EPAXDeveloperGuide

0.01

Generated by Doxygen 1.7.6.1

Fri Feb 7 2014 09:40:40

Contents

1 Namespace Index **1**

1.1 Namespace List 1

2 Class Index **3**

2.1 Class Hierarchy 3

3 Class Index **5**

3.1 Class List 5

4 File Index **7**

4.1 File List 7

5 Namespace Documentation **9**

5.1 c_interface Namespace Reference 9

5.1.1 Function Documentation 9

5.1.1.1 char_range 9

5.1.1.2 error_die 9

5.1.1.3 file_exists 9

5.1.1.4 main 10

5.1.1.5 print_usage 10

5.1.2 Variable Documentation 10

5.1.2.1 objs 10

5.1.2.2 other 10

5.1.2.3 remove 10

5.2 EPAX Namespace Reference 10

5.2.1 Typedef Documentation 14

5.2.1.1 BBL 14

5.2.1.2 BIN . . . 14
5.2.1.3 CFG . . . 14
5.2.1.4 FLOW . . . 14
5.2.1.5 FUNC . . . 14
5.2.1.6 INSN . . . 14
5.2.1.7 LOOP . . . 14
5.2.1.8 SECT . . . 14
5.2.1.9 SYM . . . 14
5.2.2 Enumeration Type Documentation . . . 14
5.2.2.1 BinaryFormat . . . 15
5.2.3 Function Documentation . . . 15
5.2.3.1 BBL_addr . . . 15
5.2.3.2 BBL_countInsn . . . 15
5.2.3.3 BBL_countJumpTargets . . . 15
5.2.3.4 BBL_countSources . . . 16
5.2.3.5 BBL_countTargets . . . 16
5.2.3.6 BBL_fallthroughTarget . . . 16
5.2.3.7 BBL_findInsn . . . 17
5.2.3.8 BBL_firstInsn . . . 17
5.2.3.9 BBL_func . . . 17
5.2.3.10 BBL_hasFallthroughTarget . . . 17
5.2.3.11 BBL_head . . . 18
5.2.3.12 BBL_isHead . . . 18
5.2.3.13 BBL_isLastInsn . . . 18
5.2.3.14 BBL_isTail . . . 19
5.2.3.15 BBL_jumpTargets . . . 19
5.2.3.16 BBL_jumpTargets . . . 19
5.2.3.17 BBL_loop . . . 19
5.2.3.18 BBL_nextInsn . . . 20
5.2.3.19 BBL_size . . . 20
5.2.3.20 BBL_sources . . . 20
5.2.3.21 BBL_sources . . . 20
5.2.3.22 BBL_tail . . . 21
5.2.3.23 BBL_targets . . . 21

5.2.3.24 BBL_targets 21
5.2.3.25 BIN_countFunc 21
5.2.3.26 BIN_create 22
5.2.3.27 BIN_destroy 22
5.2.3.28 BIN_fileSize 22
5.2.3.29 BIN_findFunc 22
5.2.3.30 BIN_firstFunc 23
5.2.3.31 BIN_getName 23
5.2.3.32 BIN_isExecutable 23
5.2.3.33 BIN_isLastFunc 24
5.2.3.34 BIN_nextFunc 24
5.2.3.35 BIN_printStaticFile 24
5.2.3.36 BIN_run 25
5.2.3.37 BIN_run 25
5.2.3.38 CFG_countLoop 25
5.2.3.39 CFG_findLoop 25
5.2.3.40 CFG_firstLoop 26
5.2.3.41 CFG_isLastLoop 26
5.2.3.42 CFG_nextLoop 26
5.2.3.43 compareMemory 27
5.2.3.44 DFS 27
5.2.3.45 findBackEdges 27
5.2.3.46 findDominators 27
5.2.3.47 FUNC_addr 27
5.2.3.48 FUNC_bin 27
5.2.3.49 FUNC_cfg 28
5.2.3.50 FUNC_countBbl 28
5.2.3.51 FUNC_countInsn 28
5.2.3.52 FUNC_countTargets 29
5.2.3.53 FUNC_create 29
5.2.3.54 FUNC_Destroy 29
5.2.3.55 FUNC_destroy 29
5.2.3.56 FUNC_findBbl 30
5.2.3.57 FUNC_findInsn 30

5.2.3.58 FUNC_firstBbl . . . 30
5.2.3.59 FUNC_firstInsn . . . 30
5.2.3.60 FUNC_isLastBbl . . . 31
5.2.3.61 FUNC_isLastInsn . . . 31
5.2.3.62 FUNC_name . . . 31
5.2.3.63 FUNC_nextBbl . . . 32
5.2.3.64 FUNC_nextInsn . . . 32
5.2.3.65 FUNC_print . . . 32
5.2.3.66 FUNC_secName . . . 33
5.2.3.67 FUNC_size . . . 33
5.2.3.68 FUNC_targets . . . 33
5.2.3.69 INSN_addr . . . 34
5.2.3.70 INSN_bbl . . . 34
5.2.3.71 INSN_callTarget . . . 34
5.2.3.72 INSN_condName . . . 34
5.2.3.73 INSN_fallsThrough . . . 35
5.2.3.74 INSN_func . . . 35
5.2.3.75 INSN_isBranch . . . 35
5.2.3.76 INSN_isFpop . . . 36
5.2.3.77 INSN_isMemop . . . 36
5.2.3.78 INSN_loop . . . 36
5.2.3.79 INSN_size . . . 37
5.2.3.80 INSN_sourceDatatypeSizeInBits . . . 37
5.2.3.81 INSN_sourceRegisterSizeInBits . . . 37
5.2.3.82 INSN_string . . . 37
5.2.3.83 INSN_targets . . . 38
5.2.3.84 LOOP_cfg . . . 38
5.2.3.85 LOOP_countBbl . . . 38
5.2.3.86 LOOP_countExits . . . 39
5.2.3.87 LOOP_countInsn . . . 39
5.2.3.88 LOOP_depth . . . 39
5.2.3.89 LOOP_exits . . . 40
5.2.3.90 LOOP_exits . . . 40
5.2.3.91 LOOP_findBbl . . . 40

5.2.3.92 LOOP_findInsn . . . 40
5.2.3.93 LOOP_firstBbl . . . 41
5.2.3.94 LOOP_firstInsn . . . 41
5.2.3.95 LOOP_func . . . 41
5.2.3.96 LOOP_head . . . 41
5.2.3.97 LOOP_index . . . 42
5.2.3.98 LOOP_isInnerLoop . . . 42
5.2.3.99 LOOP_isLastBbl . . . 42
5.2.3.100 LOOP_isLastInsn . . . 43
5.2.3.101 LOOP_nextBbl . . . 43
5.2.3.102 LOOP_nextInsn . . . 43
5.2.3.103 LOOP_parent . . . 44
5.2.3.104 LOOP_size . . . 44
5.2.3.105 LOOP_tail . . . 44
5.3 EPAX::Elf Namespace Reference . . . 45
5.4 EPAX::MachO Namespace Reference . . . 45

6 Class Documentation **47**

6.1 EPAX::BaseBinary Class Reference . . . 47
6.1.1 Detailed Description . . . 48
6.1.2 Constructor & Destructor Documentation . . . 48
6.1.2.1 BaseBinary . . . 48
6.1.2.2 ~BaseBinary . . . 48
6.1.3 Member Function Documentation . . . 49
6.1.3.1 countFunctions . . . 49
6.1.3.2 describe . . . 49
6.1.3.3 emit . . . 49
6.1.3.4 findFunction . . . 49
6.1.3.5 findFunctions . . . 49
6.1.3.6 findSymbols . . . 49
6.1.3.7 functionEndAddress . . . 49
6.1.3.8 getFileSize . . . 50
6.1.3.9 getFirstFunction . . . 50
6.1.3.10 getFormat . . . 50

