRRPP

Note

check with Kai

2.5.3.37 KEIL CARM

KEIL CARM

Macros	Type	Value
__CA__	detection	
__KEIL↔ __	detection	
__CA__	version	10VRR

Note

check with KeilCarm

2.5.3.38 KEIL C166

KEIL C166

Macros	Type	Value
__C166↔ __	detection	
__C166↔ __	version	10VRR

Note

check with KeilC166

2.5.3.39 KEIL C51

KEIL C51

Macros	Type	Value
__C51_↔ —	detection	
__CX51_↔ —	detection	
__C51_↔ —	version	10VRR

Note

check with KeilC51

2.5.3.40 LCC

LCC

Macro	Type
LCC	detection

Note

check with LCC

2.5.3.41 LLVM

LLVM

Macros	Type
__llvm_↔ —	detection

Note

check with LLVM

2.5.3.42 MetaWare High C/C++

MetaWare High C/C++

Macros	Type
__HIGHC↔ __	detection

Note

check with MetaWare

2.5.3.43 Metrowerks CodeWarrior

Metrowerks CodeWarrior

Macros	Type	Value
__MWERKS↔ __	detection	
__CWCC__	detection	
__CWCC__	version	0X_VRPP
__MWERKS↔ __	version	0X_VRPP >= 4.2.0
__MWERKS↔ __	version	MWERKS >= 0x3204 note the "skip": 04->9.3
__MWERKS↔ __	version	MWERKS >= 0x3200
__MWERKS↔ __	version	MWERKS >= 0x3000

Note

check with Metrowerks

2.5.3.44 Microsoft Visual C/C++

Microsoft Visual C/C++

Macros	Type	Value
_MSC_VER	detection	
_MSC_FULL_VER	version	
_MSC_VER	version	
_MSC_BUILD	version	

Note

check with MSVC

2.5.3.45 Microtec C/C++

Microtec C/C++

Macros	Type	Value
<code>_MRI</code>	detection	

Note

check with Microtec

2.5.3.46 Microway NDP C**Microway NDP C**

Macros	Type	Value
<code>__NDPC↔</code> <code>__</code>	detection	
<code>__NDPX↔</code> <code>__</code>	detection	

Note

check with Microway

2.5.3.47 SGI MIPSpro**SGI MIPSpro**

Macros	Type	Value
<code>__sgi</code>	detection	
<code>sgi</code>	detection	
<code>_SGI_COMPILER_VERSION</code>	version	<code>10_VRP</code>
<code>_COMPILER_VERSION</code>	version	<code>10_VRP</code>

Note

check with MIPSpro

2.5.3.48 Miracle C**Miracle C**

Macro	Type
<code>MIRACLE</code>	detection

Note

check with Miracle

2.5.3.49 MPW C++

MPW C++

Macros	Type	Value
__MRC__	detection	
MPW_C	detection	
MPW_CPLUS	detection	
__MRC__	version	0X_VVRR

Note

check with MPW

2.5.3.50 Norcroft C

Norcroft C

Macro	Type	Format
__CC_NORCROFT	detection	
__ARMCC_VERSION	Version	V.R

Note

check with Norcroft

2.5.3.51 NWCC

NWCC

Macro	Type
NWCC	detection

Note

check with NWCC

2.5.3.52 Open64

Open64

Macro	Type	Format
OPEN64	detection	
OPENCC	detection	
OPENCC	Version	V
OPENCC_MINOR	Version	R
OPENCC_PATCHLEVEL	Version	P.B

Note

check with Open64

2.5.3.53 Oracle Pro*C Precompiler**Oracle Pro*C Precompiler**

Macro	Type
ORA_PROC	detection

Note

check with Oracle

2.5.3.54 Oracle Solaris Studio**Oracle Solaris Studio**

Macros	Type	Value
__SUNPRO_CC	detection	
__SUNPRO_C	detection	
__SUNPRO_CC	version	0X_VRP
__SUNPRO_C	version	0X_VRP
__SUNPRO_CC	version	0X_VVRRP
__SUNPRO_C	version	0X_VVRRP

Note

check with Solaris

2.5.3.55 Pacific C

Pacific C

Macro	Type
PACIFIC	detection

Note

check with Pacific

2.5.3.56 Palm C/C++

Macros	Type	Value
_PACC_VER	detection	
_PACC_VER	version	0X_VRRPPBBB

Note

check with Palm

2.5.3.57 Pelles C

Pelles C

Macro	Type	Format
POCC	detection	
POCC	Version	10VRR

Note

check with Pelles

2.5.3.58 Portland Group C/C++

Portland Group C/C++

Macros	Type	Value
__PGI	detection	
__PGI	version	V
__PGIC_MINOR__	version	M
__PGIC_PATCHLEVEL__↔	version	P
__		

Note

check with Portland

2.5.3.59 Renesas C/C++

Renesas C/C++

Macro	Type	Format
RENESAS	detection	
HITACHI	detection	
RENESAS_VERSION	Version	0xVVRR
RENESAS_VERSION	Version	0xVVRRPP00
HITACHI_VERSION	Version	0xVVRR

Note

check with Renesas

2.5.3.60 SAS/C

SAS/C

Macros	Type	Value
SASC	detection	
__SASC	detection	
__SASC__	detection	
__VERSION_↔	version	
— __REVISION_↔ —	version	
__SASC__	version	VRR

Note

check with SASC

2.5.3.61 SCO OpenServer

SCO OpenServer

Macro	Type
_SCO_DS	detection

Note

check with SCO

2.5.3.62 Small Device C Compiler

Small Device C Compiler

Macro	Type	Format
SDCC	detection	
SDCC	Version	VRP

Note

check with SDCC

2.5.3.63 SN Compiler

SN Compiler

Macro	Type
SNC	detection

Note

check with SNC

2.5.3.64 Stratus VOS C

Stratus VOS C

Macro	Type
VOSC	detection 0 = K&R compiler 1 = ANSI C compiler

Note

check with (StratusKR or StratusANSI) or Stratus

2.5.3.65 Symantec C++

Symantec C++

Macros	Type	Value
__SC↔ —	detection	
__SC↔ —	version	0xVRRR

Note

check with Symantec

2.5.3.66 TenDRA C/C++

TenDRA C/C++

Macro	Type
TenDRA	detection

Note

check with Tendra

2.5.3.67 Texas Instruments C/C++ Compiler

Texas Instruments C/C++ Compiler

Macro	Type	Format
TI_COMPILER_VERSION	detection	
_TMS320C6X	detection	
TI_COMPILER_VERSION	Version	10VRR

Note

check with TI

2.5.3.68 THINK C

THINK C

Macro	Type
THINKC3	detection
THINKC4	detection

Note

check with Think

2.5.3.69 Tiny C

Tiny C

Macro	Type
TINYC	detection

Note

check with Tiny

2.5.3.70 Turbo C/C++

Turbo C/C++

Macro	Type	Format
TURBOC	detection	
TURBOC	Version	0xVVRR

Note

check with Turbo

2.5.3.71 Ultimate C/C++

Ultimate C/C++

Macro	Type	Format
_UCC	detection	
_MAJOR_REV	Version	V
_MINOR_REV	Version	M

Note

check with Ultimate

2.5.3.72 USL C

USL C

Macro	Type	Format
USLC	detection	
SCO_VERSION	Version	VRRYYYYMM V = Version RR = Revision YYYY = Year MM = Month

Note

check with USL

2.5.3.73 VBCC

VBCC

Macro	Type
VBCC	detection

Note

check with VBCC

2.5.3.74 Watcom C++

Watcom C++

Macros	Type	Value
__WATCOMC↵ —	detection	
__WATCOMC↵ —	version	10_VVRR

Note

check with Watcom

2.5.3.75 Zortech C++

Zortech C++

Macros	Type	Value
<code>__ZTC↔</code> —	detection	
<code>__ZTC↔</code> —	version	0xVRP

Note

check with Zortech

2.5.3.76 NVCC**NVCC**

Macros	Type	Value
<code>__NVCC__</code>	detection	
<code>__CUDA__VER__MAJOR↔</code> —	version	V
<code>__CUDA__VER__MINOR↔</code> —	version	M
<code>__CUDA__VER__BUILD↔</code> —	version	P

Note

check with NVCC and KHAOS_NVCC_VERSION

Definition at line 1005 of file [Compiler.h](#).

2.5.3.77 COMPILER_IS_GREATER_EQUAL

```
#define COMPILER_IS_GREATER_EQUAL(  
    compiler,  
    major,  
    minor,  
    patch ) (KHAOS_COMPILER_##compiler >= KHAOS_SET_VERSION(major, minor, patch))
```

Definition at line 1006 of file [Compiler.h](#).

2.5.3.78 COMPILER_IS_LOWER

```
#define COMPILER_IS_LOWER(  
    compiler,  
    major,  
    minor,  
    patch ) (KHAOS_COMPILER_##compiler < KHAOS_SET_VERSION(major, minor, patch))
```

Definition at line 1007 of file [Compiler.h](#).

2.5.3.79 KHAOS_CLANG_EMULATED_VERSION

```
#define KHAOS_CLANG_EMULATED_VERSION KHAOS_SET_VERSION(0, 0, 0)
```

Definition at line 1410 of file [Compiler.h](#).

2.5.3.80 KHAOS_CLANG_EMULATED_VERSION_TWEAK

```
#define KHAOS_CLANG_EMULATED_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0, 0, 0)
```

Definition at line 1412 of file [Compiler.h](#).

2.5.3.81 KHAOS_COMPILER_ACC

```
#define KHAOS_COMPILER_ACC 0
```

Definition at line 1437 of file [Compiler.h](#).

2.5.3.82 KHAOS_COMPILER_Altium

```
#define KHAOS_COMPILER_Altium 0
```

Definition at line 1443 of file [Compiler.h](#).

2.5.3.83 KHAOS_COMPILER_AltiumMicroblaze

```
#define KHAOS_COMPILER_AltiumMicroblaze 0
```

Definition at line 1440 of file [Compiler.h](#).

2.5.3.84 KHAOS_COMPILER_Amsterdam

```
#define KHAOS_COMPILER_Amsterdam 0
```

Definition at line 1446 of file [Compiler.h](#).

2.5.3.85 KHAOS_COMPILER_AppleClang

```
#define KHAOS_COMPILER_AppleClang 0
```

Definition at line 1671 of file [Compiler.h](#).

2.5.3.86 KHAOS_COMPILER_ARM

```
#define KHAOS_COMPILER_ARM 0
```

Definition at line 1449 of file [Compiler.h](#).

2.5.3.87 KHAOS_COMPILER_Aztec

```
#define KHAOS_COMPILER_Aztec 0
```

Definition at line 1452 of file [Compiler.h](#).

2.5.3.88 KHAOS_COMPILER_Borland

```
#define KHAOS_COMPILER_Borland 0
```

Definition at line 1455 of file [Compiler.h](#).

2.5.3.89 KHAOS_COMPILER_CC65

```
#define KHAOS_COMPILER_CC65 0
```

Definition at line 1458 of file [Compiler.h](#).

2.5.3.90 KHAOS_COMPILER_Clang

```
#define KHAOS_COMPILER_Clang 0
```

Definition at line 1674 of file [Compiler.h](#).

2.5.3.91 KHAOS_COMPILER_ClangEmulated

```
#define KHAOS_COMPILER_ClangEmulated
```

Detect the emulated compilers.

Detected emulated compilers are :

Clang :

Note

check with ClangEmulated and KHAOS_CLANG_EMULATED_VERSION

LLVM :

Note

check with LLVMEmulated

GCC :

Note

check with GCCEmulated KHAOS_GCC_EMULATED_VERSION

Definition at line 1409 of file [Compiler.h](#).

2.5.3.92 KHAOS_COMPILER_Comeau

```
#define KHAOS_COMPILER_Comeau 0
```

Definition at line 1461 of file [Compiler.h](#).

2.5.3.93 KHAOS_COMPILER_Compacq

```
#define KHAOS_COMPILER_Compacq 0
```

Definition at line 1464 of file [Compiler.h](#).

2.5.3.94 KHAOS_COMPILER_Compcert

```
#define KHAOS_COMPILER_Compcert 0
```

Definition at line 1470 of file [Compiler.h](#).

2.5.3.95 KHAOS_COMPILER_Convex

```
#define KHAOS_COMPILER_Convex 0
```

Definition at line 1467 of file [Compiler.h](#).

2.5.3.96 KHAOS_COMPILER_Coverity

```
#define KHAOS_COMPILER_Coverity 0
```

Definition at line 1473 of file [Compiler.h](#).

2.5.3.97 KHAOS_COMPILER_CrayC

```
#define KHAOS_COMPILER_CrayC 0
```

Definition at line 1476 of file [Compiler.h](#).

2.5.3.98 KHAOS_COMPILER_Diab

```
#define KHAOS_COMPILER_Diab 0
```

Definition at line 1479 of file [Compiler.h](#).

2.5.3.99 KHAOS_COMPILER_DICE

```
#define KHAOS_COMPILER_DICE 0
```

Definition at line 1482 of file [Compiler.h](#).

2.5.3.100 KHAOS_COMPILER_DigitalMars

```
#define KHAOS_COMPILER_DigitalMars 0
```

Definition at line 1485 of file [Compiler.h](#).

2.5.3.101 KHAOS_COMPILER_Dignus

```
#define KHAOS_COMPILER_Dignus 0
```

Definition at line 1488 of file [Compiler.h](#).

2.5.3.102 KHAOS_COMPILER_DJGPP

```
#define KHAOS_COMPILER_DJGPP 0
```

Definition at line 1491 of file [Compiler.h](#).

2.5.3.103 KHAOS_COMPILER_EDG

```
#define KHAOS_COMPILER_EDG 0
```

Definition at line 1494 of file [Compiler.h](#).

2.5.3.104 KHAOS_COMPILER_EKOPath

```
#define KHAOS_COMPILER_EKOPath 0
```

Definition at line 1497 of file [Compiler.h](#).

2.5.3.105 KHAOS_COMPILER_Emscripten

```
#define KHAOS_COMPILER_Emscripten 0
```

Definition at line 1500 of file [Compiler.h](#).

2.5.3.106 KHAOS_COMPILER_Fujitsu

```
#define KHAOS_COMPILER_Fujitsu 0
```

Definition at line 1503 of file [Compiler.h](#).

2.5.3.107 KHAOS_COMPILER_GCC

```
#define KHAOS_COMPILER_GCC 0
```

Definition at line 1680 of file [Compiler.h](#).

2.5.3.108 KHAOS_COMPILER_GCCEmulated

```
#define KHAOS_COMPILER_GCCEmulated 0
```

Definition at line 1422 of file [Compiler.h](#).

2.5.3.109 KHAOS_COMPILER_GCCXML

```
#define KHAOS_COMPILER_GCCXML 0
```

Definition at line 1506 of file [Compiler.h](#).

2.5.3.110 KHAOS_COMPILER_GreenHills

```
#define KHAOS_COMPILER_GreenHills 0
```

Definition at line 1509 of file [Compiler.h](#).

2.5.3.111 KHAOS_COMPILER_HPCC

```
#define KHAOS_COMPILER_HPCC 0
```

Definition at line 1515 of file [Compiler.h](#).

2.5.3.112 KHAOS_COMPILER_HPCC

```
#define KHAOS_COMPILER_HPCC 0
```

Definition at line 1512 of file [Compiler.h](#).

2.5.3.113 KHAOS_COMPILER_IAR

```
#define KHAOS_COMPILER_IAR 0
```

Definition at line 1518 of file [Compiler.h](#).

2.5.3.114 KHAOS_COMPILER_IBM

```
#define KHAOS_COMPILER_IBM 0
```

Definition at line 1521 of file [Compiler.h](#).

2.5.3.115 KHAOS_COMPILER_IBMC370

```
#define KHAOS_COMPILER_IBMC370 0
```

Definition at line 1530 of file [Compiler.h](#).

2.5.3.116 KHAOS_COMPILER_IBMClang

```
#define KHAOS_COMPILER_IBMClang 0
```

Definition at line 1524 of file [Compiler.h](#).

2.5.3.117 KHAOS_COMPILER_IBMLegacy

```
#define KHAOS_COMPILER_IBMLegacy 0
```

Definition at line 1527 of file [Compiler.h](#).

2.5.3.118 KHAOS_COMPILER_IBMMVS

```
#define KHAOS_COMPILER_IBMMVS 0
```

Definition at line 1533 of file [Compiler.h](#).

2.5.3.119 KHAOS_COMPILER_IBMOS390

```
#define KHAOS_COMPILER_IBMOS390 0
```

Definition at line 1536 of file [Compiler.h](#).

2.5.3.120 KHAOS_COMPILER_IBMZOS

```
#define KHAOS_COMPILER_IBMZOS 0
```

Definition at line 1539 of file [Compiler.h](#).

2.5.3.121 KHAOS_COMPILER_ImageCraft

```
#define KHAOS_COMPILER_ImageCraft 0
```

Definition at line 1542 of file [Compiler.h](#).

2.5.3.122 KHAOS_COMPILER_Intel

```
#define KHAOS_COMPILER_Intel 0
```

Definition at line 1545 of file [Compiler.h](#).

2.5.3.123 KHAOS_COMPILER_Kai

```
#define KHAOS_COMPILER_Kai 0
```

Definition at line 1548 of file [Compiler.h](#).

2.5.3.124 KHAOS_COMPILER_KeilC166

```
#define KHAOS_COMPILER_KeilC166 0
```

Definition at line 1554 of file [Compiler.h](#).

2.5.3.125 KHAOS_COMPILER_KeilC51

```
#define KHAOS_COMPILER_KeilC51 0
```

Definition at line 1557 of file [Compiler.h](#).

2.5.3.126 KHAOS_COMPILER_KeilCarm

```
#define KHAOS_COMPILER_KeilCarm 0
```

Definition at line 1551 of file [Compiler.h](#).

2.5.3.127 KHAOS_COMPILER_LCC

```
#define KHAOS_COMPILER_LCC 0
```

Definition at line 1560 of file [Compiler.h](#).

2.5.3.128 KHAOS_COMPILER_LLVM

```
#define KHAOS_COMPILER_LLVM 0
```

Definition at line 1677 of file [Compiler.h](#).

2.5.3.129 KHAOS_COMPILER_LLVMEmulated

```
#define KHAOS_COMPILER_LLVMEmulated 0
```

Definition at line 1430 of file [Compiler.h](#).

2.5.3.130 KHAOS_COMPILER_Metaware

```
#define KHAOS_COMPILER_Metaware 0
```

Definition at line 1563 of file [Compiler.h](#).

2.5.3.131 KHAOS_COMPILER_Metrowerks

```
#define KHAOS_COMPILER_Metrowerks 0
```

Definition at line 1566 of file [Compiler.h](#).

2.5.3.132 KHAOS_COMPILER_Microtec

```
#define KHAOS_COMPILER_Microtec 0
```

Definition at line 1572 of file [Compiler.h](#).

2.5.3.133 KHAOS_COMPILER_Microway

```
#define KHAOS_COMPILER_Microway 0
```

Definition at line 1575 of file [Compiler.h](#).

2.5.3.134 KHAOS_COMPILER_MIPSpro

```
#define KHAOS_COMPILER_MIPSpro 0
```

Definition at line 1578 of file [Compiler.h](#).

2.5.3.135 KHAOS_COMPILER_Miracle

```
#define KHAOS_COMPILER_Miracle 0
```

Definition at line 1581 of file [Compiler.h](#).

2.5.3.136 KHAOS_COMPILER_MPW

```
#define KHAOS_COMPILER_MPW 0
```

Definition at line 1584 of file [Compiler.h](#).

2.5.3.137 KHAOS_COMPILER_MSVC

```
#define KHAOS_COMPILER_MSVC 0
```

Definition at line 1569 of file [Compiler.h](#).

2.5.3.138 KHAOS_COMPILER_Norcroft

```
#define KHAOS_COMPILER_Norcroft 0
```

Definition at line 1587 of file [Compiler.h](#).

2.5.3.139 KHAOS_COMPILER_NVCC

```
#define KHAOS_COMPILER_NVCC 0
```

Definition at line 1016 of file [Compiler.h](#).

2.5.3.140 KHAOS_COMPILER_NWCC

```
#define KHAOS_COMPILER_NWCC 0
```

Definition at line 1590 of file [Compiler.h](#).

2.5.3.141 KHAOS_COMPILER_Open64

```
#define KHAOS_COMPILER_Open64 0
```

Definition at line 1593 of file [Compiler.h](#).

2.5.3.142 KHAOS_COMPILER_Oracle

```
#define KHAOS_COMPILER_Oracle 0
```

Definition at line 1596 of file [Compiler.h](#).

2.5.3.143 KHAOS_COMPILER_Pacific

```
#define KHAOS_COMPILER_Pacific 0
```

Definition at line 1602 of file [Compiler.h](#).

2.5.3.144 KHAOS_COMPILER_Palm

```
#define KHAOS_COMPILER_Palm 0
```

Definition at line 1605 of file [Compiler.h](#).

2.5.3.145 KHAOS_COMPILER_Pelles

```
#define KHAOS_COMPILER_Pelles 0
```

Definition at line 1608 of file [Compiler.h](#).

2.5.3.146 KHAOS_COMPILER_Portland

```
#define KHAOS_COMPILER_Portland 0
```

Definition at line 1611 of file [Compiler.h](#).

2.5.3.147 KHAOS_COMPILER_Renesas

```
#define KHAOS_COMPILER_Renesas 0
```

Definition at line 1614 of file [Compiler.h](#).

2.5.3.148 KHAOS_COMPILER_SASC

```
#define KHAOS_COMPILER_SASC 0
```

Definition at line 1617 of file [Compiler.h](#).

2.5.3.149 KHAOS_COMPILER_SCO

```
#define KHAOS_COMPILER_SCO 0
```

Definition at line 1620 of file [Compiler.h](#).

2.5.3.150 KHAOS_COMPILER_SDCC

```
#define KHAOS_COMPILER_SDCC 0
```

Definition at line 1623 of file [Compiler.h](#).

2.5.3.151 KHAOS_COMPILER_SNC

```
#define KHAOS_COMPILER_SNC 0
```

Definition at line 1626 of file [Compiler.h](#).

2.5.3.152 KHAOS_COMPILER_Solaris

```
#define KHAOS_COMPILER_Solaris 0
```

Definition at line 1599 of file [Compiler.h](#).

2.5.3.153 KHAOS_COMPILER_Stratus

```
#define KHAOS_COMPILER_Stratus 0
```

Definition at line 1629 of file [Compiler.h](#).

2.5.3.154 KHAOS_COMPILER_StratusANSI

```
#define KHAOS_COMPILER_StratusANSI 0
```

Definition at line 1635 of file [Compiler.h](#).

2.5.3.155 KHAOS_COMPILER_StratusKR

```
#define KHAOS_COMPILER_StratusKR 0
```

Definition at line 1632 of file [Compiler.h](#).

2.5.3.156 KHAOS_COMPILER_Symantec

```
#define KHAOS_COMPILER_Symantec 0
```

Definition at line 1638 of file [Compiler.h](#).

2.5.3.157 KHAOS_COMPILER_Tendra

```
#define KHAOS_COMPILER_Tendra 0
```

Definition at line 1641 of file [Compiler.h](#).

2.5.3.158 KHAOS_COMPILER_Think

```
#define KHAOS_COMPILER_Think 0
```

Definition at line 1647 of file [Compiler.h](#).