6.1.3.11 getFormatName . . . 50
6.1.3.12 getID . . . 50
6.1.3.13 getInputFile . . . 50
6.1.3.14 getNextFunction . . . 50
6.1.3.15 getStartAddr . . . 50
6.1.3.16 insideTextRange . . . 50
6.1.3.17 is32Bit . . . 51
6.1.3.18 is64Bit . . . 51
6.1.3.19 isARM . . . 51
6.1.3.20 isExecutable . . . 51
6.1.3.21 isLastFunction . . . 51
6.1.3.22 lazyFunctions . . . 51
6.1.3.23 lazySymbols . . . 51
6.1.3.24 printFunctions . . . 51
6.1.3.25 printSections . . . 51
6.1.3.26 verify . . . 51
6.1.4 Member Data Documentation . . . 52
6.1.4.1 foundfunctions . . . 52
6.1.4.2 foundsymbols . . . 52
6.1.4.3 functions . . . 52
6.1.4.4 inputfile . . . 52
6.1.4.5 strtabs . . . 52
6.1.4.6 symtabs . . . 52
6.2 EPAX::BasicBlock Class Reference . . . 52
6.2.1 Detailed Description . . . 53
6.2.2 Constructor & Destructor Documentation . . . 53
6.2.2.1 BasicBlock . . . 53
6.2.2.2 ~BasicBlock . . . 53
6.2.3 Member Function Documentation . . . 53
6.2.3.1 addInstruction . . . 53
6.2.3.2 addSource . . . 54
6.2.3.3 addTarget . . . 54
6.2.3.4 countInstructions . . . 54
6.2.3.5 countSources . . . 54

6.2.3.6 countTargets . . . 54
6.2.3.7 findInstruction . . . 54
6.2.3.8 getControlFlow . . . 54
6.2.3.9 getFunction . . . 54
6.2.3.10 getInstruction . . . 54
6.2.3.11 getLoop . . . 54
6.2.3.12 getSource . . . 55
6.2.3.13 getTarget . . . 55
6.2.3.14 head . . . 55
6.2.3.15 isFallThrough . . . 55
6.2.3.16 isReachable . . . 55
6.2.3.17 print . . . 55
6.2.3.18 setLoop . . . 55
6.2.3.19 setUnreachable . . . 55
6.2.3.20 tail . . . 55
6.3 EPAX::Binary Class Reference . . . 56
6.3.1 Detailed Description . . . 56
6.3.2 Constructor & Destructor Documentation . . . 57
6.3.2.1 Binary . . . 57
6.3.2.2 Binary . . . 57
6.3.2.3 ∼Binary . . . 57
6.3.3 Member Function Documentation . . . 57
6.3.3.1 countFunctions . . . 57
6.3.3.2 detectFormat . . . 57
6.3.3.3 findFunction . . . 58
6.3.3.4 getFileSize . . . 58
6.3.3.5 getFirstFunction . . . 58
6.3.3.6 getFormat . . . 58
6.3.3.7 getFormatName . . . 58
6.3.3.8 getName . . . 58
6.3.3.9 getNextFunction . . . 59
6.3.3.10 getStartAddr . . . 59
6.3.3.11 isExecutable . . . 59
6.3.3.12 isLastFunction . . . 59

6.3.3.13 printStaticFile . . . 59
6.3.3.14 printStaticFile . . . 59
6.3.3.15 runBasic . . . 60
6.4 EPAX::ControlFlow Class Reference . . . 60
6.4.1 Detailed Description . . . 60
6.4.2 Constructor & Destructor Documentation . . . 61
6.4.2.1 ControlFlow . . . 61
6.4.2.2 ~ControlFlow . . . 61
6.4.3 Member Function Documentation . . . 61
6.4.3.1 countBasicBlocks . . . 61
6.4.3.2 countInstructions . . . 61
6.4.3.3 countLoops . . . 61
6.4.3.4 dot_print . . . 61
6.4.3.5 findBasicBlock . . . 61
6.4.3.6 findInstruction . . . 61
6.4.3.7 findLoop . . . 61
6.4.3.8 getBasicBlock . . . 62
6.4.3.9 getFunction . . . 62
6.4.3.10 getInstruction . . . 62
6.4.3.11 getLoop . . . 62
6.4.3.12 getParentOf . . . 62
6.4.3.13 print . . . 62
6.5 EPAX::DetachedText Class Reference . . . 62
6.5.1 Detailed Description . . . 63
6.5.2 Constructor & Destructor Documentation . . . 63
6.5.2.1 DetachedText . . . 63
6.5.2.2 ~DetachedText . . . 63
6.5.3 Member Function Documentation . . . 63
6.5.3.1 print . . . 63
6.5.4 Member Data Documentation . . . 63
6.5.4.1 instructions . . . 63
6.6 EPAX::dyn_bitset Class Reference . . . 64
6.6.1 Detailed Description . . . 64
6.6.2 Constructor & Destructor Documentation . . . 64

6.6.2.1 dyn_bitset . . . 64
6.6.2.2 ~dyn_bitset . . . 64
6.6.3 Member Function Documentation . . . 64
6.6.3.1 clear . . . 64
6.6.3.2 has . . . 65
6.6.3.3 operator!= . . . 65
6.6.3.4 operator&= . . . 65
6.6.3.5 operator= . . . 65
6.6.3.6 operator== . . . 65
6.6.3.7 operator|= . . . 65
6.6.3.8 print . . . 65
6.6.3.9 set . . . 65
6.6.3.10 set . . . 65
6.6.3.11 size . . . 65
6.6.4 Member Data Documentation . . . 66
6.6.4.1 _elements . . . 66
6.6.4.2 _size . . . 66
6.7 EPAX::Elf::ElfBinary Class Reference . . . 66
6.7.1 Detailed Description . . . 67
6.7.2 Constructor & Destructor Documentation . . . 67
6.7.2.1 ElfBinary . . . 67
6.7.2.2 ~ElfBinary . . . 67
6.7.3 Member Function Documentation . . . 67
6.7.3.1 describe . . . 67
6.7.3.2 emit . . . 67
6.7.3.3 findFunctions . . . 68
6.7.3.4 findSections . . . 68
6.7.3.5 findSegments . . . 68
6.7.3.6 findStringtable . . . 68
6.7.3.7 findSymbols . . . 68
6.7.3.8 functionEndAddress . . . 68
6.7.3.9 getFormat . . . 68
6.7.3.10 getStartAddr . . . 69
6.7.3.11 insideTextRange . . . 69

6.7.3.12 is32Bit . . . 69
6.7.3.13 is64Bit . . . 69
6.7.3.14 isARM . . . 69
6.7.3.15 isExecutable . . . 69
6.7.3.16 printFunctions . . . 69
6.7.3.17 printSections . . . 69
6.7.3.18 vaddrToFile . . . 70
6.7.3.19 verify . . . 70
6.7.4 Member Data Documentation . . . 70
6.7.4.1 fileheader . . . 70
6.7.4.2 foundsections . . . 70
6.7.4.3 foundsegments . . . 70
6.7.4.4 sections . . . 70
6.7.4.5 segments . . . 70
6.8 EPAX::Elf::ElfBinary32 Class Reference . . . 70
6.8.1 Detailed Description . . . 71
6.8.2 Constructor & Destructor Documentation . . . 71
6.8.2.1 ElfBinary32 . . . 71
6.8.2.2 ∼ElfBinary32 . . . 71
6.8.3 Member Function Documentation . . . 71
6.8.3.1 getFormat . . . 71
6.9 EPAX::Elf::ElfBinary64 Class Reference . . . 72
6.9.1 Detailed Description . . . 72
6.9.2 Constructor & Destructor Documentation . . . 72
6.9.2.1 ElfBinary64 . . . 72
6.9.2.2 ∼ElfBinary64 . . . 72
6.9.3 Member Function Documentation . . . 72
6.9.3.1 getFormat . . . 72
6.10 EPAX::Elf::ElfStringTable Class Reference . . . 73
6.10.1 Detailed Description . . . 73
6.10.2 Constructor & Destructor Documentation . . . 73
6.10.2.1 ElfStringTable . . . 73
6.10.2.2 ∼ElfStringTable . . . 73
6.10.3 Member Function Documentation . . . 74