2.5.3.159 KHAOS_COMPILER_TI

```
#define KHAOS_COMPILER_TI 0
```

Definition at line 1644 of file [Compiler.h](#).

2.5.3.160 KHAOS_COMPILER_Tiny

```
#define KHAOS_COMPILER_Tiny 0
```

Definition at line 1650 of file [Compiler.h](#).

2.5.3.161 KHAOS_COMPILER_Turbo

```
#define KHAOS_COMPILER_Turbo 0
```

Definition at line 1653 of file [Compiler.h](#).

2.5.3.162 KHAOS_COMPILER_Ultimate

```
#define KHAOS_COMPILER_Ultimate 0
```

Definition at line 1656 of file [Compiler.h](#).

2.5.3.163 KHAOS_COMPILER_Unknown

```
#define KHAOS_COMPILER_Unknown 1
```

Definition at line 1372 of file [Compiler.h](#).

2.5.3.164 KHAOS_COMPILER_USLC

```
#define KHAOS_COMPILER_USLC 0
```

Definition at line 1659 of file [Compiler.h](#).

2.5.3.165 KHAOS_COMPILER_VBCC

```
#define KHAOS_COMPILER_VBCC 0
```

Definition at line 1662 of file [Compiler.h](#).

2.5.3.166 KHAOS_COMPILER_VERSION

```
#define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(0, 0, 0)
```

Definition at line 1376 of file [Compiler.h](#).

2.5.3.167 KHAOS_COMPILER_VERSION_TWEAK

```
#define KHAOS_COMPILER_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)
```

Definition at line 1380 of file [Compiler.h](#).

2.5.3.168 KHAOS_COMPILER_Watcom

```
#define KHAOS_COMPILER_Watcom 0
```

Definition at line 1665 of file [Compiler.h](#).

2.5.3.169 KHAOS_COMPILER_Zortech

```
#define KHAOS_COMPILER_Zortech 0
```

Definition at line 1668 of file [Compiler.h](#).

2.5.3.170 KHAOS_GCC_EMULATED_VERSION

```
#define KHAOS_GCC_EMULATED_VERSION KHAOS_SET_VERSION(0, 0, 0)
```

Definition at line 1423 of file [Compiler.h](#).

2.5.3.171 KHAOS_GCC_EMULATED_VERSION_TWEAK

```
#define KHAOS_GCC_EMULATED_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)
```

Definition at line 1425 of file [Compiler.h](#).

2.5.3.172 KHAOS_LLVM_EMULATED_VERSION

```
#define KHAOS_LLVM_EMULATED_VERSION KHAOS_SET_VERSION(0, 0, 0)
```

Definition at line 1432 of file [Compiler.h](#).

2.5.3.173 KHAOS_LLVM_EMULATED_VERSION_TWEAK

```
#define KHAOS_LLVM_EMULATED_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)
```

Definition at line 1433 of file [Compiler.h](#).

2.5.3.174 KHAOS_NVCC_VERSION

```
#define KHAOS_NVCC_VERSION KHAOS_SET_VERSION(0, 0, 0)
```

Definition at line 1017 of file [Compiler.h](#).

2.5.3.175 KHAOS_NVCC_VERSION_TWEAK

```
#define KHAOS_NVCC_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0, 0, 0)
```

Definition at line 1018 of file [Compiler.h](#).

2.6 Compiler.h

[Go to the documentation of this file.](#)

```
00001 #ifndef KHAOS_COMPILER_MACROS_H
00002 #define KHAOS_COMPILER_MACROS_H
00003
00004 #include "khaos/VersionMacros.h"
00005
01005 #define COMPILER_IS(compiler) (KHAOS_COMPILER_##compiler > 0)
01006 #define COMPILER_IS_GREATER_EQUAL(compiler, major, minor, patch) (KHAOS_COMPILER_##compiler >=
    KHAOS_SET_VERSION(major, minor, patch))
01007 #define COMPILER_IS_LOWER(compiler, major, minor, patch) (KHAOS_COMPILER_##compiler <
    KHAOS_SET_VERSION(major, minor, patch))
01008
01009 #if defined(__NVCC__)
01010 // /warn https://github.com/boostorg/predef/blob/develop/include/boost/predef/compiler/nvcc.h
01011 #if defined(__CUDACC_VER_MAJOR__) && defined(__CUDACC_VER_MINOR__) && defined(__CUDACC_VER_BUILD__)
01012 #define KHAOS_NVCC_VERSION KHAOS_SET_VERSION(__CUDACC_VER_MAJOR__, __CUDACC_VER_MINOR__,
    __CUDACC_VER_BUILD__)
01013 #endif
01014 #define KHAOS_COMPILER_NVCC KHAOS_NVCC_VERSION
01015 #else
01016 #define KHAOS_COMPILER_NVCC 0
01017 #define KHAOS_NVCC_VERSION KHAOS_SET_VERSION(0, 0, 0)
01018 #define KHAOS_NVCC_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0, 0, 0)
01019 #endif
01020
01021 // Put all compiler first then MSVC clang llvm gcc as some mimic others.
01022
01023 #if defined(_ACC_)
01024 #define KHAOS_COMPILER_ACC 1
01025 #elif defined(__CMB__)
01026 #if defined(__BUILD__)
01027 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__BUILD__ / 1000000) % 10, ((__BUILD__) / 1000)
    % 1000, __BUILD__ % 1000)
01028 #else
01029 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__VERSION__ / 1000) % 10, __VERSION__ % 1000,
    __REVISION__)
01030 #endif
01031 #define KHAOS_COMPILER_AltiumMicroblaze KHAOS_COMPILER_VERSION
01032 #elif defined(__CHC__)
01033 #if defined(__BUILD__)
```

```

01034     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__BUILD__ / 1000000) % 10, ((__BUILD__) / 1000)
% 1000, __BUILD__ % 1000)
01035     #else
01036     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__VERSION__ / 1000) % 10, __VERSION__ % 1000,
__REVISION__)
01037     #endif
01038     #define KHAOS_COMPILER_Altium KHAOS_COMPILER_VERSION
01039 #elif defined(__ACK__)
01040     #define KHAOS_COMPILER_Amsterdam 1
01041 #elif defined(__CC_ARM)
01042     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__ARMCC_VERSION / 100000) % 10,
((__ARMCC_VERSION) / 10000) % 10, (__ARMCC_VERSION / 1000) % 10)
01043     #define KHAOS_COMPILER_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(__ARMCC_VERSION % 1000)
01044     #define KHAOS_COMPILER_ARM KHAOS_COMPILER_VERSION
01045 #elif defined(AZTEC_C) || defined(__AZTEC_C__)
01046     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(((__VERSION__) / 100) % 10, (__VERSION__) % 100, 0)
01047     #define KHAOS_COMPILER_Aztec KHAOS_COMPILER_VERSION
01048 #elif defined(__BORLANDC__) || defined(__CODEGEARC__)
01049     #if defined(__CODEGEARC__)
01050     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__CODEGEARC__ & 0xFF00) >> 8, (__CODEGEARC__ &
0xF0) >> 4, (__CODEGEARC__ & 0xF))
01051     #elif defined(__BORLANDC__)
01052     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__BORLANDC__ & 0xFF00) >> 8, (__BORLANDC__ &
0xF0) >> 4, (__BORLANDC__ & 0xF))
01053     #endif
01054     #define KHAOS_COMPILER_Borland KHAOS_COMPILER_VERSION
01055 #elif defined(__CC65__)
01056     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__CC65__ & 0xF00) >> 8, (__CC65__ & 0xF0) >> 4,
(__CC65__ & 0xF))
01057     #define KHAOS_COMPILER_CC65 KHAOS_COMPILER_VERSION
01058 #elif defined(__COMO__)
01059     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__COMO_VERSION__ & 0xF00) >> 8, (__COMO_VERSION__ &
0xF0) >> 4, (__COMO_VERSION__ & 0xF))
01060     #define KHAOS_COMPILER_Comeau KHAOS_COMPILER_VERSION
01061 #elif defined(__DECC) || defined(__DECCXX) || defined(__VAXC) || defined(VAXC)
01062     #if defined(__DECCXX_VER)
01063     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(((__DECCXX_VER) / 1000000) % 100,
((__DECCXX_VER) / 100000) % 100, ((__DECCXX_VER) / 100) % 100)
01064     #elif defined(__DECC_VER)
01065     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(((__DECC_VER) / 1000000) % 100, ((__DECC_VER) /
100000) % 100, ((__DECC_VER) / 100) % 100)
01066     #endif
01067     #define KHAOS_COMPILER_Compaq KHAOS_COMPILER_VERSION
01068 #elif defined(__convexc__)
01069     #define KHAOS_COMPILER_Convex 1
01070 #elif defined(__COMP CERT__)
01071     #define KHAOS_COMPILER_Compcert 1
01072 #elif defined(__COVERITY__)
01073     #define KHAOS_COMPILER_Coverity 1
01074 #elif defined(__CRAYC__)
01075     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(_RELEASE, _RELEASE_MINOR, 0)
01076     #define KHAOS_COMPILER_CrayC KHAOS_COMPILER_VERSION
01077 #elif defined(__DCC__)
01078     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(((__VERSION_NUMBER__) / 1000) % 10,
((__VERSION_NUMBER__) / 100) % 10, (__VERSION_NUMBER__) % 100)
01079     #define KHAOS_COMPILER_Diab KHAOS_COMPILER_VERSION
01080 #elif defined(__DICE__)
01081     #define KHAOS_COMPILER_DICE 1
01082 #elif defined(__DMC__)
01083     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__DMC__ & 0xF00) >> 8, (__DMC__ & 0xF0) >> 4,
(__DMC__ & 0xF))
01084     #define KHAOS_COMPILER_DigitalMars KHAOS_COMPILER_VERSION
01085 #elif defined(__SYSC__)
01086     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(((__SYSC_VER__) / 10000) % 10, ((__SYSC_VER__) /
100) % 100, (__SYSC_VER__) % 100)
01087     #define KHAOS_COMPILER_Dignus KHAOS_COMPILER_VERSION
01088 #elif defined(__DJGPP__) || defined(__GO32__)
01089     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(__DJGPP__, __DJGPP_MINOR__, 0)
01090     #define KHAOS_COMPILER_DJGPP KHAOS_COMPILER_VERSION
01091 #elif defined(__EDG__)
01092     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__EDG_VERSION__) / 100000) % 10,
((__EDG_VERSION__) / 1000) % 100, 0)
01093     #define KHAOS_COMPILER_EDG KHAOS_COMPILER_VERSION
01094 #elif defined(__PATHCC__)
01095     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(__PATHCC__, __PATHCC_MINOR__,
__PATHCC_PATCHLEVEL__)
01096     #define KHAOS_COMPILER_EKOPath KHAOS_COMPILER_VERSION
01097 #elif defined(__EMSCRIPTEN__)
01098     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(__EMSCRIPTEN_major__, __EMSCRIPTEN_minor__,
__EMSCRIPTEN_tiny__)
01099     #define KHAOS_COMPILER_Emscripten KHAOS_COMPILER_VERSION
01100 #elif defined(__FCC_VERSION__)
01101     #define KHAOS_COMPILER_Fujitsu 1
01102 #elif defined(__GCCXML__)
01103     #define KHAOS_COMPILER_GCCXML 1
01104 #elif defined(__ghs__) || defined(__ghs__)
01105     #if defined(__GHS_VERSION_NUMBER__)

```



```

01106     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(((GHS_VERSION_NUMBER) / 100) % 10,
01107     ((GHS_VERSION_NUMBER) / 10) % 10, (GHS_VERSION_NUMBER) % 10)
01108     #elif defined(__ghs)
01109     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(((ghs) / 100) % 10, ((ghs) / 10) % 10,
01110     (ghs) % 10)
01111     #endif
01112     #define KHAOS_COMPILER_GreenHills KHAOS_COMPILER_VERSION
01113     #elif defined(__HP_cc)
01114     #define KHAOS_COMPILER_HPCC 1
01115     #elif defined(__HP_acc)
01116     #if __HP_acc > 1
01117     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(((HP_acc) / 10000) % 100, ((HP_acc) / 100) %
01118     100, (HP_acc) % 100)
01119     #define KHAOS_COMPILER_HPacc KHAOS_COMPILER_VERSION
01120     #else
01121     #define KHAOS_COMPILER_HPacc 1
01122     #endif
01123     #elif defined(__IAR_SYSTEMS_ICC__)
01124     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(((VER) / 100) % 100, (VER) % 100, 0)
01125     #define KHAOS_COMPILER_IAR KHAOS_COMPILER_VERSION
01126     #elif defined(__ibmxl) && defined(__clang__)
01127     //Should be include before clang as it defined some macro of Clang
01128     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(__ibmxl_version__, __ibmxl_release__,
01129     __ibmxl_modification__)
01130     #define KHAOS_COMPILER_IBM KHAOS_COMPILER_VERSION
01131     #define KHAOS_COMPILER_IBMclang KHAOS_COMPILER_VERSION
01132     #define KHAOS_COMPILER_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(__ibmxl_ptf_fix_level__)
01133     #elif defined(__xlc__)
01134     #if defined(__IBMC__)
01135     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__IBMC__ & 0xFF0000) >> 16, (__IBMC__ & 0xFF00) >>
01136     8, (__IBMC__ & 0xFF))
01137     #elif defined(__IBMCPP__)
01138     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__IBMCPP__ & 0xFF0000) >> 16, (__IBMCPP__ &
01139     0xFF00) >> 8, (__IBMCPP__ & 0xFF))
01140     #endif
01141     #if defined(__xlc__)
01142     #define KHAOS_COMPILER_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK((__xlc__ & 0xFF) % 100)
01143     #elif defined(__xlc_ver__)
01144     #define KHAOS_COMPILER_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK((__xlc_ver__ & 0xFF) % 100)
01145     #endif
01146     #define KHAOS_COMPILER_IBM KHAOS_COMPILER_VERSION
01147     #define KHAOS_COMPILER_IBMLegacy KHAOS_COMPILER_VERSION
01148     #elif defined(__IBMC__) || defined(__IBMCPP__)
01149     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__COMPILER_VER__ & 0xF000000) >> 24,
01150     (__COMPILER_VER__ & 0xFF0000) >> 16, (__COMPILER_VER__ & 0xFF00) >> 8)
01151     #define KHAOS_COMPILER_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK((__COMPILER_VER__ & 0xFF))
01152     #define KHAOS_COMPILER_IBM KHAOS_COMPILER_VERSION
01153     #if __COMPILER_VER__ & 0xF0000000 >> 28 == 0
01154     #define KHAOS_COMPILER_IBM370 KHAOS_COMPILER_VERSION
01155     #elif __COMPILER_VER__ & 0xF0000000 >> 28 == 1
01156     #define KHAOS_COMPILER_IBM370 KHAOS_COMPILER_VERSION
01157     #define KHAOS_COMPILER_IBM370 KHAOS_COMPILER_VERSION
01158     #elif __COMPILER_VER__ & 0xF0000000 >> 28 == 2
01159     #define KHAOS_COMPILER_IBM370 KHAOS_COMPILER_VERSION
01160     #elif __COMPILER_VER__ & 0xF0000000 >> 28 == 3
01161     #define KHAOS_COMPILER_IBM370 KHAOS_COMPILER_VERSION
01162     #elif __COMPILER_VER__ & 0xF0000000 >> 28 == 4
01163     #define KHAOS_COMPILER_IBM370 KHAOS_COMPILER_VERSION
01164     #define KHAOS_COMPILER_IBM370 KHAOS_COMPILER_VERSION
01165     #endif
01166     #elif defined(__IMAGECRAFT__)
01167     #define KHAOS_COMPILER_ImageCraft 1
01168     #elif defined(__INTEL_COMPILER) || defined(__ICL) || defined(__ICC) || defined(__ECC)
01169     #if defined(__INTEL_COMPILER) && (__INTEL_COMPILER == 9999)
01170     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(12, 1, 0)
01171     #elif defined(__INTEL_COMPILER) && defined(__INTEL_COMPILER_UPDATE)
01172     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(((INTEL_COMPILER) / 100) % 100,
01173     (INTEL_COMPILER) % 100, __INTEL_COMPILER_UPDATE)
01174     #elif defined(__INTEL_COMPILER)
01175     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(((INTEL_COMPILER) / 100) % 100,
01176     (INTEL_COMPILER) % 100, 0)
01177     #endif
01178     #define KHAOS_COMPILER_Intel KHAOS_COMPILER_VERSION
01179     #elif defined(__KCC)
01180     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__KCC_VERSION & 0xF000) >> 12, (__KCC_VERSION &
01181     0xF00) >> 8, (__KCC_VERSION & 0xFF))
01182     #define KHAOS_COMPILER_Kai KHAOS_COMPILER_VERSION
01183     #elif defined(__CA__) || defined(__KEIL__)
01184     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(((CA__) / 100) % 10, (CA__) % 100, 0)
01185     #define KHAOS_COMPILER_KeilCarm KHAOS_COMPILER_VERSION
01186     #elif defined(__C166__)
01187     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(((C166__) / 100) % 10, (C166__) % 100, 0)
01188     #define KHAOS_COMPILER_KeilC166 KHAOS_COMPILER_VERSION
01189     #elif defined(__C51__) || defined(__CX51__)
01190     #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(((C51__) / 100) % 10, (C51__) % 100, 0)
01191     #define KHAOS_COMPILER_KeilC51 KHAOS_COMPILER_VERSION
01192     #elif defined(__LCC__)
01193     #define KHAOS_COMPILER_LCC 1
01194     #elif defined(__HIGHC__)
01195     #define KHAOS_COMPILER_Metaware 1
01196     #elif defined(__MWERKS__) || defined(__CWCC__)

```

```

01186 #if defined(__CWCC__)
01187 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__CWCC__ & 0xF000) >> 12, (__CWCC__ & 0xF00) >> 8,
01188 (__CWCC__ & 0xFF))
01189 #elif(__MWERKS__ >= 0x4200)
01189 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__MWERKS__ & 0xF000) >> 12, (__MWERKS__ & 0xF00)
01190 >> 8, (__MWERKS__ & 0xFF))
01190 #elif(__MWERKS__ >= 0x3204) // note the "skip": 04->9.3
01191 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(9, (__MWERKS__) % 100 - 1, 0)
01192 #elif(__MWERKS__ >= 0x3200)
01193 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(9, (__MWERKS__) % 100, 0)
01194 #elif(__MWERKS__ >= 0x3000)
01195 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(8, (__MWERKS__) % 100, 0)
01196 #endif
01197 #define KHAOS_COMPILER_Metrowerks KHAOS_COMPILER_VERSION
01198 #elif defined(_MSC_VER)
01199 #if defined(_MSC_BUILD)
01200 #define KHAOS_COMPILER_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(_MSC_BUILD)
01201 #endif
01202
01203 // how many digits does the build number have?
01204 #if defined(_MSC_FULL_VER)
01205 #if _MSC_FULL_VER / 10000 == _MSC_VER
01206 // four digits
01207 #define COMP_MSVC_BUILD_PATCH (_MSC_FULL_VER % 10000)
01208 #elif _MSC_FULL_VER / 100000 == _MSC_VER
01209 // five digits
01210 #define COMP_MSVC_BUILD_PATCH (_MSC_FULL_VER % 100000)
01211 #endif
01212 #else
01213 #define COMP_MSVC_BUILD_PATCH 0
01214 #endif
01215 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((_MSC_VER / 100) % 1000, (_MSC_VER) % 1000,
01216 COMP_MSVC_BUILD_PATCH)
01217 #define KHAOS_COMPILER_MSVC KHAOS_COMPILER_VERSION
01218 #elif defined(_MRI)
01218 #define KHAOS_COMPILER_Microtec 1
01219 #elif defined(__NDPC__) || defined(__NDPX__)
01220 #define KHAOS_COMPILER_Microway 1
01221 #elif defined(__sgi) || defined(sgi)
01222 #if defined(_SGI_COMPILER_VERSION)
01223 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((_SGI_COMPILER_VERSION) / 100) % 10,
01224 ((_SGI_COMPILER_VERSION) / 10) % 10, (_SGI_COMPILER_VERSION) % 10)
01224 #elif defined(_COMPILER_VERSION)
01225 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((_COMPILER_VERSION) / 100) % 10,
01226 ((_COMPILER_VERSION) / 10) % 10, (_COMPILER_VERSION) % 10)
01226 #endif
01227 #define KHAOS_COMPILER_MIPSPRO KHAOS_COMPILER_VERSION
01228 #elif defined(MIRACLE)
01229 #define KHAOS_COMPILER_Miracle 1
01230 #elif defined(__MRC__) || defined(MPW_C) || defined(MPW_CPLUS)
01231 #if defined(__MRC__)
01232 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__MRC__ & 0xFF00) >> 8, (__MRC__ & 0xFF), 0)
01233 #define KHAOS_COMPILER_MPW KHAOS_COMPILER_VERSION
01234 #else
01235 #define KHAOS_COMPILER_MPW 1
01236 #endif
01237 #elif defined(__CC_NORCROFT)
01238 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(((__ARMCC_VERSION * 100) / 100), (__ARMCC_VERSION
01239 * 100) % 100, 0)
01239 #define KHAOS_COMPILER_Norcroft KHAOS_COMPILER_VERSION
01240 #elif defined(__NWCC__)
01241 #define KHAOS_COMPILER_NWCC 1
01242 #elif defined(__OPEN64__) || defined(__OPENCC__)
01243 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(__OPENCC__, __OPENCC_MINOR__,
01244 (__OPENCC_PATCHLEVEL__ * 10) / 10)
01244 #define KHAOS_COMPILER_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK((__OPENCC_PATCHLEVEL__ * 10) % 10)
01245 #define KHAOS_COMPILER_Open64 KHAOS_COMPILER_VERSION
01246 #elif defined(ORA_PROC)
01247 #define KHAOS_COMPILER_Oracle 1
01248 #elif defined(__SUNPRO_CC) || defined(__SUNPRO_C)
01249 #if defined(__SUNPRO_CC)
01250 #if (__SUNPRO_CC < 0x5100)
01251 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__SUNPRO_CC & 0xF00) >> 8, (__SUNPRO_CC & 0xF0)
01252 >> 4, (__SUNPRO_CC & 0xF))
01252 #else
01253 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__SUNPRO_CC & 0xFF000) >> 12, (__SUNPRO_CC &
01254 0xFF0) >> 4, (__SUNPRO_CC & 0xF))
01254 #endif
01255 #elif defined(__SUNPRO_C)
01256 #if (__SUNPRO_C < 0x5100)
01257 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__SUNPRO_C & 0xF00) >> 8, (__SUNPRO_C & 0xF0) >>
01258 4, (__SUNPRO_C & 0xF))
01258 #else
01259 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__SUNPRO_C & 0xFF000) >> 12, (__SUNPRO_C &
01260 0xFF0) >> 4, (__SUNPRO_C & 0xF))
01260 #endif
01261 #endif