6.10.3.1 getStringAt . . . 74
6.10.3.2 print . . . 74
6.11 EPAX::Elf::ElfSymbol Class Reference . . . 74
6.11.1 Detailed Description . . . 75
6.11.2 Constructor & Destructor Documentation . . . 75
6.11.2.1 ElfSymbol . . . 75
6.11.2.2 ~ElfSymbol . . . 75
6.11.3 Member Function Documentation . . . 75
6.11.3.1 getBinding . . . 75
6.11.3.2 getFunctionAddress . . . 75
6.11.3.3 getNameIndex . . . 75
6.11.3.4 getSection . . . 75
6.11.3.5 getSize . . . 76
6.11.3.6 getType . . . 76
6.11.3.7 getValue . . . 76
6.11.3.8 getVisibility . . . 76
6.11.3.9 isFunction . . . 76
6.11.3.10 isThumbFunction . . . 76
6.11.3.11 print . . . 76
6.11.4 Member Data Documentation . . . 76
6.11.4.1 entry . . . 76
6.12 EPAX::Elf::ElfSymbol32 Class Reference . . . 77
6.12.1 Detailed Description . . . 77
6.12.2 Constructor & Destructor Documentation . . . 77
6.12.2.1 ElfSymbol32 . . . 77
6.12.2.2 ~ElfSymbol32 . . . 77
6.12.3 Member Function Documentation . . . 78
6.12.3.1 getBinding . . . 78
6.12.3.2 getNameIndex . . . 78
6.12.3.3 getSection . . . 78
6.12.3.4 getSize . . . 78
6.12.3.5 getType . . . 78
6.12.3.6 getValue . . . 78
6.12.3.7 getVisibility . . . 78

6.13 EPAX::Elf::ElfSymbol64 Class Reference . . . 79
6.13.1 Detailed Description . . . 79
6.13.2 Constructor & Destructor Documentation . . . 79
6.13.2.1 ElfSymbol64 . . . 79
6.13.2.2 ~ElfSymbol64 . . . 79
6.13.3 Member Function Documentation . . . 80
6.13.3.1 getBinding . . . 80
6.13.3.2 getNameIndex . . . 80
6.13.3.3 getSection . . . 80
6.13.3.4 getSize . . . 80
6.13.3.5 getType . . . 80
6.13.3.6 getValue . . . 80
6.13.3.7 getVisibility . . . 80
6.14 EPAX::Elf::ElfSymbolTable Class Reference . . . 81
6.14.1 Detailed Description . . . 81
6.14.2 Constructor & Destructor Documentation . . . 81
6.14.2.1 ElfSymbolTable . . . 81
6.14.2.2 ~ElfSymbolTable . . . 81
6.14.3 Member Function Documentation . . . 81
6.14.3.1 print . . . 81
6.15 EPAXExport Class Reference . . . 82
6.15.1 Detailed Description . . . 82
6.15.2 Constructor & Destructor Documentation . . . 83
6.15.2.1 EPAXExport . . . 83
6.15.2.2 ~EPAXExport . . . 83
6.15.3 Member Function Documentation . . . 83
6.15.3.1 getClass . . . 83
6.15.4 Member Data Documentation . . . 83
6.15.4.1 expclass . . . 83
6.16 EPAX::FileBase Class Reference . . . 83
6.16.1 Detailed Description . . . 84
6.16.2 Constructor & Destructor Documentation . . . 84
6.16.2.1 FileBase . . . 84
6.16.2.2 ~FileBase . . . 84

6.16.3 Member Function Documentation 84
6.16.3.1 getBinary 84
6.16.3.2 getFileOffset 84
6.16.3.3 getFileSize 84
6.16.3.4 getInputFile 84
6.16.3.5 is32Bit 84
6.16.3.6 setFileSize 85
6.17 EPAX::Elf::FileHeader Class Reference 85
6.17.1 Detailed Description 86
6.17.2 Constructor & Destructor Documentation 86
6.17.2.1 FileHeader 86
6.17.2.2 ∼FileHeader 86
6.17.3 Member Function Documentation 86
6.17.3.1 describe 86
6.17.3.2 describeISA 86
6.17.3.3 getFileType 86
6.17.3.4 getPhdrSize 86
6.17.3.5 getSecTableOffset 87
6.17.3.6 getSectionCount 87
6.17.3.7 getSegmentCount 87
6.17.3.8 getSegTableOffset 87
6.17.3.9 getShdrSize 87
6.17.3.10 getShdrStringIndex 87
6.17.3.11 getStartAddr 87
6.17.3.12 isARM 87
6.17.3.13 verify 87
6.17.4 Member Data Documentation 88
6.17.4.1 entry 88
6.18 EPAX::Elf::FileHeader32 Class Reference 88
6.18.1 Detailed Description 88
6.18.2 Constructor & Destructor Documentation 89
6.18.2.1 FileHeader32 89
6.18.2.2 ∼FileHeader32 89
6.18.3 Member Function Documentation 89

6.18.3.1 describe . . . 89
6.18.3.2 getFileType . . . 89
6.18.3.3 getPhdrSize . . . 89
6.18.3.4 getSecTableOffset . . . 89
6.18.3.5 getSectionCount . . . 89
6.18.3.6 getSegmentCount . . . 89
6.18.3.7 getSegTableOffset . . . 90
6.18.3.8 getShdrSize . . . 90
6.18.3.9 getShdrStringIndex . . . 90
6.18.3.10 getStartAddr . . . 90
6.18.3.11 isARM . . . 90
6.18.3.12 verify . . . 90
6.19 EPAX::Elf::FileHeader64 Class Reference . . . 90
6.19.1 Detailed Description . . . 91
6.19.2 Constructor & Destructor Documentation . . . 91
6.19.2.1 FileHeader64 . . . 91
6.19.2.2 ∼FileHeader64 . . . 91
6.19.3 Member Function Documentation . . . 92
6.19.3.1 describe . . . 92
6.19.3.2 getFileType . . . 92
6.19.3.3 getPhdrSize . . . 92
6.19.3.4 getSecTableOffset . . . 92
6.19.3.5 getSectionCount . . . 92
6.19.3.6 getSegmentCount . . . 92
6.19.3.7 getSegTableOffset . . . 92
6.19.3.8 getShdrSize . . . 92
6.19.3.9 getShdrStringIndex . . . 93
6.19.3.10 getStartAddr . . . 93
6.19.3.11 isARM . . . 93
6.19.3.12 verify . . . 93
6.20 EPAX::Function Class Reference . . . 93
6.20.1 Detailed Description . . . 94
6.20.2 Constructor & Destructor Documentation . . . 94
6.20.2.1 Function . . . 94

6.20.2.2 ~Function . . . 94
6.20.3 Member Function Documentation . . . 94
6.20.3.1 countBasicBlocks . . . 94
6.20.3.2 countInstructions . . . 94
6.20.3.3 disassemble . . . 95
6.20.3.4 findBasicBlock . . . 95
6.20.3.5 findInstruction . . . 95
6.20.3.6 getBasicBlock . . . 95
6.20.3.7 getControlFlow . . . 95
6.20.3.8 getInstruction . . . 95
6.20.3.9 print . . . 95
6.20.3.10 printHeader . . . 95
6.21 EPAX::IndexBase Class Reference . . . 96
6.21.1 Detailed Description . . . 96
6.21.2 Constructor & Destructor Documentation . . . 96
6.21.2.1 IndexBase . . . 96
6.21.2.2 ~IndexBase . . . 97
6.21.3 Member Function Documentation . . . 97
6.21.3.1 getIndex . . . 97
6.21.3.2 setIndex . . . 97
6.22 EPAX::InputFile Class Reference . . . 97
6.22.1 Detailed Description . . . 97
6.22.2 Constructor & Destructor Documentation . . . 98
6.22.2.1 InputFile . . . 98
6.22.2.2 ~InputFile . . . 98
6.22.3 Member Function Documentation . . . 98
6.22.3.1 getBytes . . . 98
6.22.3.2 getFileSize . . . 98
6.23 EPAX::Instruction Class Reference . . . 98
6.23.1 Detailed Description . . . 99
6.23.2 Constructor & Destructor Documentation . . . 99
6.23.2.1 Instruction . . . 99
6.23.2.2 ~Instruction . . . 99
6.23.3 Member Function Documentation . . . 99