```

```

01262 #define KHAOS_COMPILER_Solaris KHAOS_COMPILER_VERSION
01263 #elif defined(__PACIFIC__)
01264 #define KHAOS_COMPILER_Pacific 1
01265 #elif defined(__PACC_VER)
01266 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__PACC_VER & 0xF0000000) >> 28, (__PACC_VER &
01267 0xFF000000) >> 20, (__PACC_VER & 0xFF0000) >> 12)
01267 #define KHAOS_COMPILER_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK((__PACC_VER & 0xFF))
01268 #define KHAOS_COMPILER_Palm KHAOS_COMPILER_VERSION
01269 #elif defined(__POCC__)
01270 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(((__POCC__) / 100) % 10, (__POCC__) % 100, 0)
01271 #define KHAOS_COMPILER_Pelles KHAOS_COMPILER_VERSION
01272 #elif defined(__PGI)
01273 #if defined(__PGIC__) && defined(__PGIC_MINOR__) && defined(__PGIC_PATCHLEVEL__)
01274 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(__PGIC__, __PGIC_MINOR__, __PGIC_PATCHLEVEL__)
01275 #define KHAOS_COMPILER_Portland KHAOS_COMPILER_VERSION
01276 #else
01277 #define KHAOS_COMPILER_Portland 1
01278 #endif
01279 #elif defined(__RENESAS__) || defined(__HITACHI__)
01280 #if defined(__RENESAS_VERSION__)
01281 #if __RENESAS_VERSION__ >= 0x1000000
01282 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__RENESAS_VERSION__ & 0xFF000000) >> 24,
01283 (__RENESAS_VERSION__ & 0xFF0000) >> 16, (__RENESAS_VERSION__ & 0xFF00) >> 8)
01283 #else
01284 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__RENESAS_VERSION__ & 0xFF00) >> 8,
01285 (__RENESAS_VERSION__ & 0xFF), 0)
01285 #endif
01286 #elif defined(__HITACHI_VERSION__)
01287 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__HITACHI_VERSION__ & 0xFF00) >> 8,
01288 (__HITACHI_VERSION__ & 0xFF), 0)
01288 #endif
01289 #define KHAOS_COMPILER_Renesas KHAOS_COMPILER_VERSION
01290 #elif defined(__SASC__) || defined(__SASC__) || defined(__SASC__)
01291 #if defined(__SASC__)
01292 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__SASC__) / 100) % 10, (__SASC__) % 100, 0)
01293 #elif defined(__VERSION__)
01294 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(__VERSION__, __REVISION__, 0)
01295 #endif
01296 #define KHAOS_COMPILER_SASC KHAOS_COMPILER_VERSION
01297 #elif defined(__SCO_DS)
01298 #define KHAOS_COMPILER_SCO 1
01299 #elif defined(__SDCC)
01300 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(((SDCC) / 100) % 10, ((SDCC) / 10) % 10, (SDCC) %
01301 10)
01301 #define KHAOS_COMPILER_SDCC KHAOS_COMPILER_VERSION
01302 #elif defined(__SNC__)
01303 #define KHAOS_COMPILER_SNC 1
01304 #elif defined(__VOSC__)
01305 #define KHAOS_COMPILER_Stratus 1
01306 #if __VOSC__ == 0
01307 #define KHAOS_COMPILER_StratusKR 1
01308 #elif __VOSC__ == 1
01309 #define KHAOS_COMPILER_StratusANSI 1
01310 #endif
01311 #elif defined(__SC__)
01312 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__SC__ & 0xFF00) >> 8, (__SC__ & 0xFF), 0)
01313 #define KHAOS_COMPILER_Symantec KHAOS_COMPILER_VERSION
01314 #elif defined(__TenDRA__)
01315 #define KHAOS_COMPILER_Tendra 1
01316 #elif defined(__TI_COMPILER_VERSION__) || defined(__TMS320C6X)
01317 #if defined(__TI_COMPILER_VERSION__)
01318 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__TI_COMPILER_VERSION__ / 1000000) % 1000,
01319 (__TI_COMPILER_VERSION__ / 1000) % 1000, __TI_COMPILER_VERSION__ % 1000)
01319 #define KHAOS_COMPILER_TI KHAOS_COMPILER_VERSION
01320 #else
01321 #define KHAOS_COMPILER_TI 1
01322 #endif
01323 #elif defined(THINKC3) || defined(THINKC4)
01324 #if defined(THINKC3)
01325 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(3, 0, 0)
01326 #elif defined(THINKC4)
01327 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(4, 0, 0)
01328 #endif
01329 #define KHAOS_COMPILER_Think KHAOS_COMPILER_VERSION
01330 #elif defined(__TINYC__)
01331 #define KHAOS_COMPILER_Tiny 1
01332 #elif defined(__TURBOC__)
01333 #if __TURBOC__ < 0x295 || __TURBOC__ > 0x400
01334 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__TURBOC__ & 0xFF00) >> 8, (__TURBOC__ & 0xFF),
01335 0)
01335 #define KHAOS_COMPILER_Turbo KHAOS_COMPILER_VERSION
01336 #else
01337 #define KHAOS_COMPILER_Turbo 1
01338 #endif
01339 #elif defined(__UCC)
01340 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(__MAJOR_REV__, __MINOR_REV__, 0)
01341 #define KHAOS_COMPILER_Ultimate KHAOS_COMPILER_VERSION

```

```

01342 #elif defined(__USLC__)
01343 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__SCO_VERSION__ / 1000000), (__SCO_VERSION__ /
100000) % 100)
01344 #define KHAOS_COMPILER_USLC KHAOS_COMPILER_VERSION
01345 #elif defined(__VBCC__)
01346 #define KHAOS_COMPILER_VBCC 1
01347 #elif defined(__WATCOMC__)
01348 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(((__WATCOMC__) / 100) % 100, (__WATCOMC__) % 100,
0)
01349 #define KHAOS_COMPILER_Watcom KHAOS_COMPILER_VERSION
01350 #elif defined(__ZTC__)
01351 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION((__ZTC__ & 0xF00) >> 8, (__ZTC__ & 0xF0) >> 4,
(__ZTC__ & 0xF))
01352 #define KHAOS_COMPILER_Zortech KHAOS_COMPILER_VERSION
01353 #elif defined(__clang__)
01354 //Should be include before gcc as it defined some macro of GCC and MSVC
01355 //Note that marketing version numbers should not be used to check for language features, as
different vendors use different numbering schemes. Instead, use the Feature Checking Macros.
01356 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(__clang_major__, __clang_minor__,
__clang_patchlevel__)
01357 #if defined(__apple_build_version__)
01358 #define KHAOS_COMPILER_AppleClang KHAOS_COMPILER_VERSION
01359 #endif
01360 #define KHAOS_COMPILER_Clang KHAOS_COMPILER_VERSION
01361 #elif defined(__llvm__)
01362 //Should be after clang because clang mimic llvm
01363 #define KHAOS_COMPILER_LLVM 1
01364 #elif defined(__GNUC__) || defined(__GNUG__)
01365 #if defined(__GNUC_PATCHLEVEL__)
01366 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(__GNUC__, __GNUC_MINOR__, __GNUC_PATCHLEVEL__)
01367 #else
01368 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(__GNUC__, __GNUC_MINOR__, 0)
01369 #endif
01370 #define KHAOS_COMPILER_GCC KHAOS_COMPILER_VERSION
01371 #else
01372 #define KHAOS_COMPILER_Unknown 1
01373 #endif
01374
01375 #if !defined(KHAOS_COMPILER_VERSION)
01376 #define KHAOS_COMPILER_VERSION KHAOS_SET_VERSION(0, 0, 0)
01377 #endif
01378
01379 #if !defined(KHAOS_COMPILER_VERSION_TWEAK)
01380 #define KHAOS_COMPILER_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)
01381 #endif
01382
01401 #if defined(__clang__)
01402 #if defined(__clang_major__)
01403 #define KHAOS_CLANG_EMULATED_VERSION KHAOS_SET_VERSION(__clang_major__, __clang_minor__,
__clang_patchlevel__)
01404 #define KHAOS_COMPILER_ClangEmulated KHAOS_CLANG_EMULATED_VERSION
01405 #else
01406 #define KHAOS_COMPILER_ClangEmulated 1
01407 #endif
01408 #else
01409 #define KHAOS_COMPILER_ClangEmulated 0
01410 #define KHAOS_CLANG_EMULATED_VERSION KHAOS_SET_VERSION(0, 0, 0)
01411 #endif
01412 #define KHAOS_CLANG_EMULATED_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0, 0, 0)
01413
01414 #if defined(__GNUC__) || defined(__GNUG__)
01415 #if defined(__GNUC_PATCHLEVEL__)
01416 #define KHAOS_GCC_EMULATED_VERSION KHAOS_SET_VERSION(__GNUC__, __GNUC_MINOR__,
__GNUC_PATCHLEVEL__)
01417 #else
01418 #define KHAOS_GCC_EMULATED_VERSION KHAOS_SET_VERSION(__GNUC__, __GNUC_MINOR__, 0)
01419 #endif
01420 #define KHAOS_COMPILER_GCCEmulated KHAOS_GCC_EMULATED_VERSION
01421 #else
01422 #define KHAOS_COMPILER_GCCEmulated 0
01423 #define KHAOS_GCC_EMULATED_VERSION KHAOS_SET_VERSION(0, 0, 0)
01424 #endif
01425 #define KHAOS_GCC_EMULATED_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)
01426
01427 #if defined(__llvm__)
01428 #define KHAOS_COMPILER_LLVMEmulated 1
01429 #else
01430 #define KHAOS_COMPILER_LLVMEmulated 0
01431 #endif
01432 #define KHAOS_LLVM_EMULATED_VERSION KHAOS_SET_VERSION(0, 0, 0)
01433 #define KHAOS_LLVM_EMULATED_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)
01434
01435 //Set to 0 to avoid warning
01436 #if !defined(KHAOS_COMPILER_ACC)
01437 #define KHAOS_COMPILER_ACC 0
01438 #endif
01439 #if !defined(KHAOS_COMPILER_AltiumMicroblaze)

```

```
01440 #define KHAOS_COMPILER_AltiumMicroblaze 0
01441 #endif
01442 #if !defined(KHAOS_COMPILER_Altium)
01443 #define KHAOS_COMPILER_Altium 0
01444 #endif
01445 #if !defined(KHAOS_COMPILER_Amsterdam)
01446 #define KHAOS_COMPILER_Amsterdam 0
01447 #endif
01448 #if !defined(KHAOS_COMPILER_ARM)
01449 #define KHAOS_COMPILER_ARM 0
01450 #endif
01451 #if !defined(KHAOS_COMPILER_Aztec)
01452 #define KHAOS_COMPILER_Aztec 0
01453 #endif
01454 #if !defined(KHAOS_COMPILER_Borland)
01455 #define KHAOS_COMPILER_Borland 0
01456 #endif
01457 #if !defined(KHAOS_COMPILER_CC65)
01458 #define KHAOS_COMPILER_CC65 0
01459 #endif
01460 #if !defined(KHAOS_COMPILER_Comeau)
01461 #define KHAOS_COMPILER_Comeau 0
01462 #endif
01463 #if !defined(KHAOS_COMPILER_Compacq)
01464 #define KHAOS_COMPILER_Compacq 0
01465 #endif
01466 #if !defined(KHAOS_COMPILER_Convex)
01467 #define KHAOS_COMPILER_Convex 0
01468 #endif
01469 #if !defined(KHAOS_COMPILER_Compcert)
01470 #define KHAOS_COMPILER_Compcert 0
01471 #endif
01472 #if !defined(KHAOS_COMPILER_Coverity)
01473 #define KHAOS_COMPILER_Coverity 0
01474 #endif
01475 #if !defined(KHAOS_COMPILER_CrayC)
01476 #define KHAOS_COMPILER_CrayC 0
01477 #endif
01478 #if !defined(KHAOS_COMPILER_Diab)
01479 #define KHAOS_COMPILER_Diab 0
01480 #endif
01481 #if !defined(KHAOS_COMPILER_DICE)
01482 #define KHAOS_COMPILER_DICE 0
01483 #endif
01484 #if !defined(KHAOS_COMPILER_DigitalMars)
01485 #define KHAOS_COMPILER_DigitalMars 0
01486 #endif
01487 #if !defined(KHAOS_COMPILER_Dignus)
01488 #define KHAOS_COMPILER_Dignus 0
01489 #endif
01490 #if !defined(KHAOS_COMPILER_DJGPP)
01491 #define KHAOS_COMPILER_DJGPP 0
01492 #endif
01493 #if !defined(KHAOS_COMPILER_EDG)
01494 #define KHAOS_COMPILER_EDG 0
01495 #endif
01496 #if !defined(KHAOS_COMPILER_EKOPath)
01497 #define KHAOS_COMPILER_EKOPath 0
01498 #endif
01499 #if !defined(KHAOS_COMPILER_Emscripten)
01500 #define KHAOS_COMPILER_Emscripten 0
01501 #endif
01502 #if !defined(KHAOS_COMPILER_Fujitsu)
01503 #define KHAOS_COMPILER_Fujitsu 0
01504 #endif
01505 #if !defined(KHAOS_COMPILER_GCCXML)
01506 #define KHAOS_COMPILER_GCCXML 0
01507 #endif
01508 #if !defined(KHAOS_COMPILER_GreenHills)
01509 #define KHAOS_COMPILER_GreenHills 0
01510 #endif
01511 #if !defined(KHAOS_COMPILER_HPCC)
01512 #define KHAOS_COMPILER_HPCC 0
01513 #endif
01514 #if !defined(KHAOS_COMPILER_HPACC)
01515 #define KHAOS_COMPILER_HPACC 0
01516 #endif
01517 #if !defined(KHAOS_COMPILER_IAR)
01518 #define KHAOS_COMPILER_IAR 0
01519 #endif
01520 #if !defined(KHAOS_COMPILER_IBM)
01521 #define KHAOS_COMPILER_IBM 0
01522 #endif
01523 #if !defined(KHAOS_COMPILER_IBMclang)
01524 #define KHAOS_COMPILER_IBMclang 0
01525 #endif
01526 #if !defined(KHAOS_COMPILER_IBMLegacy)
```

```
01527 #define KHAOS_COMPILER_IBMLegacy 0
01528 #endif
01529 #if !defined(KHAOS_COMPILER_IBMC370)
01530 #define KHAOS_COMPILER_IBMC370 0
01531 #endif
01532 #if !defined(KHAOS_COMPILER_IBMMVS)
01533 #define KHAOS_COMPILER_IBMMVS 0
01534 #endif
01535 #if !defined(KHAOS_COMPILER_IBMOS390)
01536 #define KHAOS_COMPILER_IBMOS390 0
01537 #endif
01538 #if !defined(KHAOS_COMPILER_IBMZOS)
01539 #define KHAOS_COMPILER_IBMZOS 0
01540 #endif
01541 #if !defined(KHAOS_COMPILER_ImageCraft)
01542 #define KHAOS_COMPILER_ImageCraft 0
01543 #endif
01544 #if !defined(KHAOS_COMPILER_Intel)
01545 #define KHAOS_COMPILER_Intel 0
01546 #endif
01547 #if !defined(KHAOS_COMPILER_Kai)
01548 #define KHAOS_COMPILER_Kai 0
01549 #endif
01550 #if !defined(KHAOS_COMPILER_KeilCarm)
01551 #define KHAOS_COMPILER_KeilCarm 0
01552 #endif
01553 #if !defined(KHAOS_COMPILER_KeilC166)
01554 #define KHAOS_COMPILER_KeilC166 0
01555 #endif
01556 #if !defined(KHAOS_COMPILER_KeilC51)
01557 #define KHAOS_COMPILER_KeilC51 0
01558 #endif
01559 #if !defined(KHAOS_COMPILER_LCC)
01560 #define KHAOS_COMPILER_LCC 0
01561 #endif
01562 #if !defined(KHAOS_COMPILER_Metaware)
01563 #define KHAOS_COMPILER_Metaware 0
01564 #endif
01565 #if !defined(KHAOS_COMPILER_Metrowerks)
01566 #define KHAOS_COMPILER_Metrowerks 0
01567 #endif
01568 #if !defined(KHAOS_COMPILER_MSVC)
01569 #define KHAOS_COMPILER_MSVC 0
01570 #endif
01571 #if !defined(KHAOS_COMPILER_Microtec)
01572 #define KHAOS_COMPILER_Microtec 0
01573 #endif
01574 #if !defined(KHAOS_COMPILER_Microway)
01575 #define KHAOS_COMPILER_Microway 0
01576 #endif
01577 #if !defined(KHAOS_COMPILER_MIPSpro)
01578 #define KHAOS_COMPILER_MIPSpro 0
01579 #endif
01580 #if !defined(KHAOS_COMPILER_Miracle)
01581 #define KHAOS_COMPILER_Miracle 0
01582 #endif
01583 #if !defined(KHAOS_COMPILER_MPW)
01584 #define KHAOS_COMPILER_MPW 0
01585 #endif
01586 #if !defined(KHAOS_COMPILER_Norcroft)
01587 #define KHAOS_COMPILER_Norcroft 0
01588 #endif
01589 #if !defined(KHAOS_COMPILER_NWCC)
01590 #define KHAOS_COMPILER_NWCC 0
01591 #endif
01592 #if !defined(KHAOS_COMPILER_Open64)
01593 #define KHAOS_COMPILER_Open64 0
01594 #endif
01595 #if !defined(KHAOS_COMPILER_Oracle)
01596 #define KHAOS_COMPILER_Oracle 0
01597 #endif
01598 #if !defined(KHAOS_COMPILER_Solaris)
01599 #define KHAOS_COMPILER_Solaris 0
01600 #endif
01601 #if !defined(KHAOS_COMPILER_Pacific)
01602 #define KHAOS_COMPILER_Pacific 0
01603 #endif
01604 #if !defined(KHAOS_COMPILER_Palm)
01605 #define KHAOS_COMPILER_Palm 0
01606 #endif
01607 #if !defined(KHAOS_COMPILER_Pelles)
01608 #define KHAOS_COMPILER_Pelles 0
01609 #endif
01610 #if !defined(KHAOS_COMPILER_Portland)
01611 #define KHAOS_COMPILER_Portland 0
01612 #endif
01613 #if !defined(KHAOS_COMPILER_Renesas)
```

```

01614 #define KHAOS_COMPILER_Renesas 0
01615 #endif
01616 #if !defined(KHAOS_COMPILER_SASC)
01617 #define KHAOS_COMPILER_SASC 0
01618 #endif
01619 #if !defined(KHAOS_COMPILER_SCO)
01620 #define KHAOS_COMPILER_SCO 0
01621 #endif
01622 #if !defined(KHAOS_COMPILER_SDCC)
01623 #define KHAOS_COMPILER_SDCC 0
01624 #endif
01625 #if !defined(KHAOS_COMPILER_SNC)
01626 #define KHAOS_COMPILER_SNC 0
01627 #endif
01628 #if !defined(KHAOS_COMPILER_Stratus)
01629 #define KHAOS_COMPILER_Stratus 0
01630 #endif
01631 #if !defined(KHAOS_COMPILER_StratusKR)
01632 #define KHAOS_COMPILER_StratusKR 0
01633 #endif
01634 #if !defined(KHAOS_COMPILER_StratusANSI)
01635 #define KHAOS_COMPILER_StratusANSI 0
01636 #endif
01637 #if !defined(KHAOS_COMPILER_Symantec)
01638 #define KHAOS_COMPILER_Symantec 0
01639 #endif
01640 #if !defined(KHAOS_COMPILER_Tendra)
01641 #define KHAOS_COMPILER_Tendra 0
01642 #endif
01643 #if !defined(KHAOS_COMPILER_TI)
01644 #define KHAOS_COMPILER_TI 0
01645 #endif
01646 #if !defined(KHAOS_COMPILER_Think)
01647 #define KHAOS_COMPILER_Think 0
01648 #endif
01649 #if !defined(KHAOS_COMPILER_Tiny)
01650 #define KHAOS_COMPILER_Tiny 0
01651 #endif
01652 #if !defined(KHAOS_COMPILER_Turbo)
01653 #define KHAOS_COMPILER_Turbo 0
01654 #endif
01655 #if !defined(KHAOS_COMPILER_Ultimate)
01656 #define KHAOS_COMPILER_Ultimate 0
01657 #endif
01658 #if !defined(KHAOS_COMPILER_USLC)
01659 #define KHAOS_COMPILER_USLC 0
01660 #endif
01661 #if !defined(KHAOS_COMPILER_VBCC)
01662 #define KHAOS_COMPILER_VBCC 0
01663 #endif
01664 #if !defined(KHAOS_COMPILER_Watcom)
01665 #define KHAOS_COMPILER_Watcom 0
01666 #endif
01667 #if !defined(KHAOS_COMPILER_Zortech)
01668 #define KHAOS_COMPILER_Zortech 0
01669 #endif
01670 #if !defined(KHAOS_COMPILER_AppleClang)
01671 #define KHAOS_COMPILER_AppleClang 0
01672 #endif
01673 #if !defined(KHAOS_COMPILER_Clang)
01674 #define KHAOS_COMPILER_Clang 0
01675 #endif
01676 #if !defined(KHAOS_COMPILER_LLVM)
01677 #define KHAOS_COMPILER_LLVM 0
01678 #endif
01679 #if !defined(KHAOS_COMPILER_GCC)
01680 #define KHAOS_COMPILER_GCC 0
01681 #endif
01682 #if !defined(KHAOS_COMPILER_Unknown)
01683 #define KHAOS_COMPILER_Unknown 0
01684 #endif
01685
01686 #endif

```

2.7 khaos/CStandards.h File Reference

Define the values of the official C standard.

Macros

- `#define KHAOS_STANDARD_C90 1L`
Value for the C90 standard (no value defined by the standard)
- `#define KHAOS_STANDARD_C95 199409L`
Value for the C95 standard.
- `#define KHAOS_STANDARD_C99 199901L`
Value for the C99 standard.
- `#define KHAOS_STANDARD_C11 201112L`
Value for the C11 standard.
- `#define KHAOS_STANDARD_C17 201710L`
Value for the C17 standard.

2.7.1 Detailed Description

Define the values of the official C standard.

Define macros that expand to the integer constant of type **long** whose value are the version of the C standard.

Official values for the **STDC_VERSION** macro.

Definition in file [CStandards.h](#).

2.7.2 Macro Definition Documentation

2.7.2.1 KHAOS_STANDARD_C11

```
#define KHAOS_STANDARD_C11 201112L
```

Value for the C11 standard.

Definition at line 16 of file [CStandards.h](#).

2.7.2.2 KHAOS_STANDARD_C17

```
#define KHAOS_STANDARD_C17 201710L
```

Value for the C17 standard.

Definition at line 17 of file [CStandards.h](#).

2.7.2.3 KHAOS_STANDARD_C90

```
#define KHAOS_STANDARD_C90 1L
```

Value for the C90 standard (no value defined by the standard)

Definition at line 13 of file [CStandards.h](#).

2.7.2.4 KHAOS_STANDARD_C95

```
#define KHAOS_STANDARD_C95 199409L
```

Value for the C95 standard.

Definition at line 14 of file [CStandards.h](#).

2.7.2.5 KHAOS_STANDARD_C99

```
#define KHAOS_STANDARD_C99 199901L
```

Value for the C99 standard.

Definition at line 15 of file [CStandards.h](#).

2.8 CStandards.h

[Go to the documentation of this file.](#)

```
00001 #ifndef KHAOS_CSTANDARDS_H
00002 #define KHAOS_CSTANDARDS_H
00003
00013 #define KHAOS_STANDARD_C90 1L
00014 #define KHAOS_STANDARD_C95 199409L
00015 #define KHAOS_STANDARD_C99 199901L
00016 #define KHAOS_STANDARD_C11 201112L
00017 #define KHAOS_STANDARD_C17 201710L
00018
00019 #endif
```

2.9 khaos/CXX.h File Reference

```
#include "CXXStandards.h"
```

Macros

- `#define KHAOS_CXX_STANDARD __cplusplus`
- `#define KHAOS_LANGUAGE_ISO 0`
1 if strict ISO, 0 otherwise
- `#define KHAOS_LANGUAGE_Extensions 1`
0 if strict ISO, 1 otherwise
- `#define KHAOS_LANGUAGE_Embedded 0`
0 if CPP Embedded

2.9.1 Macro Definition Documentation

2.9.1.1 KHAOS_CXX_STANDARD

```
#define KHAOS_CXX_STANDARD __cplusplus
```

Definition at line 34 of file [CXX.h](#).

2.9.1.2 KHAOS_LANGUAGE_Embedded

```
#define KHAOS_LANGUAGE_Embedded 0
```

0 if CPP Embedded

Definition at line 48 of file [CXX.h](#).

2.9.1.3 KHAOS_LANGUAGE_Extensions

```
#define KHAOS_LANGUAGE_Extensions 1
```

0 if strict ISO, 1 otherwise

Definition at line 42 of file [CXX.h](#).

2.9.1.4 KHAOS_LANGUAGE_ISO

```
#define KHAOS_LANGUAGE_ISO 0
```

1 if strict ISO, 0 otherwise

Definition at line 41 of file [CXX.h](#).

2.10 CXX.h

[Go to the documentation of this file.](#)

```

00001 #ifndef KHAOS_CXX_H
00002 #define KHAOS_CXX_H
00003
00009 #include "CXXStandards.h"
00010
00011 #if defined(_MSC_VER)
00020     #if defined(_MSVC_LANG)
00021         #define KHAOS_CXX_STANDARD _MSVC_LANG
00022         #if __cplusplus != _MSVC_LANG
00023             #if _MSC_VER >= 1914
00024                 #pragma message("[Khaos] : MSVC doesn't set the correct value for __cplusplus (cf.
https://devblogs.microsoft.com/cppblog/msvc-now-correctly-reports-__cplusplus/) !")
00025             #pragma message("[Khaos] : Use /Zc:__cplusplus option to activate it !")
00026             #else
00027                 #pragma message("[Khaos] : MSVC doesn't set the correct value for __cplusplus (cf.
https://devblogs.microsoft.com/cppblog/msvc-now-correctly-reports-__cplusplus/) !")
00028             #endif
00029         #endif
00030     #else
00031         #define KHAOS_CXX_STANDARD __cplusplus
00032     #endif
00033 #else
00034     #define KHAOS_CXX_STANDARD __cplusplus
00035 #endif
00036
00037 #if defined(__STRICT_ANSI__)
00038     #define KHAOS_LANGUAGE_ISO 1
00039     #define KHAOS_LANGUAGE_Extensions 0
00040 #else
00041     #define KHAOS_LANGUAGE_ISO 0
00042     #define KHAOS_LANGUAGE_Extensions 1
00043 #endif
00044
00045 #if defined(__embedded_cplusplus)
00046     #define KHAOS_LANGUAGE_Embedded 1
00047 #else
00048     #define KHAOS_LANGUAGE_Embedded 0
00049 #endif
00050
00051 #endif

```

2.11 khaos/CXXStandards.h File Reference

Define the version of the C++ standard.

Macros

- #define [KHAOS_STANDARD_CXX97](#) 199711L
Value for the C++97 standard.
- #define [KHAOS_STANDARD_CXX11](#) 201103L
Value for the C++11 standard.
- #define [KHAOS_STANDARD_CXX14](#) 201402L
Value for the C++14 standard.
- #define [KHAOS_STANDARD_CXX17](#) 201703L
Value for the C++17 standard.
- #define [KHAOS_STANDARD_CXX20](#) 202002L
Value for the C++20 standard.

2.11.1 Detailed Description

Define the version of the C++ standard.

Define macros that expand to the integer constant of type **long** whose value are the version of the C++ standard.

Official values for the `__cplusplus` macro.

Definition in file [CXXStandards.h](#).

2.11.2 Macro Definition Documentation

2.11.2.1 KHAOS_STANDARD_CXX11

```
#define KHAOS_STANDARD_CXX11 201103L
```

Value for the C++11 standard.

Definition at line 13 of file [CXXStandards.h](#).

2.11.2.2 KHAOS_STANDARD_CXX14

```
#define KHAOS_STANDARD_CXX14 201402L
```

Value for the C++14 standard.

Definition at line 14 of file [CXXStandards.h](#).

2.11.2.3 KHAOS_STANDARD_CXX17

```
#define KHAOS_STANDARD_CXX17 201703L
```

Value for the C++17 standard.

Definition at line 15 of file [CXXStandards.h](#).

2.11.2.4 KHAOS_STANDARD_CXX20

```
#define KHAOS_STANDARD_CXX20 202002L
```

Value for the C++20 standard.

Definition at line 16 of file [CXXStandards.h](#).

2.11.2.5 KHAOS_STANDARD_CXX97

```
#define KHAOS_STANDARD_CXX97 199711L
```

Value for the C++97 standard.

Definition at line 12 of file [CXXStandards.h](#).

2.12 CXXStandards.h

[Go to the documentation of this file.](#)

```
00001 #ifndef KHAOS_CXXSTANDARDS_H
00002 #define KHAOS_CXXSTANDARDS_H
00003
00012 #define KHAOS_STANDARD_CXX97 199711L
00013 #define KHAOS_STANDARD_CXX11 201103L
00014 #define KHAOS_STANDARD_CXX14 201402L
00015 #define KHAOS_STANDARD_CXX17 201703L
00016 #define KHAOS_STANDARD_CXX20 202002L
00017
00018 #endif
```

2.13 khaos/Features.h File Reference

```
#include "khaos/Compiler.h"
#include "khaos/Language.h"
```

Macros

- `#define __has_include(x) 0`
- `#define __has_feature_def(x) __has_feature_##x`
Implement __has_feature like Clang.
- `#define __has_feature(x) __has_feature_def(x)`
- `#define __has_cpp_attribute(x) (__has_cpp_attribute_##x)`
- `#define __has_cpp_attribute_carries_dependency 0`

2.13.1 Macro Definition Documentation

2.13.1.1 __has_cpp_attribute

```
#define __has_cpp_attribute(  
    x ) (__has_cpp_attribute_##x)
```

Definition at line 20 of file [Features.h](#).

2.13.1.2 `__has_cpp_attribute_carries_dependency`

```
#define __has_cpp_attribute_carries_dependency 0
```

Definition at line 32 of file [Features.h](#).

2.13.1.3 `__has_feature`

```
#define __has_feature(  
    x ) __has_feature_def(x)
```

Definition at line 14 of file [Features.h](#).

2.13.1.4 `__has_feature_def`

```
#define __has_feature_def(  
    x ) __has_feature_##x
```

Implement `__has_feature` like Clang.

Definition at line 13 of file [Features.h](#).

2.13.1.5 `__has_include`

```
#define __has_include(  
    x ) 0
```

Definition at line 8 of file [Features.h](#).

2.14 Features.h

[Go to the documentation of this file.](#)

```

00001 #ifndef KHAOS_FEATURES_H
00002 #define KHAOS_FEATURES_H
00003
00004 #include "khaos/Compiler.h"
00005 #include "khaos/Language.h"
00006
00007 #ifndef __has_include
00008     #define __has_include(x) 0
00009 #endif
00010
00012 #ifndef __has_feature
00013     #define __has_feature_def(x) __has_feature_##x
00014     #define __has_feature(x) __has_feature_def(x)
00015 #endif
00016
00017 #if __has_include(<version>)
00018     #include <version>
00019 #else
00020     #define __has_cpp_attribute(x) (__has_cpp_attribute_##x)
00021     /* [[carries_dependency]] */
00022     #if COMPILER_IS_GREATER_EQUAL(ClangEmulated, 3, 3, 0) && LANGUAGE_IS_GREATER_EQUAL(CXX11)
00023         #define __has_cpp_attribute_carries_dependency 200809L
00024     #endif
00025     #if COMPILER_IS_GREATER_EQUAL(GCCEmulated, 3, 3, 0) && LANGUAGE_IS_GREATER_EQUAL(CXX11)
00026         #define __has_cpp_attribute_carries_dependency 0L
00027     #endif
00028     #if COMPILER_IS_GREATER_EQUAL(MSVC, 0, 0, 1)
00029         #define __has_cpp_attribute_carries_dependency 200809L
00030     #endif
00031     #if !defined(__has_cpp_attribute_carries_dependency)
00032         #define __has_cpp_attribute_carries_dependency 0
00033     #endif
00034 #endif
00035
00036 #endif

```

2.15 khaos/Language.h File Reference

```
#include "khaos/C.h"
```

Macros

- #define [KHAOS_LANGUAGE_C](#) 1
- #define [KHAOS_LANGUAGE_Cuda](#) 0
- #define [KHAOS_LANGUAGE_ClangCuda](#) 0
- #define [KHAOS_LANGUAGE_ObjCPP](#) 0
- #define [KHAOS_LANGUAGE_ObjC](#) 0
- #define [KHAOS_LANGUAGE_Assembler](#) 0
- #define [KHAOS_LANGUAGE_Fortran](#) 0
- #define [KHAOS_LANGUAGE_CXX](#) 0
- #define [LANGUAGE_IS](#)(language)
Check the language.
- #define [LANGUAGE_IS_GREATER](#)(standard)
*Check if the C, CXX standard is greater than **standard**.*
- #define [LANGUAGE_IS_GREATER_EQUAL](#)(standard)
*Check if the C, CXX standard is greater or equal than **standard**.*
- #define [LANGUAGE_IS_EQUAL](#)(standard)
*Check if the C, CXX standard is equal than **standard**.*
- #define [LANGUAGE_IS_LOWER_EQUAL](#)(standard)
*Check if the C, CXX standard is lower or equal than **standard**.*
- #define [LANGUAGE_IS_LOWER](#)(standard)
*Check if the C, CXX standard is lower than **standard**.*

2.15.1 Macro Definition Documentation

2.15.1.1 KHAOS_LANGUAGE_Assembler

```
#define KHAOS_LANGUAGE_Assembler 0
```

Definition at line 49 of file [Language.h](#).

2.15.1.2 KHAOS_LANGUAGE_C

```
#define KHAOS_LANGUAGE_C 1
```

Definition at line 32 of file [Language.h](#).

2.15.1.3 KHAOS_LANGUAGE_ClangCuda

```
#define KHAOS_LANGUAGE_ClangCuda 0
```

Definition at line 40 of file [Language.h](#).

2.15.1.4 KHAOS_LANGUAGE_Cuda

```
#define KHAOS_LANGUAGE_Cuda 0
```

Definition at line 37 of file [Language.h](#).

2.15.1.5 KHAOS_LANGUAGE_CXX

```
#define KHAOS_LANGUAGE_CXX 0
```

Definition at line 55 of file [Language.h](#).

2.15.1.6 KHAOS_LANGUAGE_Fortran

```
#define KHAOS_LANGUAGE_Fortran 0
```

Definition at line 52 of file [Language.h](#).

2.15.1.7 KHAOS_LANGUAGE_ObjC

```
#define KHAOS_LANGUAGE_ObjC 0
```

Definition at line 46 of file [Language.h](#).

2.15.1.8 KHAOS_LANGUAGE_ObjCPP

```
#define KHAOS_LANGUAGE_ObjCPP 0
```

Definition at line 43 of file [Language.h](#).

2.15.1.9 LANGUAGE_IS

```
#define LANGUAGE_IS(  
    language )
```

Check the language.

Parameters

in	<i>language</i>	The language to test.
----	-----------------	-----------------------

Values :

Macro	(other value)
Assembler	
C	
ClangCuda	Cuda
Cuda	
CXX	
Fortran	
ObjC	
ObjCPP	

Note

Can be directly used in both preprocessor and compiler expressions.

Definition at line 79 of file [Language.h](#).

2.15.1.10 LANGUAGE_IS_EQUAL

```
#define LANGUAGE_IS_EQUAL(  
    standard )
```

Check if the C, CXX standard is equal than **standard**.

Parameters

<i>in</i>	<i>standard</i>	The language to test.
-----------	-----------------	-----------------------

Values for C and CXX : see KHAOS_LANGUAGE_IS_GREATER

Note

Can be directly used in both preprocessor and compiler expressions.

Definition at line 123 of file [Language.h](#).

2.15.1.11 LANGUAGE_IS_GREATER

```
#define LANGUAGE_IS_GREATER(  
    standard )
```

Check if the C, CXX standard is greater than **standard**.

Parameters

<i>in</i>	<i>standard</i>	The language to test.
-----------	-----------------	-----------------------

Values for C

Macro
C90
C95
C99
C11
C17

Values for CXX

Macro
CXX97
CXX11
CXX14
CXX17
CXX20

Note

Can be directly used in both preprocessor and compiler expressions.

Definition at line 105 of file [Language.h](#).

2.15.1.12 LANGUAGE_IS_GREATER_EQUAL

```
#define LANGUAGE_IS_GREATER_EQUAL(  
    standard )
```

Check if the C, CXX standard is greater or equal than **standard**.

Parameters

<code>in</code>	<code><i>standard</i></code>	The language to test.
-----------------	------------------------------	-----------------------

Values for C and CXX : see KHAOS_LANGUAGE_IS_GREATER

Note

Can be directly used in both preprocessor and compiler expressions.

Definition at line 114 of file [Language.h](#).

2.15.1.13 LANGUAGE_IS_LOWER

```
#define LANGUAGE_IS_LOWER(  
    standard )
```

Check if the C, CXX standard is lower than **standard**.

Parameters

<code>in</code>	<code><i>standard</i></code>	The language to test.
-----------------	------------------------------	-----------------------

Values for C and CXX : see KHAOS_LANGUAGE_IS_GREATER

Note

Can be directly used in both preprocessor and compiler expressions.

Definition at line 141 of file [Language.h](#).

2.15.1.14 LANGUAGE_IS_LOWER_EQUAL

```
#define LANGUAGE_IS_LOWER_EQUAL(  
    standard )
```

Check if the C, CXX standard is lower or equal than **standard**.

Parameters

in	<i>standard</i>	The language to test.
----	-----------------	-----------------------

Values for C and CXX : see KHAOS_LANGUAGE_IS_GREATER

Note

Can be directly used in both preprocessor and compiler expressions.

Definition at line 132 of file [Language.h](#).

2.16 Language.h

[Go to the documentation of this file.](#)

```
00001 #ifndef KHAOS_LANGUAGES_H
00002 #define KHAOS_LANGUAGES_H
00003
00008 #if defined(__CUDA__) || defined(__CUDACC__)
00014 #define KHAOS_LANGUAGE_Cuda 1
00015 #if defined(__clang__) && defined(__CUDA__)
00016 #define KHAOS_LANGUAGE_ClangCuda 1
00017 #endif
00018 #elif defined(__OBJC__)
00019 #if defined(__cplusplus)
00020 #define KHAOS_LANGUAGE_ObjCPP 1
00021 #else
00022 #define KHAOS_LANGUAGE_ObjC 1
00023 #endif
00024 #elif defined(__ASSEMBLER__)
00025 #define KHAOS_LANGUAGE_Assembler 1
00026 #elif defined(__GFORTRAN__)
00027 #define KHAOS_LANGUAGE_Fortran 1
00028 #elif defined(__cplusplus)
00029 #define KHAOS_LANGUAGE_CXX 1
00030 #include "khaos/CXX.h"
00031 #else
00032 #define KHAOS_LANGUAGE_C 1
00033 #include "khaos/C.h"
00034 #endif
00035
00036 #if !defined(KHAOS_LANGUAGE_Cuda)
00037 #define KHAOS_LANGUAGE_Cuda 0
```

```

00038 #endif
00039 #if !defined(KHAOS_LANGUAGE_ClangCuda)
00040 #define KHAOS_LANGUAGE_ClangCuda 0
00041 #endif
00042 #if !defined(KHAOS_LANGUAGE_ObjCPP)
00043 #define KHAOS_LANGUAGE_ObjCPP 0
00044 #endif
00045 #if !defined(KHAOS_LANGUAGE_ObjC)
00046 #define KHAOS_LANGUAGE_ObjC 0
00047 #endif
00048 #if !defined(KHAOS_LANGUAGE_Assembler)
00049 #define KHAOS_LANGUAGE_Assembler 0
00050 #endif
00051 #if !defined(KHAOS_LANGUAGE_Fortran)
00052 #define KHAOS_LANGUAGE_Fortran 0
00053 #endif
00054 #if !defined(KHAOS_LANGUAGE_CXX)
00055 #define KHAOS_LANGUAGE_CXX 0
00056 #endif
00057 #if !defined(KHAOS_LANGUAGE_C)
00058 #define KHAOS_LANGUAGE_C 0
00059 #endif
00060
00079 #define LANGUAGE_IS(language) (KHAOS_LANGUAGE_##language)
00080 #if LANGUAGE_IS(C)
00105 #define LANGUAGE_IS_GREATER(standard) (KHAOS_C_STANDARD > KHAOS_STANDARD_##standard)
00114 #define LANGUAGE_IS_GREATER_EQUAL(standard) (KHAOS_C_STANDARD >= KHAOS_STANDARD_##standard)
00123 #define LANGUAGE_IS_EQUAL(standard) (KHAOS_C_STANDARD == KHAOS_STANDARD_##standard)
00132 #define LANGUAGE_IS_LOWER_EQUAL(standard) (KHAOS_C_STANDARD <= KHAOS_STANDARD_##standard)
00141 #define LANGUAGE_IS_LOWER(standard) (KHAOS_C_STANDARD < KHAOS_STANDARD_##standard)
00142 #elif LANGUAGE_IS(CXX)
00143 #define LANGUAGE_IS_GREATER(standard) (KHAOS_CXX_STANDARD > KHAOS_STANDARD_##standard)
00144 #define LANGUAGE_IS_GREATER_EQUAL(standard) (KHAOS_CXX_STANDARD >= KHAOS_STANDARD_##standard)
00145 #define LANGUAGE_IS_EQUAL(standard) (KHAOS_CXX_STANDARD == KHAOS_STANDARD_##standard)
00146 #define LANGUAGE_IS_LOWER_EQUAL(standard) (KHAOS_CXX_STANDARD <= KHAOS_STANDARD_##standard)
00147 #define LANGUAGE_IS_LOWER(standard) (KHAOS_CXX_STANDARD < KHAOS_STANDARD_##standard)
00148 #else
00149 #define LANGUAGE_IS_GREATER(standard) (0L)
00150 #define LANGUAGE_IS_GREATER_EQUAL(standard) (0L)
00151 #define LANGUAGE_IS_EQUAL(standard) (0L)
00152 #define LANGUAGE_IS_LOWER_EQUAL(standard) (0L)
00153 #define LANGUAGE_IS_LOWER(standard) (0L)
00154 #endif
00155
00156 #endif

```

2.17 khaos/Library.h File Reference

Detect the library and define some useful macros.

```

#include "khaos/OS.h"
#include "khaos/VersionMacros.h"
#include <assert.h>
#include <stddef.h>

```

Macros

- #define [C_LIBRARY_IS](#)(library)
Detect the C library.
- #define [CXX_LIBRARY_IS](#)(library)
Detect the CXX library.
- #define [KHAOS_LIBRARYC_Bionic](#) 0
- #define [KHAOS_LIBRARYC_CloudABI](#) 0
- #define [KHAOS_LIBRARYC_GNU](#) 0
- #define [KHAOS_LIBRARYC_klibc](#) 0
- #define [KHAOS_LIBRARYC_uClibc](#) 0
- #define [KHAOS_LIBRARYC_VMC](#) 0

- `#define KHAOS_LIBRARYC_ZOS 0`
- `#define KHAOS_LIBRARYC_VERSION KHAOS_SET_VERSION(0, 0, 0)`
- `#define KHAOS_LIBRARYC_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)`
- `#define KHAOS_LIBRARYCXX_CXX 0`
- `#define KHAOS_LIBRARYCXX_Dinkumware 0`
- `#define KHAOS_LIBRARYCXX_Comeau 0`
- `#define KHAOS_LIBRARYCXX_Modena 0`
- `#define KHAOS_LIBRARYCXX_Metrowerks 0`
- `#define KHAOS_LIBRARYCXX_Roguewave 0`
- `#define KHAOS_LIBRARYCXX_SGI 0`
- `#define KHAOS_LIBRARYCXX_GNU 0`
- `#define KHAOS_LIBRARYCXX_STLport 0`
- `#define KHAOS_LIBRARYCXX_IBM 0`
- `#define KHAOS_LIBRARYCXX_MFC 0`
- `#define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)`
- `#define KHAOS_LIBRARYCXX_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)`

2.17.1 Detailed Description

Detect the library and define some useful macros.

Definition in file [Library.h](#).

2.17.2 Macro Definition Documentation

2.17.2.1 C_LIBRARY_IS

```
#define C_LIBRARY_IS(  
    library )
```

Detect the C library.

Parameters

<code>in</code>	<code><i>library</i></code>	The library name.
-----------------	-----------------------------	-------------------

Note

Use this macro only if you need preprocessor operating system checking. For example :

```
#if C_LIBRARY_IS(...)
//...
#else
//...
#endif
```

Values for the C library

Macro
Bionic

Macro
CloudABI
GNU
klibc
uClibc
VMC
ZOS

Definition at line 51 of file [Library.h](#).

2.17.2.2 CXX_LIBRARY_IS

```
#define CXX_LIBRARY_IS(  
    library )
```

Detect the CXX library.

Parameters

in	<i>library</i>	The library name.
----	----------------	-------------------

Note

Use this macro only if you need preprocessor operating system checking. For example :

```
#if CXX_LIBRARY_IS(...)  
//...  
#else  
//...  
#endif
```

Values for C++ library

Macro
CXX
Dinkumware
Comeau
Modena
Metrowerks
Roguewave
SGI
GNU
STLport
IBM
MFC

Definition at line 84 of file [Library.h](#).

2.17.2.3 KHAOS_LIBRARYC_Bionic

```
#define KHAOS_LIBRARYC_Bionic 0
```

Definition at line 177 of file [Library.h](#).

2.17.2.4 KHAOS_LIBRARYC_CloudABI

```
#define KHAOS_LIBRARYC_CloudABI 0
```

Definition at line 180 of file [Library.h](#).

2.17.2.5 KHAOS_LIBRARYC_GNU

```
#define KHAOS_LIBRARYC_GNU 0
```

Definition at line 183 of file [Library.h](#).

2.17.2.6 KHAOS_LIBRARYC_klibc

```
#define KHAOS_LIBRARYC_klibc 0
```

Definition at line 186 of file [Library.h](#).

2.17.2.7 KHAOS_LIBRARYC_uClibc

```
#define KHAOS_LIBRARYC_uClibc 0
```

Definition at line 189 of file [Library.h](#).

2.17.2.8 KHAOS_LIBRARYC_VERSION

```
#define KHAOS_LIBRARYC_VERSION KHAOS_SET_VERSION(0, 0, 0)
```

Definition at line 198 of file [Library.h](#).

2.17.2.9 KHAOS_LIBRARYC_VERSION_TWEAK

```
#define KHAOS_LIBRARYC_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)
```

Definition at line 201 of file [Library.h](#).

2.17.2.10 KHAOS_LIBRARYC_VMC

```
#define KHAOS_LIBRARYC_VMC 0
```

Definition at line 192 of file [Library.h](#).