6.23.3.1 fallthroughTarget . . . 99
6.23.3.2 getBasicBlock . . . 99
6.23.3.3 getBranchTarget . . . 100
6.23.3.4 getCondition . . . 100
6.23.3.5 getControlTargets . . . 100
6.23.3.6 getSourceDatatypeSizeInBits . . . 100
6.23.3.7 getSourceRegisterSizeInBits . . . 100
6.23.3.8 hasFallthrough . . . 100
6.23.3.9 isBranch . . . 100
6.23.3.10 isCall . . . 100
6.23.3.11 isConditionalBranch . . . 100
6.23.3.12 isFpop . . . 100
6.23.3.13 isLoad . . . 101
6.23.3.14 isMemop . . . 101
6.23.3.15 isStore . . . 101
6.23.3.16 isUnconditionalBranch . . . 101
6.23.3.17 print . . . 101
6.23.3.18 setBasicBlock . . . 101
6.23.3.19 source . . . 101
6.23.3.20 stringRep . . . 101
6.23.3.21 touchesPC . . . 101
6.24 EPAX::Loop Class Reference . . . 102
6.24.1 Detailed Description . . . 102
6.24.2 Constructor & Destructor Documentation . . . 102
6.24.2.1 Loop . . . 102
6.24.2.2 ∼Loop . . . 103
6.24.3 Member Function Documentation . . . 103
6.24.3.1 countBasicBlocks . . . 103
6.24.3.2 countInstructions . . . 103
6.24.3.3 findBasicBlock . . . 103
6.24.3.4 findInstruction . . . 103
6.24.3.5 getBasicBlock . . . 103
6.24.3.6 getControlFlow . . . 103
6.24.3.7 getDepth . . . 103

6.24.3.8 getInstruction . . . 103
6.24.3.9 getNextBasicBlock . . . 103
6.24.3.10 getSize . . . 103
6.24.3.11 hasBasicBlock . . . 104
6.24.3.12 head . . . 104
6.24.3.13 isChildOf . . . 104
6.24.3.14 isLastBasicBlock . . . 104
6.24.3.15 isLastInstruction . . . 104
6.24.3.16 setDepth . . . 104
6.24.3.17 tail . . . 104
6.25 EPAX::MachO::MachHeader Class Reference . . . 104
6.25.1 Detailed Description . . . 105
6.25.2 Constructor & Destructor Documentation . . . 105
6.25.2.1 MachHeader . . . 105
6.25.2.2 ∼MachHeader . . . 105
6.25.3 Member Function Documentation . . . 105
6.25.3.1 describe . . . 105
6.25.3.2 describeISA . . . 105
6.25.3.3 getFileType . . . 106
6.25.3.4 getStartAddr . . . 106
6.25.3.5 isARM . . . 106
6.25.3.6 verify . . . 106
6.25.4 Member Data Documentation . . . 106
6.25.4.1 entry . . . 106
6.26 EPAX::MachO::MachHeader32 Class Reference . . . 106
6.26.1 Detailed Description . . . 107
6.26.2 Constructor & Destructor Documentation . . . 107
6.26.2.1 MachHeader32 . . . 107
6.26.2.2 ∼MachHeader32 . . . 107
6.26.3 Member Function Documentation . . . 107
6.26.3.1 describe . . . 107
6.26.3.2 getFileType . . . 107
6.26.3.3 getStartAddr . . . 107
6.26.3.4 isARM . . . 108

6.26.3.5 verify . . . 108
6.27 EPAX::MachO::MachHeader64 Class Reference . . . 108
6.27.1 Detailed Description . . . 108
6.27.2 Constructor & Destructor Documentation . . . 109
6.27.2.1 MachHeader64 . . . 109
6.27.2.2 ∼MachHeader64 . . . 109
6.27.3 Member Function Documentation . . . 109
6.27.3.1 describe . . . 109
6.27.3.2 getFileType . . . 109
6.27.3.3 getStartAddr . . . 109
6.27.3.4 isARM . . . 109
6.27.3.5 verify . . . 109
6.28 EPAX::MachO::MachOBinary Class Reference . . . 110
6.28.1 Detailed Description . . . 110
6.28.2 Constructor & Destructor Documentation . . . 111
6.28.2.1 MachOBinary . . . 111
6.28.2.2 ∼MachOBinary . . . 111
6.28.3 Member Function Documentation . . . 111
6.28.3.1 describe . . . 111
6.28.3.2 emit . . . 111
6.28.3.3 findFunctions . . . 111
6.28.3.4 findSections . . . 111
6.28.3.5 findSymbols . . . 111
6.28.3.6 functionEndAddress . . . 112
6.28.3.7 getFormat . . . 112
6.28.3.8 getStartAddr . . . 112
6.28.3.9 insideTextRange . . . 112
6.28.3.10 is32Bit . . . 112
6.28.3.11 is64Bit . . . 112
6.28.3.12 isARM . . . 113
6.28.3.13 isExecutable . . . 113
6.28.3.14 printFunctions . . . 113
6.28.3.15 printSections . . . 113
6.28.3.16 verify . . . 113

6.28.4 Member Data Documentation 113
6.28.4.1 machheader 113
6.29 EPAX::MachO::MachOBinary32 Class Reference 113
6.29.1 Detailed Description 114
6.29.2 Constructor & Destructor Documentation 114
6.29.2.1 MachOBinary32 114
6.29.2.2 ∼MachOBinary32 114
6.29.3 Member Function Documentation 114
6.29.3.1 getFormat 114
6.30 EPAX::MachO::MachOBinary64 Class Reference 115
6.30.1 Detailed Description 115
6.30.2 Constructor & Destructor Documentation 115
6.30.2.1 MachOBinary64 115
6.30.2.2 ∼MachOBinary64 115
6.30.3 Member Function Documentation 115
6.30.3.1 getFormat 116
6.31 EPAX::MemoryBase Class Reference 116
6.31.1 Detailed Description 116
6.31.2 Constructor & Destructor Documentation 116
6.31.2.1 MemoryBase 116
6.31.2.2 MemoryBase 116
6.31.2.3 ∼MemoryBase 117
6.31.3 Member Function Documentation 117
6.31.3.1 getMemoryAddress 117
6.31.3.2 getMemorySize 117
6.31.3.3 inRange 117
6.31.3.4 setMemorySize 117
6.32 EPAX::NameBase Class Reference 117
6.32.1 Detailed Description 118
6.32.2 Constructor & Destructor Documentation 118
6.32.2.1 NameBase 118
6.32.2.2 NameBase 118
6.32.2.3 ∼NameBase 118
6.32.3 Member Function Documentation 118

6.32.3.1 getName . . . 118
6.32.3.2 setName . . . 118
6.33 EPAX::Elf::ProgramHeader Class Reference . . . 118
6.33.1 Detailed Description . . . 119
6.33.2 Constructor & Destructor Documentation . . . 119
6.33.2.1 ProgramHeader . . . 119
6.33.2.2 ∼ProgramHeader . . . 119
6.33.3 Member Function Documentation . . . 120
6.33.3.1 getAlignment . . . 120
6.33.3.2 getFlags . . . 120
6.33.3.3 getFOffset . . . 120
6.33.3.4 getFSize . . . 120
6.33.3.5 getMSize . . . 120
6.33.3.6 getPaddr . . . 120
6.33.3.7 getSegmentType . . . 120
6.33.3.8 getVaddr . . . 120
6.33.3.9 isValidVaddr . . . 121
6.33.3.10 vaddrToFileaddr . . . 121
6.33.4 Member Data Documentation . . . 121
6.33.4.1 entry . . . 121
6.34 EPAX::Elf::ProgramHeader32 Class Reference . . . 121
6.34.1 Detailed Description . . . 122
6.34.2 Constructor & Destructor Documentation . . . 122
6.34.2.1 ProgramHeader32 . . . 122
6.34.2.2 ∼ProgramHeader32 . . . 122
6.34.3 Member Function Documentation . . . 122
6.34.3.1 getAlignment . . . 122
6.34.3.2 getFlags . . . 122
6.34.3.3 getFOffset . . . 122
6.34.3.4 getFSize . . . 122
6.34.3.5 getMSize . . . 122
6.34.3.6 getPaddr . . . 123
6.34.3.7 getSegmentType . . . 123
6.34.3.8 getVaddr . . . 123