2.17.2.11 KHAOS_LIBRARYC_ZOS

```
#define KHAOS_LIBRARYC_ZOS 0
```

Definition at line 195 of file [Library.h](#).

2.17.2.12 KHAOS_LIBRARYCXX_Comeau

```
#define KHAOS_LIBRARYCXX_Comeau 0
```

Definition at line 211 of file [Library.h](#).

2.17.2.13 KHAOS_LIBRARYCXX_CXX

```
#define KHAOS_LIBRARYCXX_CXX 0
```

Definition at line 205 of file [Library.h](#).

2.17.2.14 KHAOS_LIBRARYCXX_Dinkumware

```
#define KHAOS_LIBRARYCXX_Dinkumware 0
```

Definition at line 208 of file [Library.h](#).

2.17.2.15 KHAOS_LIBRARYCXX_GNU

```
#define KHAOS_LIBRARYCXX_GNU 0
```

Definition at line 226 of file [Library.h](#).

2.17.2.16 KHAOS_LIBRARYCXX_IBM

```
#define KHAOS_LIBRARYCXX_IBM 0
```

Definition at line 232 of file [Library.h](#).

2.17.2.17 KHAOS_LIBRARYCXX_Metrowerks

```
#define KHAOS_LIBRARYCXX_Metrowerks 0
```

Definition at line 217 of file [Library.h](#).

2.17.2.18 KHAOS_LIBRARYCXX_MFC

```
#define KHAOS_LIBRARYCXX_MFC 0
```

Definition at line 235 of file [Library.h](#).

2.17.2.19 KHAOS_LIBRARYCXX_Modena

```
#define KHAOS_LIBRARYCXX_Modena 0
```

Definition at line 214 of file [Library.h](#).

2.17.2.20 KHAOS_LIBRARYCXX_Roguewave

```
#define KHAOS_LIBRARYCXX_Roguewave 0
```

Definition at line 220 of file [Library.h](#).

2.17.2.21 KHAOS_LIBRARYCXX_SGI

```
#define KHAOS_LIBRARYCXX_SGI 0
```

Definition at line 223 of file [Library.h](#).

2.17.2.22 KHAOS_LIBRARYCXX_STLport

```
#define KHAOS_LIBRARYCXX_STLport 0
```

Definition at line 229 of file [Library.h](#).

2.17.2.23 KHAOS_LIBRARYCXX_VERSION

```
#define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
```

Definition at line 238 of file [Library.h](#).

2.17.2.24 KHAOS_LIBRARYCXX_VERSION_TWEAK

```
#define KHAOS_LIBRARYCXX_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)
```

Definition at line 241 of file [Library.h](#).

2.18 Library.h

[Go to the documentation of this file.](#)

```
00001 #ifndef KHAOS_LIBRARY_H
00002 #define KHAOS_LIBRARY_H
00003
00004 #include "khaos/OS.h"
00005 #include "khaos/VersionMacros.h"
00006
00007 #if OS_IS(UNIX)
00008     #include <sys/types.h>
00009 #endif
00010
00011 #if defined(__cplusplus)
00012     #include <cassert>
00013     #include <cstddef>
00014     #include <exception>
00015 #else
00016     #include <assert.h>
00017     #include <stddef.h>
00018 #endif
00019
00051 #define C_LIBRARY_IS(library) (KHAOS_LIBRARYC_##library == 1)
00052
00084 #define CXX_LIBRARY_IS(library) (KHAOS_LIBRARYCXX_##library == 1)
00085
00086 #if defined(__BIONIC__)
00087     #define KHAOS_LIBRARYC_Bionic 1
00088 #elif defined(__CloudABI__)
```

```

00089  #if defined(__cloudlibc__)
00090      #define KHAOS_LIBRARYC_CloudABI 1
00091      #define KHAOS_LIBRARYC_VERSION KHAOS_SET_VERSION(__cloudlibc_major__, __cloudlibc_minor__, 0)
00092  #endif
00093  #elif defined(__GLIBC__) || defined(__GNU_LIBRARY__)
00094      #define KHAOS_LIBRARYC_GNU 1
00095      #include <limits.h>
00096      #if defined(__GLIBC__)
00097          #define KHAOS_LIBRARYC_VERSION KHAOS_SET_VERSION(__GLIBC__, __GLIBC_MINOR__, 0)
00098      #else
00099          #define KHAOS_LIBRARYC_VERSION KHAOS_SET_VERSION(__GNU_LIBRARY__, __GNU_LIBRARY_MINOR__, 0)
00100      #endif
00101  #elif defined(__KLIBC__)
00102      #define KHAOS_LIBRARYC_klibc 1
00103      #define KHAOS_LIBRARYC_VERSION KHAOS_SET_VERSION(__KLIBC__, __KLIBC_MINOR__, __KLIBC_PATCHLEVEL__)
00104  #elif defined(__UCLIBC__)
00105      #define KHAOS_LIBRARYC_uClibc 1
00106      #define KHAOS_LIBRARYC_VERSION KHAOS_SET_VERSION(__UCLIBC_MAJOR__, __UCLIBC_MINOR__,
__UCLIBC_SUBLEVEL__)
00107  #elif defined(__CRTL_VER)
00108      #define KHAOS_LIBRARYC_VMC 1
00109      #define KHAOS_LIBRARYC_VERSION KHAOS_SET_VERSION((__CRTL_VER / 10000000) % 100, (__CRTL_VER /
1000000) % 100, (__CRTL_VER / 100) % 100)
00110  #elif defined(__LIBREL__)
00111      #define KHAOS_LIBRARYC_ZOS 1
00112      #if defined(__LIBREL__)
00113          #define KHAOS_LIBRARYC_VERSION KHAOS_SET_VERSION((__LIBREL__ / 1000000) % 10, (__LIBREL__ / 10000)
% 100, __LIBREL__ % 10000)
00114      #elif defined(__TARGET_LIB__)
00115          #define KHAOS_LIBRARYC_VERSION KHAOS_SET_VERSION((__TARGET_LIB__ / 1000000) % 10, (__TARGET_LIB__
/ 10000) % 100, __TARGET_LIB__ % 10000)
00116      #endif
00117  #endif
00118
00119  #if defined(_LIBCPP_VERSION)
00120      #define KHAOS_LIBRARYCXX_CXX 1
00121      #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION((_LIBCPP_VERSION / 1000) % 100, (_LIBCPP_VERSION)
% 1000, 0)
00122  #elif (defined(_YVALS) && !defined(__IBMCPP__)) || defined(_CPPLIB_VER)
00123      #define KHAOS_LIBRARYCXX_Dinkumware 1
00124      #if defined(_CPPLIB_VER)
00125          #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION((_CPPLIB_VER / 100) % 100, _CPPLIB_VER % 100,
0)
00126      #endif
00127  #elif defined(__LIBCOMO__)
00128      #define KHAOS_LIBRARYCXX_Comeau 1
00129      #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(__LIBCOMO_VERSION__, 0, 0)
00130  #elif defined(_MSIPL_COMPILE_H) || defined(__MSIPL_COMPILE_H)
00131      #define KHAOS_LIBRARYCXX_Modena 1
00132  #elif defined(__MSL_CPP__) || defined(__MSL__)
00133      #define KHAOS_LIBRARYCXX_Metrowerks 1
00134      #if defined(__MSL_CPP__)
00135          #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION((__MSL_CPP__ >> 12) % 10, (__MSL_CPP__ >> 8) %
10, __MSL_CPP__ % 100)
00136      #else
00137          #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION((__MSL__ >> 12) % 10, (__MSL__ >> 8) % 10,
__MSL__ % 100)
00138      #endif
00139  #elif defined(__STD_RWCOMPILER_H__) || defined(_RWSTD_VER)
00140      #define KHAOS_LIBRARYCXX_Roguewave 1
00141      #if defined(_RWSTD_VER)
00142          #if _RWSTD_VER < 0x010000
00143              #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION((_RWSTD_VER >> 12) % 100, (_RWSTD_VER >> 4) %
100, _RWSTD_VER % 10)
00144          #else
00145              #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION((_RWSTD_VER >> 16) % 100, (_RWSTD_VER >> 8) %
100, _RWSTD_VER % 100)
00146          #endif
00147      #endif
00148  #elif defined(__STL_CONFIG_H)
00149      #define KHAOS_LIBRARYCXX_SGI 1
00150      #if defined(__SGI_STL)
00151          #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION((__SGI_STL >> 8) % 10, (__SGI_STL >> 4) % 10,
__SGI_STL % 10)
00152      #endif
00153  #elif defined(__GLIBCPP__) || defined(__GLIBCXX__)
00154      #define KHAOS_LIBRARYCXX_GNU 1
00155      #if defined(__GLIBCXX__)
00156          #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION((__GLIBCXX__ / 10000) % 10000, (__GLIBCXX__ /
100) % 100, __GLIBCXX__ % 100)
00157      #else
00158          #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION((__GLIBCPP__ / 10000) % 10000, (__GLIBCPP__ /
100) % 100, __GLIBCPP__ % 100)
00159      #endif
00160  #elif defined(__SGI_STL_PORT) || defined(_STLPORT_VERSION)
00161      #define KHAOS_LIBRARYCXX_STLport 1
00162      #if defined(_STLPORT_MAJOR)

```

```

00163     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(_STLPORT_MAJOR, _STLPORT_MINOR,
00164     _STLPORT_PATCHLEVEL)
00165     #elif defined(_STLPORT_VERSION)
00166     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION((_STLPORT_VERSION » 8) % 10, (_STLPORT_VERSION
00167     » 4) % 10, _STLPORT_VERSION % 10)
00168     #else
00169     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION((__SGI_STL_PORT » 8) % 10, (__SGI_STL_PORT » 4)
00170     % 10, __SGI_STL_PORT % 10)
00171     #endif
00172     #elif defined(__IBMCPP__)
00173     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00174     #endif
00175     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00176     #endif
00177     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00178     #endif
00179     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00180     #endif
00181     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00182     #endif
00183     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00184     #endif
00185     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00186     #endif
00187     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00188     #endif
00189     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00190     #endif
00191     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00192     #endif
00193     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00194     #endif
00195     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00196     #endif
00197     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00198     #endif
00199     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00200     #endif
00201     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00202     #endif
00203     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00204     #endif
00205     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00206     #endif
00207     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00208     #endif
00209     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00210     #endif
00211     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00212     #endif
00213     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00214     #endif
00215     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00216     #endif
00217     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00218     #endif
00219     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00220     #endif
00221     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00222     #endif
00223     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00224     #endif
00225     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00226     #endif
00227     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00228     #endif
00229     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00230     #endif
00231     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00232     #endif
00233     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00234     #endif
00235     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00236     #endif
00237     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00238     #endif
00239     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00240     #endif
00241     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00242     #endif
00243     #define KHAOS_LIBRARYCXX_VERSION KHAOS_SET_VERSION(0, 0, 0)
00244     #endif

```

2.19 khaos/OS.h File Reference

Detect the OS.

```
#include "VersionMacros.h"
```

Macros

- `#define OS_IS(os)`
Detect the operating system.
- `#define KHAOS_OS_AIX 0`
- `#define KHAOS_OS_AIX4p3 0`
- `#define KHAOS_OS_AIX4p1 0`
- `#define KHAOS_OS_AIX3p2 0`
- `#define KHAOS_OS_AIX3 0`
- `#define KHAOS_OS_AmigaOS 0`
- `#define KHAOS_OS_BEOS 0`
- `#define KHAOS_OS_BSDi 0`
- `#define KHAOS_OS_DragonFly 0`
- `#define KHAOS_OS_FreeBSD 0`
- `#define KHAOS_OS_FreeBSD4p10 0`
- `#define KHAOS_OS_FreeBSD4p11 0`
- `#define KHAOS_OS_NetBSD 0`
- `#define KHAOS_OS_NetBSD0p8 0`
- `#define KHAOS_OS_NetBSD0p9 0`
- `#define KHAOS_OS_NetBSD1 0`
- `#define KHAOS_OS_OpenBSD 0`
- `#define KHAOS_OS_OpenBSD2_0 0`
- `#define KHAOS_OS_OpenBSD2_1 0`
- `#define KHAOS_OS_OpenBSD2_2 0`
- `#define KHAOS_OS_OpenBSD2_3 0`
- `#define KHAOS_OS_OpenBSD2_4 0`
- `#define KHAOS_OS_OpenBSD2_5 0`
- `#define KHAOS_OS_OpenBSD2_6 0`
- `#define KHAOS_OS_OpenBSD2_7 0`
- `#define KHAOS_OS_OpenBSD2_8 0`
- `#define KHAOS_OS_OpenBSD2_9 0`
- `#define KHAOS_OS_OpenBSD3_0 0`
- `#define KHAOS_OS_OpenBSD3_1 0`
- `#define KHAOS_OS_OpenBSD3_2 0`
- `#define KHAOS_OS_OpenBSD3_3 0`
- `#define KHAOS_OS_OpenBSD3_4 0`
- `#define KHAOS_OS_OpenBSD3_5 0`
- `#define KHAOS_OS_OpenBSD3_6 0`
- `#define KHAOS_OS_OpenBSD3_7 0`
- `#define KHAOS_OS_OpenBSD3_8 0`
- `#define KHAOS_OS_OpenBSD3_9 0`
- `#define KHAOS_OS_OpenBSD4_0 0`
- `#define KHAOS_OS_OpenBSD4_1 0`
- `#define KHAOS_OS_OpenBSD4_2 0`
- `#define KHAOS_OS_OpenBSD4_3 0`
- `#define KHAOS_OS_OpenBSD4_4 0`

- `#define KHAOS_OS_OpenBSD4_5 0`
- `#define KHAOS_OS_OpenBSD4_6 0`
- `#define KHAOS_OS_OpenBSD4_7 0`
- `#define KHAOS_OS_OpenBSD4_8 0`
- `#define KHAOS_OS_OpenBSD4_9 0`
- `#define KHAOS_OS_OpenBSD5_0 0`
- `#define KHAOS_OS_OpenBSD5_1 0`
- `#define KHAOS_OS_OpenBSD5_2 0`
- `#define KHAOS_OS_OpenBSD5_3 0`
- `#define KHAOS_OS_OpenBSD5_4 0`
- `#define KHAOS_OS_OpenBSD5_5 0`
- `#define KHAOS_OS_OpenBSD5_6 0`
- `#define KHAOS_OS_OpenBSD5_7 0`
- `#define KHAOS_OS_OpenBSD5_8 0`
- `#define KHAOS_OS_OpenBSD5_9 0`
- `#define KHAOS_OS_OpenBSD6_0 0`
- `#define KHAOS_OS_OpenBSD6_1 0`
- `#define KHAOS_OS_OpenBSD6_2 0`
- `#define KHAOS_OS_OpenBSD6_3 0`
- `#define KHAOS_OS_OpenBSD6_4 0`
- `#define KHAOS_OS_OpenBSD6_5 0`
- `#define KHAOS_OS_OpenBSD6_6 0`
- `#define KHAOS_OS_OpenBSD6_7 0`
- `#define KHAOS_OS_OpenBSD6_8 0`
- `#define KHAOS_OS_OpenBSD6_9 0`
- `#define KHAOS_OS_CYGWIN 0`
- `#define KHAOS_OS_HAIKU 0`
- `#define KHAOS_OS_HPUX 0`
- `#define KHAOS_OS_iOS 0`
- `#define KHAOS_OS_IRIX 0`
- `#define KHAOS_OS_Linux 0`
- `#define KHAOS_OS_MacOS 0`
- `#define KHAOS_OS_OS400 0`
- `#define KHAOS_OS_QNX 0`
- `#define KHAOS_OS_QNX4 0`
- `#define KHAOS_OS_Solaris 0`
- `#define KHAOS_OS_VMS 0`
- `#define KHAOS_OS_Windows 0`
- `#define KHAOS_OS_BSD 0`
- `#define KHAOS_OS_BSD4p4 0`
- `#define KHAOS_OS_BSD4p3 0`
- `#define KHAOS_OS_BSD4p2 0`
- `#define KHAOS_OS_UNIX 0`
- `#define KHAOS_OS_SVR4 0`
- `#define KHAOS_OS_VERSION KHAOS_SET_VERSION(0, 0, 0)`
- `#define KHAOS_OS_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)`

2.19.1 Detailed Description

Detect the OS.

Definition in file [OS.h](#).

2.19.2 Macro Definition Documentation

2.19.2.1 KHAOS_OS_AIX

```
#define KHAOS_OS_AIX 0
```

Definition at line [309](#) of file [OS.h](#).

2.19.2.2 KHAOS_OS_AIX3

```
#define KHAOS_OS_AIX3 0
```

Definition at line [321](#) of file [OS.h](#).

2.19.2.3 KHAOS_OS_AIX3p2

```
#define KHAOS_OS_AIX3p2 0
```

Definition at line [318](#) of file [OS.h](#).

2.19.2.4 KHAOS_OS_AIX4p1

```
#define KHAOS_OS_AIX4p1 0
```

Definition at line [315](#) of file [OS.h](#).

2.19.2.5 KHAOS_OS_AIX4p3

```
#define KHAOS_OS_AIX4p3 0
```

Definition at line [312](#) of file [OS.h](#).

2.19.2.6 KHAOS_OS_AmigaOS

```
#define KHAOS_OS_AmigaOS 0
```

Definition at line 324 of file [OS.h](#).

2.19.2.7 KHAOS_OS_BEOS

```
#define KHAOS_OS_BEOS 0
```

Definition at line 327 of file [OS.h](#).

2.19.2.8 KHAOS_OS_BSD

```
#define KHAOS_OS_BSD 0
```

Definition at line 549 of file [OS.h](#).

2.19.2.9 KHAOS_OS_BSD4p2

```
#define KHAOS_OS_BSD4p2 0
```

Definition at line 558 of file [OS.h](#).

2.19.2.10 KHAOS_OS_BSD4p3

```
#define KHAOS_OS_BSD4p3 0
```

Definition at line 555 of file [OS.h](#).

2.19.2.11 KHAOS_OS_BSD4p4

```
#define KHAOS_OS_BSD4p4 0
```

Definition at line 552 of file [OS.h](#).

2.19.2.12 KHAOS_OS_BSDi

```
#define KHAOS_OS_BSDi 0
```

Definition at line 330 of file [OS.h](#).

2.19.2.13 KHAOS_OS_CYGWIN

```
#define KHAOS_OS_CYGWIN 0
```

Definition at line 510 of file [OS.h](#).

2.19.2.14 KHAOS_OS_DragonFly

```
#define KHAOS_OS_DragonFly 0
```

Definition at line 333 of file [OS.h](#).

2.19.2.15 KHAOS_OS_FreeBSD

```
#define KHAOS_OS_FreeBSD 0
```

Definition at line 336 of file [OS.h](#).

2.19.2.16 KHAOS_OS_FreeBSD4p10

```
#define KHAOS_OS_FreeBSD4p10 0
```

Definition at line 339 of file [OS.h](#).

2.19.2.17 KHAOS_OS_FreeBSD4p11

```
#define KHAOS_OS_FreeBSD4p11 0
```

Definition at line 342 of file [OS.h](#).

2.19.2.18 KHAOS_OS_HAIKU

```
#define KHAOS_OS_HAIKU 0
```

Definition at line 513 of file [OS.h](#).

2.19.2.19 KHAOS_OS_HPUX

```
#define KHAOS_OS_HPUX 0
```

Definition at line 516 of file [OS.h](#).

2.19.2.20 KHAOS_OS_iOS

```
#define KHAOS_OS_iOS 0
```

Definition at line 519 of file [OS.h](#).

2.19.2.21 KHAOS_OS_IRIX

```
#define KHAOS_OS_IRIX 0
```

Definition at line 522 of file [OS.h](#).

2.19.2.22 KHAOS_OS_Linux

```
#define KHAOS_OS_Linux 0
```

Definition at line 525 of file [OS.h](#).

2.19.2.23 KHAOS_OS_MacOS

```
#define KHAOS_OS_MacOS 0
```

Definition at line 528 of file [OS.h](#).

2.19.2.24 KHAOS_OS_NetBSD

```
#define KHAOS_OS_NetBSD 0
```

Definition at line 345 of file [OS.h](#).

2.19.2.25 KHAOS_OS_NetBSD0p8

```
#define KHAOS_OS_NetBSD0p8 0
```

Definition at line 348 of file [OS.h](#).

2.19.2.26 KHAOS_OS_NetBSD0p9

```
#define KHAOS_OS_NetBSD0p9 0
```

Definition at line 351 of file [OS.h](#).

2.19.2.27 KHAOS_OS_NetBSD1

```
#define KHAOS_OS_NetBSD1 0
```

Definition at line 354 of file [OS.h](#).

2.19.2.28 KHAOS_OS_OpenBSD

```
#define KHAOS_OS_OpenBSD 0
```

Definition at line 357 of file [OS.h](#).

2.19.2.29 KHAOS_OS_OpenBSD2_0

```
#define KHAOS_OS_OpenBSD2_0 0
```

Definition at line 360 of file [OS.h](#).

2.19.2.30 KHAOS_OS_OpenBSD2_1

```
#define KHAOS_OS_OpenBSD2_1 0
```

Definition at line 363 of file [OS.h](#).

2.19.2.31 KHAOS_OS_OpenBSD2_2

```
#define KHAOS_OS_OpenBSD2_2 0
```

Definition at line 366 of file [OS.h](#).

2.19.2.32 KHAOS_OS_OpenBSD2_3

```
#define KHAOS_OS_OpenBSD2_3 0
```

Definition at line 369 of file [OS.h](#).

2.19.2.33 KHAOS_OS_OpenBSD2_4

```
#define KHAOS_OS_OpenBSD2_4 0
```

Definition at line 372 of file [OS.h](#).

2.19.2.34 KHAOS_OS_OpenBSD2_5

```
#define KHAOS_OS_OpenBSD2_5 0
```

Definition at line 375 of file [OS.h](#).

2.19.2.35 KHAOS_OS_OpenBSD2_6

```
#define KHAOS_OS_OpenBSD2_6 0
```

Definition at line 378 of file [OS.h](#).

2.19.2.36 KHAOS_OS_OpenBSD2_7

```
#define KHAOS_OS_OpenBSD2_7 0
```

Definition at line 381 of file [OS.h](#).

2.19.2.37 KHAOS_OS_OpenBSD2_8

```
#define KHAOS_OS_OpenBSD2_8 0
```

Definition at line 384 of file [OS.h](#).

2.19.2.38 KHAOS_OS_OpenBSD2_9

```
#define KHAOS_OS_OpenBSD2_9 0
```

Definition at line 387 of file [OS.h](#).

2.19.2.39 KHAOS_OS_OpenBSD3_0

```
#define KHAOS_OS_OpenBSD3_0 0
```

Definition at line 390 of file [OS.h](#).

2.19.2.40 KHAOS_OS_OpenBSD3_1

```
#define KHAOS_OS_OpenBSD3_1 0
```

Definition at line 393 of file [OS.h](#).

2.19.2.41 KHAOS_OS_OpenBSD3_2

```
#define KHAOS_OS_OpenBSD3_2 0
```

Definition at line 396 of file [OS.h](#).

2.19.2.42 KHAOS_OS_OpenBSD3_3

```
#define KHAOS_OS_OpenBSD3_3 0
```

Definition at line 399 of file [OS.h](#).

2.19.2.43 KHAOS_OS_OpenBSD3_4

```
#define KHAOS_OS_OpenBSD3_4 0
```

Definition at line 402 of file [OS.h](#).