6.35 EPAX::Elf::ProgramHeader64 Class Reference . . . 123
6.35.1 Detailed Description . . . 124
6.35.2 Constructor & Destructor Documentation . . . 124
6.35.2.1 ProgramHeader64 . . . 124
6.35.2.2 ~ProgramHeader64 . . . 124
6.35.3 Member Function Documentation . . . 124
6.35.3.1 getAlignment . . . 124
6.35.3.2 getFlags . . . 124
6.35.3.3 getFOffset . . . 124
6.35.3.4 getFSize . . . 125
6.35.3.5 getMSize . . . 125
6.35.3.6 getPaddr . . . 125
6.35.3.7 getSegmentType . . . 125
6.35.3.8 getVaddr . . . 125
6.36 EPAX::Section Class Reference . . . 125
6.36.1 Detailed Description . . . 126
6.36.2 Constructor & Destructor Documentation . . . 126
6.36.2.1 Section . . . 126
6.36.2.2 ~Section . . . 126
6.36.3 Member Function Documentation . . . 126
6.36.3.1 isBSS . . . 126
6.36.3.2 isData . . . 126
6.36.3.3 isDebug . . . 126
6.36.3.4 isString . . . 127
6.36.3.5 isSymbol . . . 127
6.36.3.6 isText . . . 127
6.36.3.7 print . . . 127
6.37 EPAX::Elf::SectionHeader Class Reference . . . 127
6.37.1 Detailed Description . . . 128
6.37.2 Constructor & Destructor Documentation . . . 128
6.37.2.1 SectionHeader . . . 128
6.37.2.2 ~SectionHeader . . . 128
6.37.3 Member Function Documentation . . . 129
6.37.3.1 getAlignment . . . 129

6.37.3.2 getEntrySize . . . 129
6.37.3.3 getFileOffset . . . 129
6.37.3.4 getFlags . . . 129
6.37.3.5 getNameIndex . . . 129
6.37.3.6 getSectionLink . . . 129
6.37.3.7 getSize . . . 129
6.37.3.8 getType . . . 129
6.37.3.9 getVirtAddr . . . 130
6.37.3.10 inRange . . . 130
6.37.3.11 isAlloc . . . 130
6.37.3.12 isBSS . . . 130
6.37.3.13 isData . . . 130
6.37.3.14 isDebug . . . 130
6.37.3.15 isExec . . . 130
6.37.3.16 isMerge . . . 130
6.37.3.17 isRead . . . 130
6.37.3.18 isString . . . 130
6.37.3.19 isSymbol . . . 131
6.37.3.20 isText . . . 131
6.37.3.21 isWrite . . . 131
6.37.3.22 print . . . 131
6.37.4 Member Data Documentation . . . 131
6.37.4.1 entry . . . 131
6.38 EPAX::Elf::SectionHeader32 Class Reference . . . 131
6.38.1 Detailed Description . . . 132
6.38.2 Constructor & Destructor Documentation . . . 132
6.38.2.1 SectionHeader32 . . . 132
6.38.2.2 ~SectionHeader32 . . . 132
6.38.3 Member Function Documentation . . . 132
6.38.3.1 getAlignment . . . 132
6.38.3.2 getEntrySize . . . 132
6.38.3.3 getFileOffset . . . 133
6.38.3.4 getFlags . . . 133
6.38.3.5 getNameIndex . . . 133

6.38.3.6 getSectionLink . . . 133
6.38.3.7 getSize . . . 133
6.38.3.8 getType . . . 133
6.38.3.9 getVirtAddr . . . 133
6.39 EPAX::Elf::SectionHeader64 Class Reference . . . 134
6.39.1 Detailed Description . . . 134
6.39.2 Constructor & Destructor Documentation . . . 134
6.39.2.1 SectionHeader64 . . . 134
6.39.2.2 ∼SectionHeader64 . . . 134
6.39.3 Member Function Documentation . . . 135
6.39.3.1 getAlignment . . . 135
6.39.3.2 getEntrySize . . . 135
6.39.3.3 getFileOffset . . . 135
6.39.3.4 getFlags . . . 135
6.39.3.5 getNameIndex . . . 135
6.39.3.6 getSectionLink . . . 135
6.39.3.7 getSize . . . 135
6.39.3.8 getType . . . 135
6.39.3.9 getVirtAddr . . . 136
6.40 EPAX::StringTable Class Reference . . . 136
6.40.1 Detailed Description . . . 136
6.40.2 Constructor & Destructor Documentation . . . 136
6.40.2.1 StringTable . . . 136
6.40.2.2 ∼StringTable . . . 137
6.40.3 Member Function Documentation . . . 137
6.40.3.1 getStringAt . . . 137
6.40.3.2 isString . . . 137
6.41 EPAX::Symbol Class Reference . . . 137
6.41.1 Detailed Description . . . 138
6.41.2 Constructor & Destructor Documentation . . . 138
6.41.2.1 Symbol . . . 138
6.41.2.2 ∼Symbol . . . 138
6.41.3 Member Function Documentation . . . 138
6.41.3.1 isFunction . . . 138

6.41.3.2 isThumbFunction . . . 138
6.42 EPAX::SymbolBase Class Reference . . . 138
6.42.1 Detailed Description . . . 139
6.42.2 Constructor & Destructor Documentation . . . 139
6.42.2.1 SymbolBase . . . 139
6.42.2.2 ∼SymbolBase . . . 139
6.42.3 Member Function Documentation . . . 139
6.42.3.1 getName . . . 139
6.42.3.2 getSymbol . . . 139
6.42.3.3 setSymbol . . . 139
6.43 EPAX::SymbolTable Class Reference . . . 139
6.43.1 Detailed Description . . . 140
6.43.2 Constructor & Destructor Documentation . . . 140
6.43.2.1 SymbolTable . . . 140
6.43.2.2 ∼SymbolTable . . . 140
6.43.3 Member Function Documentation . . . 140
6.43.3.1 countSymbols . . . 140
6.43.3.2 getSymbol . . . 141
6.43.3.3 isSymbol . . . 141
6.43.3.4 print . . . 141
6.43.4 Member Data Documentation . . . 141
6.43.4.1 symbols . . . 141

7 File Documentation **143**

7.1 BaseClass.cpp File Reference . . . 143
7.1.1 Detailed Description . . . 143
7.1.2 LICENSE . . . 143
7.2 BaseClass.hpp File Reference . . . 144
7.2.1 Detailed Description . . . 144
7.2.2 LICENSE . . . 144
7.3 BasicBlock.cpp File Reference . . . 145
7.3.1 Detailed Description . . . 145
7.3.2 LICENSE . . . 145
7.4 BasicBlock.hpp File Reference . . . 145

7.4.1 Detailed Description . . . 145
7.4.2 LICENSE . . . 146
7.5 Binary.cpp File Reference . . . 146
7.5.1 Detailed Description . . . 146
7.5.2 LICENSE . . . 146
7.5.3 Define Documentation . . . 147
7.5.3.1 PTRACE_AND_CHECK . . . 147
7.5.3.2 VERIFY_SINGLE_FORMAT . . . 147
7.6 Binary.hpp File Reference . . . 147
7.6.1 Detailed Description . . . 148
7.6.2 LICENSE . . . 148
7.7 c_interface.py File Reference . . . 148
7.8 ControlFlow.cpp File Reference . . . 148
7.8.1 Detailed Description . . . 149
7.8.2 LICENSE . . . 149
7.9 ControlFlow.hpp File Reference . . . 149
7.9.1 Detailed Description . . . 150
7.9.2 LICENSE . . . 150
7.10 DataStruct.hpp File Reference . . . 150
7.10.1 Detailed Description . . . 150
7.10.2 LICENSE . . . 150
7.10.3 Define Documentation . . . 151
7.10.3.1 __get_index . . . 151
7.10.3.2 __has_bit . . . 151
7.10.3.3 __internal_size . . . 151
7.10.3.4 __set_bit . . . 151
7.11 ElfBinary.cpp File Reference . . . 151
7.11.1 Detailed Description . . . 152
7.11.2 LICENSE . . . 152
7.11.3 Define Documentation . . . 152
7.11.3.1 CASE . . . 152
7.11.3.2 EHDR32_ENTRY . . . 152
7.11.3.3 EHDR64_ENTRY . . . 152
7.11.3.4 PHDR32_ENTRY . . . 153

7.11.3.5 PHDR64_ENTRY . . . 153
7.11.3.6 SHDR32_ENTRY . . . 153
7.11.3.7 SHDR64_ENTRY . . . 153
7.11.3.8 SYM32_ENTRY . . . 153
7.11.3.9 SYM64_ENTRY . . . 153
7.12 ElfBinary.hpp File Reference . . . 153
7.12.1 Detailed Description . . . 154
7.12.2 LICENSE . . . 154
7.13 EPAX.cpp File Reference . . . 154
7.13.1 Detailed Description . . . 154
7.13.2 LICENSE . . . 155
7.13.3 Function Documentation . . . 155
7.13.3.1 error_out . . . 155
7.13.3.2 main . . . 155
7.14 EPAXCommonInternal.hpp File Reference . . . 155
7.14.1 Detailed Description . . . 156
7.14.2 LICENSE . . . 156
7.14.3 Define Documentation . . . 157
7.14.3.1 __do_not_call__ . . . 157
7.14.3.2 ADDRESS_IS_THUMB . . . 157
7.14.3.3 ALIGN_PWR2 . . . 157
7.14.3.4 BACKTRACE_LIMIT . . . 157
7.14.3.5 DEC . . . 157
7.14.3.6 ENDL . . . 157
7.14.3.7 EPAX_PREFACE . . . 157
7.14.3.8 EPAXAssert . . . 157
7.14.3.9 EPAXDie . . . 158
7.14.3.10 EPAXErr . . . 158
7.14.3.11 EPAXOut . . . 158
7.14.3.12 EPAXVerifyType . . . 158
7.14.3.13 EPAXWarn . . . 158
7.14.3.14 HEX . . . 158
7.14.3.15 INVALID_ADDRESS . . . 158
7.14.3.16 INVALID_PTR . . . 158

7.14.3.17 IS_VALID_PTR . . . 158
7.14.3.18 MAX_STRING_SIZE . . . 159
7.14.3.19 NAME_UNKNOWN . . . 159
7.14.3.20 ShouldNotArrive . . . 159
7.14.3.21 TAB . . . 159
7.14.4 Typedef Documentation . . . 159
7.14.4.1 rawbyte_t . . . 159
7.14.5 Enumeration Type Documentation . . . 159
7.14.5.1 EPAXExportClass . . . 159
7.15 Function.cpp File Reference . . . 160
7.15.1 Detailed Description . . . 160
7.15.2 LICENSE . . . 160
7.16 Function.hpp File Reference . . . 160
7.16.1 Detailed Description . . . 161
7.16.2 LICENSE . . . 161
7.17 InputFile.cpp File Reference . . . 161
7.17.1 Detailed Description . . . 161
7.17.2 LICENSE . . . 161
7.18 InputFile.hpp File Reference . . . 162
7.18.1 Detailed Description . . . 162
7.18.2 LICENSE . . . 162
7.19 Instruction.cpp File Reference . . . 162
7.19.1 Detailed Description . . . 163
7.19.2 LICENSE . . . 163
7.19.3 Define Documentation . . . 163
7.19.3.1 DARM_PREDICATE_UNCOND . . . 163
7.19.3.2 DARM_REGLIST_HASREG . . . 163
7.20 Instruction.hpp File Reference . . . 163
7.20.1 Detailed Description . . . 163
7.20.2 LICENSE . . . 164
7.21 Interface.cpp File Reference . . . 164
7.21.1 Detailed Description . . . 169
7.21.2 LICENSE . . . 169
7.21.3 Function Documentation . . . 169

7.21.3.1 EPAX_bbl_addr 169
7.21.3.2 EPAX_bbl_countInsn 169
7.21.3.3 EPAX_bbl_countJumpTargets 169
7.21.3.4 EPAX_bbl_countSources 169
7.21.3.5 EPAX_bbl_countTargets 169
7.21.3.6 EPAX_bbl_fallthroughTarget 169
7.21.3.7 EPAX_bbl_findInsn 170
7.21.3.8 EPAX_bbl_firstInsn 170
7.21.3.9 EPAX_bbl_func 170
7.21.3.10 EPAX_bbl_hasFallthroughTarget 170
7.21.3.11 EPAX_bbl_head 170
7.21.3.12 EPAX_bbl_isHead 170
7.21.3.13 EPAX_bbl_isLastInsn 170
7.21.3.14 EPAX_bbl_isTail 170
7.21.3.15 EPAX_bbl_jumpTargets 170
7.21.3.16 EPAX_bbl_loop 170
7.21.3.17 EPAX_bbl_nextInsn 171
7.21.3.18 EPAX_bbl_size 171
7.21.3.19 EPAX_bbl_sources 171
7.21.3.20 EPAX_bbl_tail 171
7.21.3.21 EPAX_bbl_targets 171
7.21.3.22 EPAX_bin_countFunc 171
7.21.3.23 EPAX_bin_create 171
7.21.3.24 EPAX_bin_destroy 171
7.21.3.25 EPAX_bin_fileSize 171
7.21.3.26 EPAX_bin_firstFunc 171
7.21.3.27 EPAX_bin_fundFunc 172
7.21.3.28 EPAX_bin_getName 172
7.21.3.29 EPAX_bin_isExecutable 172
7.21.3.30 EPAX_bin_isLastFunc 172
7.21.3.31 EPAX_bin_nextFunc 172
7.21.3.32 EPAX_bin_printStaticFile 172
7.21.3.33 EPAX_bin_run 172
7.21.3.34 EPAX_cfg_countLoop 172

7.21.3.35 EPAX_cfg_findLoop 172
7.21.3.36 EPAX_cfg_firstLoop 172
7.21.3.37 EPAX_cfg_isLastLoop 173
7.21.3.38 EPAX_cfg_nextLoop 173
7.21.3.39 EPAX_func_addr 173
7.21.3.40 EPAX_func_bin 173
7.21.3.41 EPAX_func_cfg 173
7.21.3.42 EPAX_func_countBbl 173
7.21.3.43 EPAX_func_countInsn 173
7.21.3.44 EPAX_func_countTargets 173
7.21.3.45 EPAX_func_create 173
7.21.3.46 EPAX_func_destroy 173
7.21.3.47 EPAX_func_findBbl 174
7.21.3.48 EPAX_func_findInsn 174
7.21.3.49 EPAX_func_firstBbl 174
7.21.3.50 EPAX_func_firstInsn 174
7.21.3.51 EPAX_func_isLastBbl 174
7.21.3.52 EPAX_func_isLastInsn 174
7.21.3.53 EPAX_func_name 174
7.21.3.54 EPAX_func_nextBbl 174
7.21.3.55 EPAX_func_nextInsn 174
7.21.3.56 EPAX_func_print 174
7.21.3.57 EPAX_func_secName 175
7.21.3.58 EPAX_func_size 175
7.21.3.59 EPAX_func_targets 175
7.21.3.60 EPAX_insn_addr 175
7.21.3.61 EPAX_insn_bbl 175
7.21.3.62 EPAX_insn_callTarget 175
7.21.3.63 EPAX_insn_condName 175
7.21.3.64 EPAX_insn_fallsThrough 175
7.21.3.65 EPAX_insn_func 175
7.21.3.66 EPAX_insn_isBranch 175
7.21.3.67 EPAX_insn_isFpop 176
7.21.3.68 EPAX_insn_isMemop 176