2.19.2.44 KHAOS_OS_OpenBSD3_5

```
#define KHAOS_OS_OpenBSD3_5 0
```

Definition at line 405 of file [OS.h](#).

2.19.2.45 KHAOS_OS_OpenBSD3_6

```
#define KHAOS_OS_OpenBSD3_6 0
```

Definition at line 408 of file [OS.h](#).

2.19.2.46 KHAOS_OS_OpenBSD3_7

```
#define KHAOS_OS_OpenBSD3_7 0
```

Definition at line 411 of file [OS.h](#).

2.19.2.47 KHAOS_OS_OpenBSD3_8

```
#define KHAOS_OS_OpenBSD3_8 0
```

Definition at line 414 of file [OS.h](#).

2.19.2.48 KHAOS_OS_OpenBSD3_9

```
#define KHAOS_OS_OpenBSD3_9 0
```

Definition at line 417 of file [OS.h](#).

2.19.2.49 KHAOS_OS_OpenBSD4_0

```
#define KHAOS_OS_OpenBSD4_0 0
```

Definition at line 420 of file [OS.h](#).

2.19.2.50 KHAOS_OS_OpenBSD4_1

```
#define KHAOS_OS_OpenBSD4_1 0
```

Definition at line 423 of file [OS.h](#).

2.19.2.51 KHAOS_OS_OpenBSD4_2

```
#define KHAOS_OS_OpenBSD4_2 0
```

Definition at line 426 of file [OS.h](#).

2.19.2.52 KHAOS_OS_OpenBSD4_3

```
#define KHAOS_OS_OpenBSD4_3 0
```

Definition at line 429 of file [OS.h](#).

2.19.2.53 KHAOS_OS_OpenBSD4_4

```
#define KHAOS_OS_OpenBSD4_4 0
```

Definition at line 432 of file [OS.h](#).

2.19.2.54 KHAOS_OS_OpenBSD4_5

```
#define KHAOS_OS_OpenBSD4_5 0
```

Definition at line 435 of file [OS.h](#).

2.19.2.55 KHAOS_OS_OpenBSD4_6

```
#define KHAOS_OS_OpenBSD4_6 0
```

Definition at line 438 of file [OS.h](#).

2.19.2.56 KHAOS_OS_OpenBSD4_7

```
#define KHAOS_OS_OpenBSD4_7 0
```

Definition at line 441 of file [OS.h](#).

2.19.2.57 KHAOS_OS_OpenBSD4_8

```
#define KHAOS_OS_OpenBSD4_8 0
```

Definition at line 444 of file [OS.h](#).

2.19.2.58 KHAOS_OS_OpenBSD4_9

```
#define KHAOS_OS_OpenBSD4_9 0
```

Definition at line 447 of file [OS.h](#).

2.19.2.59 KHAOS_OS_OpenBSD5_0

```
#define KHAOS_OS_OpenBSD5_0 0
```

Definition at line 450 of file [OS.h](#).

2.19.2.60 KHAOS_OS_OpenBSD5_1

```
#define KHAOS_OS_OpenBSD5_1 0
```

Definition at line [453](#) of file [OS.h](#).

2.19.2.61 KHAOS_OS_OpenBSD5_2

```
#define KHAOS_OS_OpenBSD5_2 0
```

Definition at line [456](#) of file [OS.h](#).

2.19.2.62 KHAOS_OS_OpenBSD5_3

```
#define KHAOS_OS_OpenBSD5_3 0
```

Definition at line [459](#) of file [OS.h](#).

2.19.2.63 KHAOS_OS_OpenBSD5_4

```
#define KHAOS_OS_OpenBSD5_4 0
```

Definition at line [462](#) of file [OS.h](#).

2.19.2.64 KHAOS_OS_OpenBSD5_5

```
#define KHAOS_OS_OpenBSD5_5 0
```

Definition at line [465](#) of file [OS.h](#).

2.19.2.65 KHAOS_OS_OpenBSD5_6

```
#define KHAOS_OS_OpenBSD5_6 0
```

Definition at line [468](#) of file [OS.h](#).

2.19.2.66 KHAOS_OS_OpenBSD5_7

```
#define KHAOS_OS_OpenBSD5_7 0
```

Definition at line 471 of file [OS.h](#).

2.19.2.67 KHAOS_OS_OpenBSD5_8

```
#define KHAOS_OS_OpenBSD5_8 0
```

Definition at line 474 of file [OS.h](#).

2.19.2.68 KHAOS_OS_OpenBSD5_9

```
#define KHAOS_OS_OpenBSD5_9 0
```

Definition at line 477 of file [OS.h](#).

2.19.2.69 KHAOS_OS_OpenBSD6_0

```
#define KHAOS_OS_OpenBSD6_0 0
```

Definition at line 480 of file [OS.h](#).

2.19.2.70 KHAOS_OS_OpenBSD6_1

```
#define KHAOS_OS_OpenBSD6_1 0
```

Definition at line 483 of file [OS.h](#).

2.19.2.71 KHAOS_OS_OpenBSD6_2

```
#define KHAOS_OS_OpenBSD6_2 0
```

Definition at line 486 of file [OS.h](#).

2.19.2.72 KHAOS_OS_OpenBSD6_3

```
#define KHAOS_OS_OpenBSD6_3 0
```

Definition at line [489](#) of file [OS.h](#).

2.19.2.73 KHAOS_OS_OpenBSD6_4

```
#define KHAOS_OS_OpenBSD6_4 0
```

Definition at line [492](#) of file [OS.h](#).

2.19.2.74 KHAOS_OS_OpenBSD6_5

```
#define KHAOS_OS_OpenBSD6_5 0
```

Definition at line [495](#) of file [OS.h](#).

2.19.2.75 KHAOS_OS_OpenBSD6_6

```
#define KHAOS_OS_OpenBSD6_6 0
```

Definition at line [498](#) of file [OS.h](#).

2.19.2.76 KHAOS_OS_OpenBSD6_7

```
#define KHAOS_OS_OpenBSD6_7 0
```

Definition at line [501](#) of file [OS.h](#).

2.19.2.77 KHAOS_OS_OpenBSD6_8

```
#define KHAOS_OS_OpenBSD6_8 0
```

Definition at line [504](#) of file [OS.h](#).

2.19.2.78 KHAOS_OS_OpenBSD6_9

```
#define KHAOS_OS_OpenBSD6_9 0
```

Definition at line 507 of file [OS.h](#).

2.19.2.79 KHAOS_OS_OS400

```
#define KHAOS_OS_OS400 0
```

Definition at line 531 of file [OS.h](#).

2.19.2.80 KHAOS_OS_QNX

```
#define KHAOS_OS_QNX 0
```

Definition at line 534 of file [OS.h](#).

2.19.2.81 KHAOS_OS_QNX4

```
#define KHAOS_OS_QNX4 0
```

Definition at line 537 of file [OS.h](#).

2.19.2.82 KHAOS_OS_Solaris

```
#define KHAOS_OS_Solaris 0
```

Definition at line 540 of file [OS.h](#).

2.19.2.83 KHAOS_OS_SVR4

```
#define KHAOS_OS_SVR4 0
```

Definition at line 564 of file [OS.h](#).

2.19.2.84 KHAOS_OS_UNIX

```
#define KHAOS_OS_UNIX 0
```

Definition at line 561 of file [OS.h](#).

2.19.2.85 KHAOS_OS_VERSION

```
#define KHAOS_OS_VERSION KHAOS_SET_VERSION(0, 0, 0)
```

Definition at line 568 of file [OS.h](#).

2.19.2.86 KHAOS_OS_VERSION_TWEAK

```
#define KHAOS_OS_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)
```

Definition at line 572 of file [OS.h](#).

2.19.2.87 KHAOS_OS_VMS

```
#define KHAOS_OS_VMS 0
```

Definition at line 543 of file [OS.h](#).

2.19.2.88 KHAOS_OS_Windows

```
#define KHAOS_OS_Windows 0
```

Definition at line 546 of file [OS.h](#).

2.19.2.89 OS_IS

```
#define OS_IS(  
    os )
```

Detect the operating system.

Parameters

in	os	The os name.
----	----	--------------

Note

Use this macro only if you need preprocessor operating system checking. For example :

```
#if OS_IS(Linux)
//...
#else
//...
#endif
```

Values for os

Macro
AIX
AIX4p3
AIX4p1
AIX3p2
AIX3
AmigaOS
BEOS
BSD
BSD4p4
BSD4p3
BSD4p2
BSDi
DragonFly
FreeBSD
FreeBSD4p10
FreeBSD4p11
NetBSD
NetBSD0p8
NetBSD0p9
NetBSD1
OpenBSD
OpenBSD2 to OpenBSD6p9
CYGWIN
HAIKU
HPUX
iOS
IRIX
Linux
MacOS
OS400
QNX
QNX4
Solaris
SVR4
UNIX
VMS
Windows

Definition at line 69 of file [OS.h](#).

2.20 OS.h

[Go to the documentation of this file.](#)

```

00001 #ifndef KHAOS_OS_H
00002 #define KHAOS_OS_H
00003
00004 #include "VersionMacros.h"
00005
00069 #define OS_IS(os) (KHAOS_OS_#os == 1)
00070
00071 #if defined(_AIX) || defined(__TOS_AIX__)
00072     #define KHAOS_OS_AIX 1
00073     #if defined(_AIX43)
00074         #define KHAOS_OS_VERSION KHAOS_SET_VERSION(4, 3, 0)
00075         #define KHAOS_OS_AIX4p3 1
00076     #elif defined(_AIX41)
00077         #define KHAOS_OS_VERSION KHAOS_SET_VERSION(4, 1, 0)
00078         #define KHAOS_OS_AIX4p1 1
00079     #elif defined(_AIX32)
00080         #define KHAOS_OS_VERSION KHAOS_SET_VERSION(3, 2, 0)
00081         #define KHAOS_OS_AIX3p2 1
00082     #elif defined(_AIX3)
00083         #define KHAOS_OS_AIX3 1
00084         #define KHAOS_OS_VERSION KHAOS_SET_VERSION(3, 0, 0)
00085     #endif
00086 #elif defined(AMIGA) || defined(__amigaos__)
00087     #define KHAOS_OS_AmigaOS 1
00088 #elif defined(__BEOS__)
00089     #define KHAOS_OS_BEOS 1
00090 #elif defined(__bsdi__)
00091     #define KHAOS_OS_BSDi 1
00092 #elif defined(__DragonFly__)
00093     #define KHAOS_OS_DragonFly 1
00094 #elif defined(__FreeBSD__)
00095     #define KHAOS_OS_FreeBSD 1
00096     #include <sys/param.h>
00097     #if defined(__FreeBSD_version)
00098         #if __FreeBSD_version == 491000
00099             #define KHAOS_OS_VERSION KHAOS_SET_VERSION(4, 10, 0)
00100             #define KHAOS_OS_FreeBSD4p10 1
00101         #elif __FreeBSD_version == 492000
00102             #define KHAOS_OS_VERSION KHAOS_SET_VERSION(4, 11, 0)
00103             #define KHAOS_OS_FreeBSD4p11 1
00104         #elif __FreeBSD_version < 500000
00105             #define KHAOS_OS_VERSION KHAOS_SET_VERSION((__FreeBSD_version / 100000) % 10, (__FreeBSD_version
00106 / 10000) % 10, __FreeBSD_version % 10000)
00106         #else
00107             #define KHAOS_OS_VERSION KHAOS_SET_VERSION((__FreeBSD_version / 100000) % 100,
00107 (__FreeBSD_version / 1000) % 100, __FreeBSD_version % 1000)
00108         #endif
00109     #endif
00110 #elif defined(__NETBSD__) || defined(__NetBSD__)
00111     #define KHAOS_OS_NetBSD 1
00112     #if defined(__NETBSD__)
00113         #if defined(__NETBSD_version)
00114             #if __NETBSD_version < 500000
00115                 #define KHAOS_OS_VERSION KHAOS_SET_VERSION((__NETBSD_version / 100000) % 10, (__NETBSD_version
00115 / 10000) % 10, (__NETBSD_version / 1000) % 10)
00116             #else
00117                 #define KHAOS_OS_VERSION KHAOS_SET_VERSION((__NETBSD_version / 100000) % 10, (__NETBSD_version
00117 / 100) % 100, 0)
00118             #endif
00119         #elif defined(__NetBSD__)
00120             #if defined(NetBSD0_8)
00121                 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(0, 8, 0)
00122                 #define KHAOS_OS_NetBSD0p8 1
00123             #elif defined(NetBSD0_9)
00124                 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(0, 9, 0)
00125                 #define KHAOS_OS_NetBSD0p9 1
00126             #elif defined(NetBSD1_0)
00127                 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(1, 0, 0)
00128                 #define KHAOS_OS_NetBSD1 1
00129             #elif defined(__NetBSD_Version)
00130                 #define KHAOS_OS_VERSION KHAOS_SET_VERSION((__NetBSD_Version / 10000000) % 100,
00130 (__NetBSD_Version / 1000000) % 100, (__NetBSD_Version / 100) % 100)
00131             #endif
00132         #endif
00133     #endif
00134 #elif defined(__OpenBSD__)

```



```
00135 #define KHAOS_OS_OpenBSD 1
00136 #if defined(OpenBSD2_0)
00137 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(2, 0, 0)
00138 #elif defined(OpenBSD2_1)
00139 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(2, 1, 0)
00140 #elif defined(OpenBSD2_2)
00141 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(2, 2, 0)
00142 #elif defined(OpenBSD2_3)
00143 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(2, 3, 0)
00144 #elif defined(OpenBSD2_4)
00145 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(2, 4, 0)
00146 #elif defined(OpenBSD2_5)
00147 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(2, 5, 0)
00148 #elif defined(OpenBSD2_6)
00149 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(2, 6, 0)
00150 #elif defined(OpenBSD2_7)
00151 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(2, 7, 0)
00152 #elif defined(OpenBSD2_8)
00153 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(2, 8, 0)
00154 #elif defined(OpenBSD2_9)
00155 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(2, 9, 0)
00156 #elif defined(OpenBSD3_0)
00157 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(3, 0, 0)
00158 #elif defined(OpenBSD3_1)
00159 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(3, 1, 0)
00160 #elif defined(OpenBSD3_2)
00161 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(3, 2, 0)
00162 #elif defined(OpenBSD3_3)
00163 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(3, 3, 0)
00164 #elif defined(OpenBSD3_4)
00165 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(3, 4, 0)
00166 #elif defined(OpenBSD3_5)
00167 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(3, 5, 0)
00168 #elif defined(OpenBSD3_6)
00169 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(3, 6, 0)
00170 #elif defined(OpenBSD3_7)
00171 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(3, 7, 0)
00172 #elif defined(OpenBSD3_8)
00173 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(3, 8, 0)
00174 #elif defined(OpenBSD3_9)
00175 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(3, 9, 0)
00176 #elif defined(OpenBSD4_0)
00177 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(4, 0, 0)
00178 #elif defined(OpenBSD4_1)
00179 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(4, 1, 0)
00180 #elif defined(OpenBSD4_2)
00181 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(4, 2, 0)
00182 #elif defined(OpenBSD4_3)
00183 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(4, 3, 0)
00184 #elif defined(OpenBSD4_4)
00185 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(4, 4, 0)
00186 #elif defined(OpenBSD4_5)
00187 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(4, 5, 0)
00188 #elif defined(OpenBSD4_6)
00189 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(4, 6, 0)
00190 #elif defined(OpenBSD4_7)
00191 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(4, 7, 0)
00192 #elif defined(OpenBSD4_8)
00193 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(4, 8, 0)
00194 #elif defined(OpenBSD4_9)
00195 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(4, 9, 0)
00196 #elif defined(OpenBSD5_0)
00197 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(5, 0, 0)
00198 #elif defined(OpenBSD5_1)
00199 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(5, 1, 0)
00200 #elif defined(OpenBSD5_2)
00201 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(5, 2, 0)
00202 #elif defined(OpenBSD5_3)
00203 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(5, 3, 0)
00204 #elif defined(OpenBSD5_4)
00205 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(5, 4, 0)
00206 #elif defined(OpenBSD5_5)
00207 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(5, 5, 0)
00208 #elif defined(OpenBSD5_6)
00209 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(5, 6, 0)
00210 #elif defined(OpenBSD5_7)
00211 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(5, 7, 0)
00212 #elif defined(OpenBSD5_8)
00213 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(5, 8, 0)
00214 #elif defined(OpenBSD5_9)
00215 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(5, 9, 0)
00216 #elif defined(OpenBSD6_0)
00217 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(6, 0, 0)
00218 #elif defined(OpenBSD6_1)
00219 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(6, 1, 0)
00220 #elif defined(OpenBSD6_2)
00221 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(6, 2, 0)
```

```

00222 #elif defined(OpenBSD6_3)
00223 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(6, 3, 0)
00224 #elif defined(OpenBSD6_4)
00225 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(6, 4, 0)
00226 #elif defined(OpenBSD6_5)
00227 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(6, 5, 0)
00228 #elif defined(OpenBSD6_6)
00229 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(6, 6, 0)
00230 #elif defined(OpenBSD6_7)
00231 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(6, 7, 0)
00232 #elif defined(OpenBSD6_8)
00233 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(6, 8, 0)
00234 #elif defined(OpenBSD6_9)
00235 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(6, 9, 0)
00236 #endif
00237 #elif defined(__CYGWIN__)
00238 #include <cygwin/version.h>
00239 #define KHAOS_OS_CYGWIN 1
00240 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(CYGWIN_VERSION_API_MAJOR, CYGWIN_VERSION_API_MINOR, 0)
00241 #elif defined(__HAIKU__)
00242 #define KHAOS_OS_HAIKU 1
00243 #elif defined(hpux) || defined(_hpux) || defined(__hpux)
00244 #define KHAOS_OS_HPUX 1
00245 #elif defined(__APPLE__) && defined(__MACH__) &&
defined(__ENVIRONMENT_IPHONE_OS_VERSION_MIN_REQUIRED__)
00246 #define KHAOS_OS_iOS 1
00247 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(__ENVIRONMENT_IPHONE_OS_VERSION_MIN_REQUIRED__ * 1000, 0,
0)
00248 #elif defined(sgi) || defined(__sgi)
00249 #define KHAOS_OS_IRIX 1
00250 #elif defined(linux) || defined(__linux) || defined(_linux) || defined(__gnu_linux__)
00251 #define KHAOS_OS_Linux 1
00252 #elif defined(macintosh) || defined(Macintosh) || (defined(__APPLE__) && defined(__MACH__))
00253 #define KHAOS_OS_MacOS 1
00254 #if defined(__APPLE__) && defined(__MACH__)
00255 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(10, 0, 0)
00256 #else
00257 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(9, 0, 0)
00258 #endif
00259 #elif defined(__OS400__)
00260 #define KHAOS_OS_OS400 1
00261 #elif defined(__QNX__) || defined(__QNXNTO__)
00262 #define KHAOS_OS_QNX 1
00263 #if defined(_NTO_VERSION)
00264 #define KHAOS_OS_VERSION KHAOS_SET_VERSION((_NTO_VERSION / 100) % 100, _NTO_VERSION % 100, 0)
00265 #elif defined(__QNX__)
00266 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(4, 0, 0)
00267 #define KHAOS_OS_QNX4 1
00268 #endif
00269 #elif defined(sun) || defined(__sun)
00270 #define KHAOS_OS_Solaris 1
00271 #elif defined(VMS) || defined(__VMS)
00272 #define KHAOS_OS_VMS 1
00273 #if defined(__VMS_VER)
00274 #define KHAOS_OS_VERSION KHAOS_SET_VERSION((__VMS_VER / 100000000) % 100, (__VMS_VER / 1000000) %
100, (__VMS_VER / 100) % 100)
00275 #endif
00276 #elif defined(_WIN32) || defined(_WIN64) || defined(__WIN32__) || defined(__TOS_WIN__) ||
defined(__WINDOWS__)
00277 #define KHAOS_OS_Windows 1
00278 #endif
00279
00280 /* Special case for BSD */
00281 #if defined(BSD) || defined(_SYSTYPE_BSD)
00282 #include <sys/param.h>
00283 #define KHAOS_OS_BSD 1
00284 #if defined(BSD4_4)
00285 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(4, 4, 0)
00286 #define KHAOS_OS_BSD4p4 1
00287 #elif defined(BSD4_3)
00288 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(4, 3, 0)
00289 #define KHAOS_OS_BSD4p3 1
00290 #elif defined(BSD4_2)
00291 #define KHAOS_OS_VERSION KHAOS_SET_VERSION(4, 2, 0)
00292 #define KHAOS_OS_BSD4p2 1
00293 #elif defined(BSD)
00294 #define KHAOS_OS_VERSION KHAOS_SET_VERSION((BSD / 100) % 100, BSD % 100, 0)
00295 #endif
00296 #endif
00297
00298 /* Special case UNIX */
00299 #if defined(unix) || defined(__unix) || defined(_XOPEN_SOURCE) || defined(_POSIX_SOURCE) ||
defined(__unix__) || (defined(__APPLE__) && defined(__MACH__))
00300 #define KHAOS_OS_UNIX 1
00301 #endif
00302
00303 /* Special case SVR4 */

```

```
00304 #if defined(__sysv__) || defined(__SVR4) || defined(__svr4__) || defined(_SYSTYPE_SVR4)
00305     #define KHAOS_OS_SVR4 1
00306 #endif
00307
00308 #if !defined(KHAOS_OS_AIX)
00309     #define KHAOS_OS_AIX 0
00310 #endif
00311 #if !defined(KHAOS_OS_AIX4p3)
00312     #define KHAOS_OS_AIX4p3 0
00313 #endif
00314 #if !defined(KHAOS_OS_AIX4p1)
00315     #define KHAOS_OS_AIX4p1 0
00316 #endif
00317 #if !defined(KHAOS_OS_AIX3p2)
00318     #define KHAOS_OS_AIX3p2 0
00319 #endif
00320 #if !defined(KHAOS_OS_AIX3)
00321     #define KHAOS_OS_AIX3 0
00322 #endif
00323 #if !defined(KHAOS_OS_AmigaOS)
00324     #define KHAOS_OS_AmigaOS 0
00325 #endif
00326 #if !defined(KHAOS_OS_BEOS)
00327     #define KHAOS_OS_BEOS 0
00328 #endif
00329 #if !defined(KHAOS_OS_BSDi)
00330     #define KHAOS_OS_BSDi 0
00331 #endif
00332 #if !defined(KHAOS_OS_DragonFly)
00333     #define KHAOS_OS_DragonFly 0
00334 #endif
00335 #if !defined(KHAOS_OS_FreeBSD)
00336     #define KHAOS_OS_FreeBSD 0
00337 #endif
00338 #if !defined(KHAOS_OS_FreeBSD4p10)
00339     #define KHAOS_OS_FreeBSD4p10 0
00340 #endif
00341 #if !defined(KHAOS_OS_FreeBSD4p11)
00342     #define KHAOS_OS_FreeBSD4p11 0
00343 #endif
00344 #if !defined(KHAOS_OS_NetBSD)
00345     #define KHAOS_OS_NetBSD 0
00346 #endif
00347 #if !defined(KHAOS_OS_NetBSD0p8)
00348     #define KHAOS_OS_NetBSD0p8 0
00349 #endif
00350 #if !defined(KHAOS_OS_NetBSD0p9)
00351     #define KHAOS_OS_NetBSD0p9 0
00352 #endif
00353 #if !defined(KHAOS_OS_NetBSD1)
00354     #define KHAOS_OS_NetBSD1 0
00355 #endif
00356 #if !defined(KHAOS_OS_OpenBSD)
00357     #define KHAOS_OS_OpenBSD 0
00358 #endif
00359 #if !defined(KHAOS_OS_OpenBSD2_0)
00360     #define KHAOS_OS_OpenBSD2_0 0
00361 #endif
00362 #if !defined(KHAOS_OS_OpenBSD2_1)
00363     #define KHAOS_OS_OpenBSD2_1 0
00364 #endif
00365 #if !defined(KHAOS_OS_OpenBSD2_2)
00366     #define KHAOS_OS_OpenBSD2_2 0
00367 #endif
00368 #if !defined(KHAOS_OS_OpenBSD2_3)
00369     #define KHAOS_OS_OpenBSD2_3 0
00370 #endif
00371 #if !defined(KHAOS_OS_OpenBSD2_4)
00372     #define KHAOS_OS_OpenBSD2_4 0
00373 #endif
00374 #if !defined(KHAOS_OS_OpenBSD2_5)
00375     #define KHAOS_OS_OpenBSD2_5 0
00376 #endif
00377 #if !defined(KHAOS_OS_OpenBSD2_6)
00378     #define KHAOS_OS_OpenBSD2_6 0
00379 #endif
00380 #if !defined(KHAOS_OS_OpenBSD2_7)
00381     #define KHAOS_OS_OpenBSD2_7 0
00382 #endif
00383 #if !defined(KHAOS_OS_OpenBSD2_8)
00384     #define KHAOS_OS_OpenBSD2_8 0
00385 #endif
00386 #if !defined(KHAOS_OS_OpenBSD2_9)
00387     #define KHAOS_OS_OpenBSD2_9 0
00388 #endif
00389 #if !defined(KHAOS_OS_OpenBSD3_0)
00390     #define KHAOS_OS_OpenBSD3_0 0
```

```
00391 #endif
00392 #if !defined(KHAOS_OS_OpenBSD3_1)
00393     #define KHAOS_OS_OpenBSD3_1 0
00394 #endif
00395 #if !defined(KHAOS_OS_OpenBSD3_2)
00396     #define KHAOS_OS_OpenBSD3_2 0
00397 #endif
00398 #if !defined(KHAOS_OS_OpenBSD3_3)
00399     #define KHAOS_OS_OpenBSD3_3 0
00400 #endif
00401 #if !defined(KHAOS_OS_OpenBSD3_4)
00402     #define KHAOS_OS_OpenBSD3_4 0
00403 #endif
00404 #if !defined(KHAOS_OS_OpenBSD3_5)
00405     #define KHAOS_OS_OpenBSD3_5 0
00406 #endif
00407 #if !defined(KHAOS_OS_OpenBSD3_6)
00408     #define KHAOS_OS_OpenBSD3_6 0
00409 #endif
00410 #if !defined(KHAOS_OS_OpenBSD3_7)
00411     #define KHAOS_OS_OpenBSD3_7 0
00412 #endif
00413 #if !defined(KHAOS_OS_OpenBSD3_8)
00414     #define KHAOS_OS_OpenBSD3_8 0
00415 #endif
00416 #if !defined(KHAOS_OS_OpenBSD3_9)
00417     #define KHAOS_OS_OpenBSD3_9 0
00418 #endif
00419 #if !defined(KHAOS_OS_OpenBSD4_0)
00420     #define KHAOS_OS_OpenBSD4_0 0
00421 #endif
00422 #if !defined(KHAOS_OS_OpenBSD4_1)
00423     #define KHAOS_OS_OpenBSD4_1 0
00424 #endif
00425 #if !defined(KHAOS_OS_OpenBSD4_2)
00426     #define KHAOS_OS_OpenBSD4_2 0
00427 #endif
00428 #if !defined(KHAOS_OS_OpenBSD4_3)
00429     #define KHAOS_OS_OpenBSD4_3 0
00430 #endif
00431 #if !defined(KHAOS_OS_OpenBSD4_4)
00432     #define KHAOS_OS_OpenBSD4_4 0
00433 #endif
00434 #if !defined(KHAOS_OS_OpenBSD4_5)
00435     #define KHAOS_OS_OpenBSD4_5 0
00436 #endif
00437 #if !defined(KHAOS_OS_OpenBSD4_6)
00438     #define KHAOS_OS_OpenBSD4_6 0
00439 #endif
00440 #if !defined(KHAOS_OS_OpenBSD4_7)
00441     #define KHAOS_OS_OpenBSD4_7 0
00442 #endif
00443 #if !defined(KHAOS_OS_OpenBSD4_8)
00444     #define KHAOS_OS_OpenBSD4_8 0
00445 #endif
00446 #if !defined(KHAOS_OS_OpenBSD4_9)
00447     #define KHAOS_OS_OpenBSD4_9 0
00448 #endif
00449 #if !defined(KHAOS_OS_OpenBSD5_0)
00450     #define KHAOS_OS_OpenBSD5_0 0
00451 #endif
00452 #if !defined(KHAOS_OS_OpenBSD5_1)
00453     #define KHAOS_OS_OpenBSD5_1 0
00454 #endif
00455 #if !defined(KHAOS_OS_OpenBSD5_2)
00456     #define KHAOS_OS_OpenBSD5_2 0
00457 #endif
00458 #if !defined(KHAOS_OS_OpenBSD5_3)
00459     #define KHAOS_OS_OpenBSD5_3 0
00460 #endif
00461 #if !defined(KHAOS_OS_OpenBSD5_4)
00462     #define KHAOS_OS_OpenBSD5_4 0
00463 #endif
00464 #if !defined(KHAOS_OS_OpenBSD5_5)
00465     #define KHAOS_OS_OpenBSD5_5 0
00466 #endif
00467 #if !defined(KHAOS_OS_OpenBSD5_6)
00468     #define KHAOS_OS_OpenBSD5_6 0
00469 #endif
00470 #if !defined(KHAOS_OS_OpenBSD5_7)
00471     #define KHAOS_OS_OpenBSD5_7 0
00472 #endif
00473 #if !defined(KHAOS_OS_OpenBSD5_8)
00474     #define KHAOS_OS_OpenBSD5_8 0
00475 #endif
00476 #if !defined(KHAOS_OS_OpenBSD5_9)
00477     #define KHAOS_OS_OpenBSD5_9 0
```

```
00478 #endif
00479 #if !defined(KHAOS_OS_OpenBSD6_0)
00480 #define KHAOS_OS_OpenBSD6_0 0
00481 #endif
00482 #if !defined(KHAOS_OS_OpenBSD6_1)
00483 #define KHAOS_OS_OpenBSD6_1 0
00484 #endif
00485 #if !defined(KHAOS_OS_OpenBSD6_2)
00486 #define KHAOS_OS_OpenBSD6_2 0
00487 #endif
00488 #if !defined(KHAOS_OS_OpenBSD6_3)
00489 #define KHAOS_OS_OpenBSD6_3 0
00490 #endif
00491 #if !defined(KHAOS_OS_OpenBSD6_4)
00492 #define KHAOS_OS_OpenBSD6_4 0
00493 #endif
00494 #if !defined(KHAOS_OS_OpenBSD6_5)
00495 #define KHAOS_OS_OpenBSD6_5 0
00496 #endif
00497 #if !defined(KHAOS_OS_OpenBSD6_6)
00498 #define KHAOS_OS_OpenBSD6_6 0
00499 #endif
00500 #if !defined(KHAOS_OS_OpenBSD6_7)
00501 #define KHAOS_OS_OpenBSD6_7 0
00502 #endif
00503 #if !defined(KHAOS_OS_OpenBSD6_8)
00504 #define KHAOS_OS_OpenBSD6_8 0
00505 #endif
00506 #if !defined(KHAOS_OS_OpenBSD6_9)
00507 #define KHAOS_OS_OpenBSD6_9 0
00508 #endif
00509 #if !defined(KHAOS_OS_CYGWIN)
00510 #define KHAOS_OS_CYGWIN 0
00511 #endif
00512 #if !defined(KHAOS_OS_HAIKU)
00513 #define KHAOS_OS_HAIKU 0
00514 #endif
00515 #if !defined(KHAOS_OS_HPUX)
00516 #define KHAOS_OS_HPUX 0
00517 #endif
00518 #if !defined(KHAOS_OS_iOS)
00519 #define KHAOS_OS_iOS 0
00520 #endif
00521 #if !defined(KHAOS_OS_IRIX)
00522 #define KHAOS_OS_IRIX 0
00523 #endif
00524 #if !defined(KHAOS_OS_Linux)
00525 #define KHAOS_OS_Linux 0
00526 #endif
00527 #if !defined(KHAOS_OS_MacOS)
00528 #define KHAOS_OS_MacOS 0
00529 #endif
00530 #if !defined(KHAOS_OS_OS400)
00531 #define KHAOS_OS_OS400 0
00532 #endif
00533 #if !defined(KHAOS_OS_QNX)
00534 #define KHAOS_OS_QNX 0
00535 #endif
00536 #if !defined(KHAOS_OS_QNX4)
00537 #define KHAOS_OS_QNX4 0
00538 #endif
00539 #if !defined(KHAOS_OS_Solaris)
00540 #define KHAOS_OS_Solaris 0
00541 #endif
00542 #if !defined(KHAOS_OS_VMS)
00543 #define KHAOS_OS_VMS 0
00544 #endif
00545 #if !defined(KHAOS_OS_Windows)
00546 #define KHAOS_OS_Windows 0
00547 #endif
00548 #if !defined(KHAOS_OS_BSD)
00549 #define KHAOS_OS_BSD 0
00550 #endif
00551 #if !defined(KHAOS_OS_BSD4p4)
00552 #define KHAOS_OS_BSD4p4 0
00553 #endif
00554 #if !defined(KHAOS_OS_BSD4p3)
00555 #define KHAOS_OS_BSD4p3 0
00556 #endif
00557 #if !defined(KHAOS_OS_BSD4p2)
00558 #define KHAOS_OS_BSD4p2 0
00559 #endif
00560 #if !defined(KHAOS_OS_UNIX)
00561 #define KHAOS_OS_UNIX 0
00562 #endif
00563 #if !defined(KHAOS_OS_SVR4)
00564 #define KHAOS_OS_SVR4 0
```

```

00565 #endif
00566
00567 #if !defined(KHAOS_OS_VERSION)
00568     #define KHAOS_OS_VERSION KHAOS_SET_VERSION(0, 0, 0)
00569 #endif
00570
00571 #if !defined(KHAOS_OS_VERSION_TWEAK)
00572     #define KHAOS_OS_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)
00573 #endif
00574
00575 #endif

```

2.21 khaos/Platform.h File Reference

Detect the platform.

```

#include "VersionMacros.h"
#include "khaos/OS.h"

```

Macros

- #define [PLATFORM_IS\(platform\)](#)
Detect the platform.
- #define [KHAOS_PLATFORM_Android](#) 0
- #define [KHAOS_PLATFORM_CloudABI](#) 0
- #define [KHAOS_PLATFORM_iOSTarget](#) 0
- #define [KHAOS_PLATFORM_iOSIphone](#) 0
- #define [KHAOS_PLATFORM_iOSSimulator](#) 0
- #define [KHAOS_PLATFORM_MINGW](#) 0
- #define [KHAOS_PLATFORM_MINGW32](#) 0
- #define [KHAOS_PLATFORM_MINGW64](#) 0
- #define [KHAOS_PLATFORM_UWP](#) 0
- #define [KHAOS_PLATFORM_WindowsDesktop](#) 0
- #define [KHAOS_PLATFORM_WindowsPhone](#) 0
- #define [KHAOS_PLATFORM_WindowsServer](#) 0
- #define [KHAOS_PLATFORM_WindowsStore](#) 0
- #define [KHAOS_PLATFORM_WindowsSystem](#) 0
- #define [KHAOS_PLATFORM_MINGWEmulated](#) 0
- #define [KHAOS_PLATFORM_MINGW32Emulated](#) 0
- #define [KHAOS_PLATFORM_MINGW64Emulated](#) 0
- #define [KHAOS_MINGW_VERSION](#) [KHAOS_SET_VERSION](#)(0, 0, 0)
- #define [KHAOS_MINGW_VERSION_TWEAK](#) [KHAOS_SET_VERSION_TWEAK](#)(0)
- #define [KHAOS_PLATFORM_VERSION](#) [KHAOS_SET_VERSION](#)(0, 0, 0)
- #define [KHAOS_PLATFORM_VERSION_TWEAK](#) [KHAOS_SET_VERSION_TWEAK](#)(0)

2.21.1 Detailed Description

Detect the platform.

Definition in file [Platform.h](#).

2.21.2 Macro Definition Documentation

2.21.2.1 KHAOS_MINGW_VERSION

```
#define KHAOS_MINGW_VERSION KHAOS_SET_VERSION(0, 0, 0)
```

Definition at line 182 of file [Platform.h](#).

2.21.2.2 KHAOS_MINGW_VERSION_TWEAK

```
#define KHAOS_MINGW_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)
```

Definition at line 185 of file [Platform.h](#).

2.21.2.3 KHAOS_PLATFORM_Android

```
#define KHAOS_PLATFORM_Android 0
```

Definition at line 130 of file [Platform.h](#).

2.21.2.4 KHAOS_PLATFORM_CloudABI

```
#define KHAOS_PLATFORM_CloudABI 0
```

Definition at line 133 of file [Platform.h](#).

2.21.2.5 KHAOS_PLATFORM_iOSIphone

```
#define KHAOS_PLATFORM_iOSIphone 0
```

Definition at line 139 of file [Platform.h](#).

2.21.2.6 KHAOS_PLATFORM_iOSSimulator

```
#define KHAOS_PLATFORM_iOSSimulator 0
```

Definition at line 142 of file [Platform.h](#).

2.21.2.7 KHAOS_PLATFORM_iOSTarget

```
#define KHAOS_PLATFORM_iOSTarget 0
```

Definition at line 136 of file [Platform.h](#).

2.21.2.8 KHAOS_PLATFORM_MINGW

```
#define KHAOS_PLATFORM_MINGW 0
```

Definition at line 145 of file [Platform.h](#).

2.21.2.9 KHAOS_PLATFORM_MINGW32

```
#define KHAOS_PLATFORM_MINGW32 0
```

Definition at line 148 of file [Platform.h](#).

2.21.2.10 KHAOS_PLATFORM_MINGW32Emulated

```
#define KHAOS_PLATFORM_MINGW32Emulated 0
```

Definition at line 175 of file [Platform.h](#).

2.21.2.11 KHAOS_PLATFORM_MINGW64

```
#define KHAOS_PLATFORM_MINGW64 0
```

Definition at line 151 of file [Platform.h](#).

2.21.2.12 KHAOS_PLATFORM_MINGW64Emulated

```
#define KHAOS_PLATFORM_MINGW64Emulated 0
```

Definition at line 178 of file [Platform.h](#).

2.21.2.13 KHAOS_PLATFORM_MINGWEmulated

```
#define KHAOS_PLATFORM_MINGWEmulated 0
```

Definition at line 172 of file [Platform.h](#).

2.21.2.14 KHAOS_PLATFORM_UWP

```
#define KHAOS_PLATFORM_UWP 0
```

Definition at line 154 of file [Platform.h](#).

2.21.2.15 KHAOS_PLATFORM_VERSION

```
#define KHAOS_PLATFORM_VERSION KHAOS_SET_VERSION(0, 0, 0)
```

Definition at line 189 of file [Platform.h](#).

2.21.2.16 KHAOS_PLATFORM_VERSION_TWEAK

```
#define KHAOS_PLATFORM_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)
```

Definition at line 192 of file [Platform.h](#).

2.21.2.17 KHAOS_PLATFORM_WindowsDesktop

```
#define KHAOS_PLATFORM_WindowsDesktop 0
```

Definition at line 157 of file [Platform.h](#).

2.21.2.18 KHAOS_PLATFORM_WindowsPhone

```
#define KHAOS_PLATFORM_WindowsPhone 0
```

Definition at line 160 of file [Platform.h](#).

2.21.2.19 KHAOS_PLATFORM_WindowsServer

```
#define KHAOS_PLATFORM_WindowsServer 0
```

Definition at line 163 of file [Platform.h](#).

2.21.2.20 KHAOS_PLATFORM_WindowsStore

```
#define KHAOS_PLATFORM_WindowsStore 0
```

Definition at line 166 of file [Platform.h](#).

2.21.2.21 KHAOS_PLATFORM_WindowsSystem

```
#define KHAOS_PLATFORM_WindowsSystem 0
```

Definition at line 169 of file [Platform.h](#).

2.21.2.22 PLATFORM_IS

```
#define PLATFORM_IS(  
    platform )
```

Detect the platform.

Parameters

in	<i>platform</i>	The platform name.
----	-----------------	--------------------

Note

Use this macro only if you need preprocessor platform checking. For example :

```
#if PLATFORM_IS(Android)  
//...  
#else
```

```
//...
#endif
```

Values for platform

Macro	Type
Android	detection
CloudABI	detection
iOSTarget	detection
iOSIphone	detection
iOSSimulator	detection
MINGW	detection
MINGW64	detection
MINGW32	detection
UWP	detection
WindowsDesktop	detection
WindowsPhone	detection
WindowsServer	detection
WindowsStore	detection
WindowsSystem	detection
MINGWEmulated	detection
MINGW64Emulated	detection
MINGW32Emulated	detection

Definition at line 50 of file [Platform.h](#).

2.22 Platform.h

[Go to the documentation of this file.](#)

```
00001 #ifndef KHAOS_PLATFORM_H
00002 #define KHAOS_PLATFORM_H
00003
00004 #include "VersionMacros.h"
00005
00006 #include "khaos/OS.h"
00007
00008 #define PLATFORM_IS(platform) (KHAOS_PLATFORM_##platform == 1)
00009
00010 #if defined(__ANDROID__)
00011     #define KHAOS_PLATFORM_Android 1
00012 #elif defined(__CLOUDABI__)
00013     #define KHAOS_PLATFORM_CloudABI 1
00014 #elif OS_IS(iOS)
00015     #include <TargetConditionals.h>
00016     #if TARGET_OS_SIMULATOR == 1
00017         #define KHAOS_PLATFORM_iOSTarget 1
00018     #elif TARGET_IPHONE_SIMULATOR
00019         #define KHAOS_PLATFORM_iOSIphone 1
00020     #else
00021         #define KHAOS_PLATFORM_iOSSimulator 1
00022     #endif
00023 #elif defined(__MINGW32__) || defined(__MINGW64__)
00024     #define KHAOS_PLATFORM_MINGW 1
00025     #if defined(__MINGW64__)
00026         #define KHAOS_PLATFORM_MINGW64 1
00027     #else
00028         #define KHAOS_PLATFORM_MINGW32 1
00029     #endif
00030     #include <_mingw.h>
00031     #if defined(__MINGW64_VERSION_MAJOR) && defined(__MINGW64_VERSION_MINOR)
00032         #define KHAOS_PLATFORM_VERSION KHAOS_SET_VERSION(__MINGW64_VERSION_MAJOR, __MINGW64_VERSION_MINOR,
00033 0)
00034     #elif defined(__MINGW32_VERSION_MAJOR) && defined(__MINGW32_VERSION_MINOR)
```

```

00076     #define KHAOS_PLATFORM_VERSION KHAOS_SET_VERSION(__MINGW32_VERSION_MAJOR, __MINGW32_VERSION_MINOR,
00077     0)
00077 #endif
00078 #endif
00079
00080 #if OS_IS(Windows)
00081     // MinGW (32-bit), WinCE, and wineg++ don't have a ntverp.h header
00082     #if !defined(__MINGW32__) && !defined(_WIN32_WCE) && !defined(__WINE__)
00083         #include <ntverp.h>
00084     #endif
00085
00086     // 9200 is Windows SDK 8.0 from ntverp.h which introduced family support
00087     #if (VER_PRODUCTBUILD >= 9200) || (defined(__MINGW64__) && __MINGW64_VERSION_MAJOR >= 3)
00088         #include <winapifamily.h> // Windows SDK
00089     #define KHAOS_PLATFORM_UWP 1
00090 #endif
00091
00092     #if ((defined(WINAPI_FAMILY_DESKTOP_APP) && WINAPI_FAMILY == WINAPI_FAMILY_DESKTOP_APP) ||
00093 !KHAOS_PLATFORM_UWP)
00093         #define KHAOS_PLATFORM_WindowsDesktop 1
00094     #endif
00095
00096     #if defined(WINAPI_FAMILY_PHONE_APP) && WINAPI_FAMILY == WINAPI_FAMILY_PHONE_APP
00097         #define KHAOS_PLATFORM_WindowsPhone 1
00098     #endif
00099
00100     #if defined(WINAPI_FAMILY_SERVER) && WINAPI_FAMILY == WINAPI_FAMILY_SERVER
00101         #define KHAOS_PLATFORM_WindowsServer 1
00102     #endif
00103
00104     #if ((defined(WINAPI_FAMILY_PC_APP) && WINAPI_FAMILY == WINAPI_FAMILY_PC_APP) ||
00105 (defined(WINAPI_FAMILY_APP) && WINAPI_FAMILY == WINAPI_FAMILY_APP))
00105         #define KHAOS_PLATFORM_WindowsStore 1
00106     #endif
00107
00108     #if defined(WINAPI_FAMILY_SYSTEM) && WINAPI_FAMILY == WINAPI_FAMILY_SYSTEM
00109         #define KHAOS_PLATFORM_WindowsSystem 1
00110     #endif
00111 #endif
00112
00113 /* Special case for MINGW */
00114 #if defined(__MINGW32__) || defined(__MINGW64__)
00115     #define KHAOS_PLATFORM_MINGWEmulated 1
00116     #if defined(__MINGW64__)
00117         #define KHAOS_PLATFORM_MINGW64Emulated 1
00118     #else
00119         #define KHAOS_PLATFORM_MINGW32Emulated 1
00120     #endif
00121     #include <_mingw.h>
00122     #if defined(__MINGW64_VERSION_MAJOR) && defined(__MINGW64_VERSION_MINOR)
00123         #define KHAOS_MINGW_VERSION KHAOS_SET_VERSION(__MINGW64_VERSION_MAJOR, __MINGW64_VERSION_MINOR, 0)
00124     #elif defined(__MINGW32_VERSION_MAJOR) && defined(__MINGW32_VERSION_MINOR)
00125         #define KHAOS_MINGW_VERSION KHAOS_SET_VERSION(__MINGW32_VERSION_MAJOR, __MINGW32_VERSION_MINOR, 0)
00126     #endif
00127 #endif
00128
00129 #if !defined(KHAOS_PLATFORM_Android)
00130     #define KHAOS_PLATFORM_Android 0
00131 #endif
00132 #if !defined(KHAOS_PLATFORM_CloudABI)
00133     #define KHAOS_PLATFORM_CloudABI 0
00134 #endif
00135 #if !defined(KHAOS_PLATFORM_iOSTarget)
00136     #define KHAOS_PLATFORM_iOSTarget 0
00137 #endif
00138 #if !defined(KHAOS_PLATFORM_iOSIphone)
00139     #define KHAOS_PLATFORM_iOSIphone 0
00140 #endif
00141 #if !defined(KHAOS_PLATFORM_iOSSimulator)
00142     #define KHAOS_PLATFORM_iOSSimulator 0
00143 #endif
00144 #if !defined(KHAOS_PLATFORM_MINGW)
00145     #define KHAOS_PLATFORM_MINGW 0
00146 #endif
00147 #if !defined(KHAOS_PLATFORM_MINGW32)
00148     #define KHAOS_PLATFORM_MINGW32 0
00149 #endif
00150 #if !defined(KHAOS_PLATFORM_MINGW64)
00151     #define KHAOS_PLATFORM_MINGW64 0
00152 #endif
00153 #if !defined(KHAOS_PLATFORM_UWP)
00154     #define KHAOS_PLATFORM_UWP 0
00155 #endif
00156 #if !defined(KHAOS_PLATFORM_WindowsDesktop)
00157     #define KHAOS_PLATFORM_WindowsDesktop 0
00158 #endif
00159 #if !defined(KHAOS_PLATFORM_WindowsPhone)

```

```

00160 #define KHAOS_PLATFORM_WindowsPhone 0
00161 #endif
00162 #if !defined(KHAOS_PLATFORM_WindowsServer)
00163 #define KHAOS_PLATFORM_WindowsServer 0
00164 #endif
00165 #if !defined(KHAOS_PLATFORM_WindowsStore)
00166 #define KHAOS_PLATFORM_WindowsStore 0
00167 #endif
00168 #if !defined(KHAOS_PLATFORM_WindowsSystem)
00169 #define KHAOS_PLATFORM_WindowsSystem 0
00170 #endif
00171 #if !defined(KHAOS_PLATFORM_MINGWEmulated)
00172 #define KHAOS_PLATFORM_MINGWEmulated 0
00173 #endif
00174 #if !defined(KHAOS_PLATFORM_MINGW32Emulated)
00175 #define KHAOS_PLATFORM_MINGW32Emulated 0
00176 #endif
00177 #if !defined(KHAOS_PLATFORM_MINGW64Emulated)
00178 #define KHAOS_PLATFORM_MINGW64Emulated 0
00179 #endif
00180
00181 #if !defined(KHAOS_MINGW_VERSION)
00182 #define KHAOS_MINGW_VERSION KHAOS_SET_VERSION(0, 0, 0)
00183 #endif
00184 #if !defined(KHAOS_MINGW_VERSION_TWEAK)
00185 #define KHAOS_MINGW_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)
00186 #endif
00187
00188 #if !defined(KHAOS_PLATFORM_VERSION)
00189 #define KHAOS_PLATFORM_VERSION KHAOS_SET_VERSION(0, 0, 0)
00190 #endif
00191 #if !defined(KHAOS_PLATFORM_VERSION_TWEAK)
00192 #define KHAOS_PLATFORM_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)
00193 #endif
00194
00195 #endif

```

2.23 khaos/Pragma.h File Reference

Cross-compiler pragma macro.

```

#include "khaos/Compiler.h"
#include "khaos/Languages.h"

```

2.23.1 Detailed Description

Cross-compiler pragma macro.