7.21.3.69 EPAX_insn_loop . . . 176
7.21.3.70 EPAX_insn_size . . . 176
7.21.3.71 EPAX_insn_sourceDatatypeSizeInBits . . . 176
7.21.3.72 EPAX_insn_sourceRegisterSizeInBits . . . 176
7.21.3.73 EPAX_insn_string . . . 176
7.21.3.74 EPAX_insn_targets . . . 176
7.21.3.75 EPAX_loop_cfg . . . 176
7.21.3.76 EPAX_loop_countBbl . . . 176
7.21.3.77 EPAX_loop_countExits . . . 177
7.21.3.78 EPAX_loop_countInsn . . . 177
7.21.3.79 EPAX_loop_depth . . . 177
7.21.3.80 EPAX_loop_exits . . . 177
7.21.3.81 EPAX_loop_findBbl . . . 177
7.21.3.82 EPAX_loop_findInsn . . . 177
7.21.3.83 EPAX_loop_firstBbl . . . 177
7.21.3.84 EPAX_loop_firstInsn . . . 177
7.21.3.85 EPAX_loop_func . . . 177
7.21.3.86 EPAX_loop_head . . . 177
7.21.3.87 EPAX_loop_index . . . 178
7.21.3.88 EPAX_loop_isInnerLoop . . . 178
7.21.3.89 EPAX_loop_isLastBbl . . . 178
7.21.3.90 EPAX_loop_isLastInsn . . . 178
7.21.3.91 EPAX_loop_nextBbl . . . 178
7.21.3.92 EPAX_loop_nextInsn . . . 178
7.21.3.93 EPAX_loop_parent . . . 178
7.21.3.94 EPAX_loop_size . . . 178
7.21.3.95 EPAX_loop_tail . . . 178
7.22 Interface.hpp File Reference . . . 178
7.22.1 Detailed Description . . . 181
7.22.2 LICENSE . . . 181
7.23 Loop.cpp File Reference . . . 181
7.23.1 Detailed Description . . . 182
7.23.2 LICENSE . . . 182
7.24 Loop.hpp File Reference . . . 182

7.24.1 Detailed Description . . . 182
7.24.2 LICENSE . . . 182
7.25 MachOBinary.cpp File Reference . . . 183
7.25.1 Detailed Description . . . 183
7.25.2 LICENSE . . . 183
7.25.3 Define Documentation . . . 184
7.25.3.1 CCASE . . . 184
7.25.3.2 MHDR32_ENTRY . . . 184
7.25.3.3 MHDR64_ENTRY . . . 184
7.25.3.4 SCASE . . . 184
7.26 MachOBinary.hpp File Reference . . . 184
7.26.1 Detailed Description . . . 184
7.26.2 LICENSE . . . 184
7.27 Section.cpp File Reference . . . 185
7.27.1 Detailed Description . . . 185
7.27.2 LICENSE . . . 185
7.28 Section.hpp File Reference . . . 185
7.28.1 Detailed Description . . . 186
7.28.2 LICENSE . . . 186
7.29 Symbol.cpp File Reference . . . 186
7.29.1 Detailed Description . . . 186
7.29.2 LICENSE . . . 186
7.30 Symbol.hpp File Reference . . . 187
7.30.1 Detailed Description . . . 187
7.30.2 LICENSE . . . 187

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

c_interface 9
EPAX 10
EPAX::Elf 45
EPAX::MachO 45

Chapter 2

Class Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

EPAX::dyn_bitset . . . 64
EPAXExport . . . 82
- EPAX::BasicBlock . . . 52
- EPAX::Binary . . . 56
- EPAX::ControlFlow . . . 60
- EPAX::Function . . . 93
- EPAX::Instruction . . . 98
- EPAX::Loop . . . 102
- EPAX::Section . . . 125
  - EPAX::StringTable . . . 136
    - EPAX::Elf::ElfStringTable . . . 73
  - EPAX::SymbolTable . . . 139
    - EPAX::Elf::ElfSymbolTable . . . 81
- EPAX::Symbol . . . 137
  - EPAX::Elf::ElfSymbol . . . 74
    - EPAX::Elf::ElfSymbol32 . . . 77
    - EPAX::Elf::ElfSymbol64 . . . 79

EPAX::FileBase . . . 83
- EPAX::DetachedText . . . 62
  - EPAX::Function . . . 93
- EPAX::Elf::FileHeader . . . 85
  - EPAX::Elf::FileHeader32 . . . 88
  - EPAX::Elf::FileHeader64 . . . 90
- EPAX::Elf::ProgramHeader . . . 118
  - EPAX::Elf::ProgramHeader32 . . . 121
  - EPAX::Elf::ProgramHeader64 . . . 123
- EPAX::Elf::SectionHeader . . . 127
  - EPAX::Elf::SectionHeader32 . . . 131

EPAX::Elf::SectionHeader64 134
EPAX::MachO::MachHeader 104
EPAX::MachO::MachHeader32 106
EPAX::MachO::MachHeader64 108
EPAX::Section 125
EPAX::Symbol 137
EPAX::IndexBase 96
EPAX::BasicBlock 52
EPAX::DetachedText 62
EPAX::Elf::ProgramHeader 118
EPAX::Elf::SectionHeader 127
EPAX::Instruction 98
EPAX::Loop 102
EPAX::Section 125
EPAX::Symbol 137
EPAX::MemoryBase 116
EPAX::BasicBlock 52
EPAX::DetachedText 62
EPAX::Elf::ProgramHeader 118
EPAX::Instruction 98
EPAX::Section 125
EPAX::Symbol 137
EPAX::NameBase 117
EPAX::BaseBinary 47
EPAX::Elf::ElfBinary 66
EPAX::Elf::ElfBinary32 70
EPAX::Elf::ElfBinary64 72
EPAX::MachO::MachOBinary 110
EPAX::MachO::MachOBinary32 113
EPAX::MachO::MachOBinary64 115
EPAX::Elf::SectionHeader 127
EPAX::InputFile 97
EPAX::Section 125
EPAX::Symbol 137
EPAX::SymbolBase 138
EPAX::Function 93

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

EPAX::BaseBinary 47
EPAX::BasicBlock 52
EPAX::Binary 56
EPAX::ControlFlow 60
EPAX::DetachedText 62
EPAX::dyn_bitset 64
EPAX::Elf::ElfBinary 66
EPAX::Elf::ElfBinary32 70
EPAX::Elf::ElfBinary64 72
EPAX::Elf::ElfStringTable 73
EPAX::Elf::ElfSymbol 74
EPAX::Elf::ElfSymbol32 77
EPAX::Elf::ElfSymbol64 79
EPAX::Elf::ElfSymbolTable 81
EPAXExport 82
EPAX::FileBase 83
EPAX::Elf::FileHeader 85
EPAX::Elf::FileHeader32 88
EPAX::Elf::FileHeader64 90
EPAX::Function 93
EPAX::IndexBase 96
EPAX::InputFile 97
EPAX::Instruction 98
EPAX::Loop 102
EPAX::MachO::MachHeader 104
EPAX::MachO::MachHeader32 106
EPAX::MachO::MachHeader64 108
EPAX::MachO::MachOBinary 110
EPAX::MachO::MachOBinary32 113

EPAX::MachO::MachOBinary64 115
EPAX::MemoryBase 116
EPAX::NameBase 117
EPAX::Elf::ProgramHeader 118
EPAX::Elf::ProgramHeader32 121
EPAX::Elf::ProgramHeader64 123
EPAX::Section 125
EPAX::Elf::SectionHeader 127
EPAX::Elf::SectionHeader32 131
EPAX::Elf::SectionHeader64 134
EPAX::StringTable 136
EPAX::Symbol 137
EPAX::SymbolBase 138
EPAX::SymbolTable 139