Definition in file [Pragma.h](#).

2.24 Pragma.h

[Go to the documentation of this file.](#)

```

00001 #ifndef KHAOS_PRAGMA_H
00002 #define KHAOS_PRAGMA_H
00003
00004 #include "khaos/Compiler.h"
00005 #include "khaos/Languages.h"
00006
00011 #if LANGUAGE_IS_GREATER(C99) || COMPILER_IS(ClangEmulates) || COMPILER_IS_GREATER_EQUAL(GCCEmulated,
3, 0, 0) || COMPILER_IS_GREATER_EQUAL(Intel, 13, 0, 0) || COMPILER_IS_GREATER_EQUAL(IAR, 8, 0, 0)
\
00012 || COMPILER_IS_GREATER_EQUAL(Portland, 18, 4, 0) || COMPILER_IS_GREATER_EQUAL(ARM, 4, 1, 0) ||
COMPILER_IS_GREATER_EQUAL(TI, 15, 12, 0) || HEDLEY_TI_ARMCL_VERSION_CHECK(4, 7, 0) ||
HEDLEY_TI_CL430_VERSION_CHECK(2, 0, 1) \

```

```

00013     || HEDLEY_TI_CL2000_VERSION_CHECK(6, 1, 0) || HEDLEY_TI_CL6X_VERSION_CHECK(7, 0, 0) ||
HEDLEY_TI_CL7X_VERSION_CHECK(1, 2, 0) || HEDLEY_TI_CLPRU_VERSION_CHECK(2, 1, 0) ||
00014     COMPILER_IS_GREATER_EQUAL(CrayC, 5, 0, 0) \
|| COMPILER_IS_GREATER_EQUAL(Tiny, 0, 9, 17) || COMPILER_IS_GREATER_EQUAL(Solaris, 8, 0, 0) ||
(COMPILER_IS_GREATER_EQUAL(IBM, 10, 1, 0) && defined(__C99_PRAGMA_OPERATOR))
00015     #define PRAGMA(x) _Pragma(#x)
00016     #if COMPILER_IS(MSVC)
00017         #define PRAGMA(x) __pragma(#x)
00018     #else
00019         #define PRAGMA(x) _Pragma(#x)
00020     #endif
00021
00022 #endif

```

2.25 khaos/Unix.h File Reference

Define the values of the official POSIX,X/Open and LSB standards.

```

#include "khaos/OS.h"
#include "khaos/VersionMacros.h"

```

Macros

- #define [UNIX_STANDARD](#)(standard) (KHAOS_UNIX_STANDARD_##standard == 1)
- #define [KHAOS_UNIX_STANDARD_POSIX2](#) 0
- #define [KHAOS_UNIX_STANDARD_POSIX_1988](#) 0
- #define [KHAOS_UNIX_STANDARD_POSIX_1990](#) 0
- #define [KHAOS_UNIX_STANDARD_POSIX_1993](#) 0
- #define [KHAOS_UNIX_STANDARD_POSIX_1995](#) 0
- #define [KHAOS_UNIX_STANDARD_POSIX_2001](#) 0
- #define [KHAOS_UNIX_STANDARD_POSIX_2008](#) 0
- #define [KHAOS_UNIX_STANDARD_XPG3](#) 0
- #define [KHAOS_UNIX_STANDARD_XPG4](#) 0
- #define [KHAOS_UNIX_STANDARD_SUS](#) 0
- #define [KHAOS_UNIX_STANDARD_SUSv2](#) 0
- #define [KHAOS_UNIX_STANDARD_SUSv3](#) 0
- #define [KHAOS_UNIX_STANDARD_SUSv4](#) 0
- #define [KHAOS_UNIX_STANDARD_LSB](#) 0
- #define [KHAOS_UNIX_LSB_VERSION](#) [KHAOS_SET_VERSION](#)(0, 0, 0)
- #define [KHAOS_UNIX_LSB_VERSION_TWEAK](#) [KHAOS_SET_VERSION_TWEAK](#)(0)

2.25.1 Detailed Description

Define the values of the official POSIX,X/Open and LSB standards.

Define macros that expand to the integer constant of type **long** whose value are the version of the POSIX,X/Open and LSB standards

Definition in file [Unix.h](#).

2.25.2 Macro Definition Documentation

2.25.2.1 KHAOS_UNIX_LSB_VERSION

```
#define KHAOS_UNIX_LSB_VERSION KHAOS_SET_VERSION(0, 0, 0)
```

Definition at line 100 of file [Unix.h](#).

2.25.2.2 KHAOS_UNIX_LSB_VERSION_TWEAK

```
#define KHAOS_UNIX_LSB_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)
```

Definition at line 103 of file [Unix.h](#).

2.25.2.3 KHAOS_UNIX_STANDARD_LSB

```
#define KHAOS_UNIX_STANDARD_LSB 0
```

Definition at line 97 of file [Unix.h](#).

2.25.2.4 KHAOS_UNIX_STANDARD_POSIX2

```
#define KHAOS_UNIX_STANDARD_POSIX2 0
```

Definition at line 58 of file [Unix.h](#).

2.25.2.5 KHAOS_UNIX_STANDARD_POSIX_1988

```
#define KHAOS_UNIX_STANDARD_POSIX_1988 0
```

Definition at line 61 of file [Unix.h](#).

2.25.2.6 KHAOS_UNIX_STANDARD_POSIX_1990

```
#define KHAOS_UNIX_STANDARD_POSIX_1990 0
```

Definition at line 64 of file [Unix.h](#).

2.25.2.7 KHAOS_UNIX_STANDARD_POSIX_1993

```
#define KHAOS_UNIX_STANDARD_POSIX_1993 0
```

Definition at line 67 of file [Unix.h](#).

2.25.2.8 KHAOS_UNIX_STANDARD_POSIX_1995

```
#define KHAOS_UNIX_STANDARD_POSIX_1995 0
```

Definition at line 70 of file [Unix.h](#).

2.25.2.9 KHAOS_UNIX_STANDARD_POSIX_2001

```
#define KHAOS_UNIX_STANDARD_POSIX_2001 0
```

Definition at line 73 of file [Unix.h](#).

2.25.2.10 KHAOS_UNIX_STANDARD_POSIX_2008

```
#define KHAOS_UNIX_STANDARD_POSIX_2008 0
```

Definition at line 76 of file [Unix.h](#).

2.25.2.11 KHAOS_UNIX_STANDARD_SUS

```
#define KHAOS_UNIX_STANDARD_SUS 0
```

Definition at line 85 of file [Unix.h](#).

2.25.2.12 KHAOS_UNIX_STANDARD_SUSv2

```
#define KHAOS_UNIX_STANDARD_SUSv2 0
```

Definition at line 88 of file [Unix.h](#).

2.25.2.13 KHAOS_UNIX_STANDARD_SUSv3

```
#define KHAOS_UNIX_STANDARD_SUSv3 0
```

Definition at line 91 of file [Unix.h](#).

2.25.2.14 KHAOS_UNIX_STANDARD_SUSv4

```
#define KHAOS_UNIX_STANDARD_SUSv4 0
```

Definition at line 94 of file [Unix.h](#).

2.25.2.15 KHAOS_UNIX_STANDARD_XPG3

```
#define KHAOS_UNIX_STANDARD_XPG3 0
```

Definition at line 79 of file [Unix.h](#).

2.25.2.16 KHAOS_UNIX_STANDARD_XPG4

```
#define KHAOS_UNIX_STANDARD_XPG4 0
```

Definition at line 82 of file [Unix.h](#).

2.25.2.17 UNIX_STANDARD

```
#define UNIX_STANDARD(  
    standard ) (KHAOS_UNIX_STANDARD_##standard == 1)
```

Definition at line 14 of file [Unix.h](#).

2.26 Unix.h

[Go to the documentation of this file.](#)

```

00001 #ifndef KHAOS_UNIX_H
00002 #define KHAOS_UNIX_H
00003
00004 #include "khaos/OS.h"
00005 #include "khaos/VersionMacros.h"
00006
00014 #define UNIX_STANDARD(standard) (KHAOS_UNIX_STANDARD_##standard == 1)
00015
00016 #if OS_IS(UNIX)
00017     #include <unistd.h>
00018     #if defined(_POSIX2_C_VERSION)
00019         #define KHAOS_UNIX_STANDARD_POSIX2 1
00020     #elif defined(_POSIX_VERSION)
00021         #if _POSIX_VERSION == 198808L
00022             #define KHAOS_UNIX_STANDARD_POSIX_1988 1
00023         #elif _POSIX_VERSION == 199009L
00024             #define KHAOS_UNIX_STANDARD_POSIX_1990 1
00025         #elif _POSIX_VERSION == 199309L
00026             #define KHAOS_UNIX_STANDARD_POSIX_1993 1
00027         #elif _POSIX_VERSION == 199506L
00028             #define KHAOS_UNIX_STANDARD_POSIX_1995 1
00029         #elif _POSIX_VERSION == 200112L
00030             #define KHAOS_UNIX_STANDARD_POSIX_2001 1
00031         #elif _POSIX_VERSION == 200809L
00032             #define KHAOS_UNIX_STANDARD_POSIX_2008 1
00033         #endif
00034     #endif
00035     #if defined(_XOPEN_VERSION)
00036         #if _XOPEN_VERSION == 3
00037             #define KHAOS_UNIX_STANDARD_XPG3 1
00038         #elif _XOPEN_VERSION == 4
00039             #define KHAOS_UNIX_STANDARD_XPG4 1
00040         #elif _XOPEN_VERSION == 4 && defined(_XOPEN_UNIX)
00041             #define KHAOS_UNIX_STANDARD_SUS 1
00042         #elif _XOPEN_VERSION == 500
00043             #define KHAOS_UNIX_STANDARD_SUSv2 1
00044         #elif _XOPEN_VERSION == 600
00045             #define KHAOS_UNIX_STANDARD_SUSv3 1
00046         #elif _XOPEN_VERSION == 700
00047             #define KHAOS_UNIX_STANDARD_SUSv4 1
00048         #endif
00049     #endif
00050
00051     #if defined(__LSB_VERSION__)
00052         #define KHAOS_UNIX_STANDARD_LSB 1
00053         #define KHAOS_UNIX_LSB_VERSION KHAOS_SET_VERSION((__LSB_VERSION__ / 10) % 10, __LSB_VERSION__ %
10, 0)
00054     #endif
00055 #endif
00056
00057 #if !defined(KHAOS_UNIX_STANDARD_POSIX2)
00058     #define KHAOS_UNIX_STANDARD_POSIX2 0
00059 #endif
00060 #if !defined(KHAOS_UNIX_STANDARD_POSIX_1988)
00061     #define KHAOS_UNIX_STANDARD_POSIX_1988 0
00062 #endif
00063 #if !defined(KHAOS_UNIX_STANDARD_POSIX_1990)
00064     #define KHAOS_UNIX_STANDARD_POSIX_1990 0
00065 #endif
00066 #if !defined(KHAOS_UNIX_STANDARD_POSIX_1993)
00067     #define KHAOS_UNIX_STANDARD_POSIX_1993 0
00068 #endif
00069 #if !defined(KHAOS_UNIX_STANDARD_POSIX_1995)
00070     #define KHAOS_UNIX_STANDARD_POSIX_1995 0
00071 #endif
00072 #if !defined(KHAOS_UNIX_STANDARD_POSIX_2001)
00073     #define KHAOS_UNIX_STANDARD_POSIX_2001 0
00074 #endif
00075 #if !defined(KHAOS_UNIX_STANDARD_POSIX_2008)
00076     #define KHAOS_UNIX_STANDARD_POSIX_2008 0
00077 #endif
00078 #if !defined(KHAOS_UNIX_STANDARD_XPG3)
00079     #define KHAOS_UNIX_STANDARD_XPG3 0
00080 #endif
00081 #if !defined(KHAOS_UNIX_STANDARD_XPG4)
00082     #define KHAOS_UNIX_STANDARD_XPG4 0
00083 #endif
00084 #if !defined(KHAOS_UNIX_STANDARD_SUS)
00085     #define KHAOS_UNIX_STANDARD_SUS 0
00086 #endif
00087 #if !defined(KHAOS_UNIX_STANDARD_SUSv2)
00088     #define KHAOS_UNIX_STANDARD_SUSv2 0

```

```

00089 #endif
00090 #if !defined(KHAOS_UNIX_STANDARD_SUSv3)
00091 #define KHAOS_UNIX_STANDARD_SUSv3 0
00092 #endif
00093 #if !defined(KHAOS_UNIX_STANDARD_SUSv4)
00094 #define KHAOS_UNIX_STANDARD_SUSv4 0
00095 #endif
00096 #if !defined(KHAOS_UNIX_STANDARD_LSB)
00097 #define KHAOS_UNIX_STANDARD_LSB 0
00098 #endif
00099 #if !defined(KHAOS_UNIX_LSB_VERSION)
00100 #define KHAOS_UNIX_LSB_VERSION KHAOS_SET_VERSION(0, 0, 0)
00101 #endif
00102 #if !defined(KHAOS_UNIX_LSB_VERSION_TWEAK)
00103 #define KHAOS_UNIX_LSB_VERSION_TWEAK KHAOS_SET_VERSION_TWEAK(0)
00104 #endif
00105
00106 #endif

```

2.27 khaos/VersionMacros.h File Reference

Define macros to set, get and compare versions.

Macros

- #define [KHAOS_SET_VERSION](#)(major, minor, patch)
Set standard version numbers.
- #define [KHAOS_SET_VERSION_TWEAK](#)(tweak)
Set standard tweak (build) numbers.
- #define [KHAOS_GET_VERSION_MAJOR](#)(version)
Get standard major version numbers.
- #define [KHAOS_GET_VERSION_MINOR](#)(version)
Get standard minor version numbers.
- #define [KHAOS_GET_VERSION_PATCH](#)(version)
Get standard patch version numbers.
- #define [KHAOS_GET_VERSION_TWEAK](#)(version)
Get standard tweak version numbers.

2.27.1 Detailed Description

Define macros to set, get and compare versions.

Note

Use these macros only if you need preprocessor version checking.

For example :

```

#if KHAOS_GET_VERSION_MAJOR(version1) == KHAOS_GET_VERSION_MAJOR(version2)
//...
#else
//...
#endif

```

Definition in file [VersionMacros.h](#).

2.27.2 Macro Definition Documentation

2.27.2.1 KHAOS_GET_VERSION_MAJOR

```
#define KHAOS_GET_VERSION_MAJOR(  
    version )
```

Get standard major version numbers.

Parameters

<i>in</i>	<i>version</i>	The standard version number.
-----------	----------------	------------------------------

Note

Value can be directly used in both preprocessor and compiler expressions for comparison to other similarly defined values.

Definition at line 49 of file [VersionMacros.h](#).

2.27.2.2 KHAOS_GET_VERSION_MINOR

```
#define KHAOS_GET_VERSION_MINOR(  
    version )
```

Get standard minor version numbers.

Parameters

<i>in</i>	<i>version</i>	The standard version number.
-----------	----------------	------------------------------

Note

Value can be directly used in both preprocessor and compiler expressions for comparison to other similarly defined values.

Definition at line 56 of file [VersionMacros.h](#).

2.27.2.3 KHAOS_GET_VERSION_PATCH

```
#define KHAOS_GET_VERSION_PATCH(  
    version )
```

Get standard patch version numbers.

Parameters

<i>in</i>	<i>version</i>	The standard version number.
-----------	----------------	------------------------------

Note

Value can be directly used in both preprocessor and compiler expressions for comparison to other similarly defined values.

Definition at line 63 of file [VersionMacros.h](#).

2.27.2.4 KHAOS_GET_VERSION_TWEAK

```
#define KHAOS_GET_VERSION_TWEAK(  
    version )
```

Get standard tweak version numbers.

Parameters

<i>in</i>	<i>version</i>	The standard version number.
-----------	----------------	------------------------------

Note

Value can be directly used in both preprocessor and compiler expressions for comparison to other similarly defined values.

Definition at line 70 of file [VersionMacros.h](#).

2.27.2.5 KHAOS_SET_VERSION

```
#define KHAOS_SET_VERSION(  
    major,  
    minor,  
    patch )
```

Set standard version numbers.

Parameters

<i>in</i>	<i>major</i>	The major number can be in the [0,99] range.
<i>in</i>	<i>minor</i>	The minor number can be in the [0,99] range.
<i>in</i>	<i>patch</i>	The patch number can be in the [0,99999] range.

Remarks

Decimal base whole numbers in the range [0,1000000000).

The number range is designed to allow for a (2,2,5) triplet; Which fits within a 32 bit value.

Note

Value can be directly used in both preprocessor and compiler expressions for comparison to other similarly defined values.

Values can be specified in any base. As the defined value is an constant expression.

The implementation enforces the individual ranges for the major, minor, and patch numbers.

Warning

Values over the ranges are truncated (modulo).

Definition at line 31 of file [VersionMacros.h](#).

2.27.2.6 KHAOS_SET_VERSION_TWEAK

```
#define KHAOS_SET_VERSION_TWEAK(  
    tweak )
```

Set standard tweak (build) numbers.

Parameters

in	<i>tweak</i>	The tweak number can be in the [0,9999] range.
----	--------------	--

Note

Value can be directly used in both preprocessor and compiler expressions for comparison to other similarly defined values.

Values can be specified in any base. As the defined value is an constant expression.

The implementation enforces the individual ranges for the major, minor, and patch numbers.

Warning

Values over the ranges are truncated (modulo).

Definition at line 42 of file [VersionMacros.h](#).

2.28 VersionMacros.h

[Go to the documentation of this file.](#)

```
00001 #ifndef KHAOS_VERSION_MACROS_H
00002 #define KHAOS_VERSION_MACROS_H
00003
00031 #define KHAOS_SET_VERSION(major, minor, patch) (((major) % 100) * 10000000) + (((minor) % 100) *
        100000) + ((patch) % 100000)
00032
00042 #define KHAOS_SET_VERSION_TWEAK(tweak) tweak % 100000
00043
00049 #define KHAOS_GET_VERSION_MAJOR(version) ((version) / 10000000) % 100
00050
00056 #define KHAOS_GET_VERSION_MINOR(version) ((version) / 100000) % 100
00057
00063 #define KHAOS_GET_VERSION_PATCH(version) (version) % 100000
00064
00070 #define KHAOS_GET_VERSION_TWEAK(version) (version##_TWEAK) % 100000
00071
00072 #endif
```

2.29 khaos/Warning.h File Reference

Cross-compiler warning push and pop macros.

```
#include "khaos/Compiler.h"
#include "khaos/Pragma.h"
```

Macros

- #define [WARNING_PUSH](#)
- #define [WARNING_POP](#)
- #define [DISABLE_WARNING](#)(warningName)

2.29.1 Detailed Description

Cross-compiler warning push and pop macros.

Definition in file [Warning.h](#).

2.29.2 Macro Definition Documentation

2.29.2.1 DISABLE_WARNING

```
#define DISABLE_WARNING(  
    warningName )
```

Definition at line 26 of file [Warning.h](#).

2.29.2.2 WARNING_POP

```
#define WARNING_POP
```

Definition at line 25 of file [Warning.h](#).

2.29.2.3 WARNING_PUSH

```
#define WARNING_PUSH
```

Definition at line 24 of file [Warning.h](#).

2.30 Warning.h

[Go to the documentation of this file.](#)

```
00001 #ifndef KHAOS_WARNING_H
00002 #define KHAOS_WARNING_H
00003
00008 #include "khaos/Compiler.h"
00009 #include "khaos/Pragma.h"
00010
00011 #if COMPILER_IS(MSVC)
00012     #define WARNING_PUSH                PRAGMA(warning(push))
00013     #define WARNING_POP                PRAGMA(warning(pop))
00014     #define DISABLE_WARNING(warningNumber) PRAGMA(warning(disable : warningNumber))
00015 #elif COMPILER_IS(ClangEmulated) || COMPILER_IS(LLVMEEmulated)
00016     #define WARNING_PUSH                PRAGMA(clang diagnostic push)
00017     #define WARNING_POP                PRAGMA(clang diagnostic pop)
00018     #define DISABLE_WARNING(warningName) PRAGMA(clang diagnostic ignored #warningName)
00019 #elif COMPILER_IS(GCCEmulated)
00020     #define WARNING_PUSH                PRAGMA(GCC diagnostic push)
00021     #define WARNING_POP                PRAGMA(GCC diagnostic pop)
00022     #define DISABLE_WARNING(warningName) PRAGMA(GCC diagnostic ignored #warningName)
00023 #else
00024     #define WARNING_PUSH
00025     #define WARNING_POP
00026     #define DISABLE_WARNING(warningName)
00027 #endif
00028
00029 #endif
```