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

BaseClass.cpp . . . 143
BaseClass.hpp . . . 144
BasicBlock.cpp . . . 145
BasicBlock.hpp . . . 145
Binary.cpp . . . 146
Binary.hpp . . . 147
c_interface.py . . . 148
ControlFlow.cpp . . . 148
ControlFlow.hpp . . . 149
DataStruct.hpp . . . 150
ElfBinary.cpp . . . 151
ElfBinary.hpp . . . 153
EPAX.cpp . . . 154
EPAXCommonInternal.hpp . . . 155
Function.cpp . . . 160
Function.hpp . . . 160
InputFile.cpp . . . 161
InputFile.hpp . . . 162
Instruction.cpp . . . 162
Instruction.hpp . . . 163
Interface.cpp . . . 164
Interface.hpp . . . 178
Loop.cpp . . . 181
Loop.hpp . . . 182
MachOBinary.cpp . . . 183
MachOBinary.hpp . . . 184
Section.cpp . . . 185
Section.hpp . . . 185
Symbol.cpp . . . 186

Symbol.hpp 187

Chapter 5

Namespace Documentation

5.1 c_interface Namespace Reference

Functions

- def file_exists
- def error_die
- def print_usage
- def char_range
- def main

Variables

- list objs = ['BIN', 'SECT', 'FUNC', 'CFG', 'LOOP', 'BBL', 'INSN', 'SYM', 'FLOW']
- dictionary other = {}
- dictionary remove = {}

5.1.1 Function Documentation

5.1.1.1 def c_interface.char_range (*c1, c2*)

Definition at line 39 of file c_interface.py.

5.1.1.2 def c_interface.error_die (*msg = ''*)

Definition at line 30 of file c_interface.py.

5.1.1.3 def c_interface.file_exists (*f*)

Definition at line 25 of file c_interface.py.

5.1.1.4 def c_interface.main ( )

Definition at line 43 of file c_interface.py.

5.1.1.5 def c_interface.print_usage (*msg = ''*)

Definition at line 35 of file c_interface.py.

5.1.2 Variable Documentation

5.1.2.1 list c_interface::objs = ['BIN', 'SECT', 'FUNC', 'CFG', 'LOOP', 'BBL', 'INSN', 'SYM', 'FLOW']

Definition at line 7 of file c_interface.py.

5.1.2.2 dictionary c_interface::other = {}

Definition at line 8 of file c_interface.py.

5.1.2.3 dictionary c_interface::remove = {}

Definition at line 20 of file c_interface.py.

5.2 EPAX Namespace Reference

Namespaces

- namespace Elf
- namespace MachO

Classes

- class FileBase
- class MemoryBase
- class SymbolBase
- class IndexBase
- class NameBase
- class BaseBinary
- class BasicBlock
- class Binary
- class ControlFlow
- class dyn_bitset
- class DetachedText

- class Function
- class InputFile
- class Instruction
- class Loop
- class Section
- class Symbol
- class SymbolTable
- class StringTable

Typedefs

- typedef Binary * BIN
- typedef Section * SECT
- typedef Function * FUNC
- typedef ControlFlow * CFG
- typedef Loop * LOOP
- typedef BasicBlock * BBL
- typedef Instruction * INSN
- typedef Symbol * SYM
- typedef FlowEquation * FLOW

Enumerations

- enum BinaryFormat { BinaryFormat_undefined = 0, BinaryFormat_Elf32, BinaryFormat_Elf64, BinaryFormat_MachO32, BinaryFormat_MachO64, BinaryFormat_PE, BinaryFormat_total }

Functions

- bool compareMemory (MemoryBase *m1, MemoryBase *m2)
- void DFS (std::vector< BasicBlock * > &backedg, BasicBlock *start, dyn_bitset &v, dyn_bitset &c)
- void findDominators (std::vector< dyn_bitset * > &dominators, BasicBlock *start, std::vector< BasicBlock * > &bbs)
- void findBackEdges (std::vector< BasicBlock * > &backedg, BasicBlock *start, std::vector< BasicBlock * > bbs)
- BIN BIN_create (std::string fileName)
- std::string BIN_getName (BIN bin)
- void BIN_destroy (BIN bin)
- void BIN_run (BIN bin, int argc, char *argv[])
- FUNC BIN_firstFunc (BIN bin)
- FUNC BIN_nextFunc (BIN bin, FUNC func)
- bool BIN_isLastFunc (BIN bin, FUNC func)
- uint32_t BIN_countFunc (BIN bin)
- bool BIN_isExecutable (BIN bin)

- uint32_t BIN_fileSize (BIN bin)
- void BIN_printStaticFile (BIN bin, std::string fname)
- FUNC BIN_findFunc (BIN bin, uint64_t addr)
- FUNC FUNC_create (uint8_t *bytes, uint32_t size)
- void FUNC_destroy (FUNC func)
- void FUNC_print (FUNC func)
- std::string FUNC_name (FUNC func)
- uint32_t FUNC_size (FUNC func)
- uint64_t FUNC_addr (FUNC func)
- std::string FUNC_secName (FUNC func)
- BIN FUNC_bin (FUNC func)
- uint32_t FUNC_countBbl (FUNC func)
- BBL FUNC_findBbl (FUNC func, uint64_t addr)
- BBL FUNC_firstBbl (FUNC func)
- BBL FUNC_nextBbl (FUNC func, BBL bbl)
- bool FUNC_isLastBbl (FUNC func, BBL bbl)
- uint32_t FUNC_countInsn (FUNC func)
- INSN FUNC_findInsn (FUNC func, uint64_t addr)
- INSN FUNC_firstInsn (FUNC func)
- INSN FUNC_nextInsn (FUNC func, INSN insn)
- bool FUNC_isLastInsn (FUNC func, INSN insn)
- CFG FUNC_cfg (FUNC func)
- uint32_t FUNC_countTargets (FUNC func)
- uint32_t FUNC_targets (FUNC func, std::vector< FUNC > &funcList)
- uint32_t CFG_countLoop (CFG cfg)
- LOOP CFG_findLoop (CFG cfg, uint64_t addr)
- LOOP CFG_firstLoop (CFG cfg)
- LOOP CFG_nextLoop (CFG cfg, LOOP loop)
- bool CFG_isLastLoop (CFG cfg, LOOP loop)
- CFG LOOP_cfg (LOOP loop)
- FUNC LOOP_func (LOOP loop)
- uint32_t LOOP_size (LOOP loop)
- uint32_t LOOP_countBbl (LOOP loop)
- BBL LOOP_findBbl (LOOP loop, uint64_t addr)
- BBL LOOP_firstBbl (LOOP loop)
- BBL LOOP_nextBbl (LOOP loop, BBL bbl)
- bool LOOP_isLastBbl (LOOP loop, BBL bbl)
- uint32_t LOOP_countInsn (LOOP loop)
- INSN LOOP_findInsn (LOOP loop, uint64_t addr)
- INSN LOOP_firstInsn (LOOP loop)
- INSN LOOP_nextInsn (LOOP loop, INSN insn)
- bool LOOP_isLastInsn (LOOP loop, INSN insn)
- BBL LOOP_head (LOOP loop)
- BBL LOOP_tail (LOOP loop)
- uint32_t LOOP_countExits (LOOP loop)
- uint32_t LOOP_exits (LOOP loop, std::vector< EPAX::INSN > &insnList)

- bool LOOP_isInnerLoop (LOOP loop1, LOOP loop2)
- LOOP LOOP_parent (LOOP loop)
- uint32_t LOOP_index (LOOP loop)
- uint32_t LOOP_depth (LOOP loop)
- bool BBL_isHead (BBL bbl, INSN insn)
- bool BBL_isTail (BBL bbl, INSN insn)
- INSN BBL_head (BBL bbl)
- INSN BBL_tail (BBL bbl)
- FUNC BBL_func (BBL bbl)
- LOOP BBL_loop (BBL bbl)
- uint32_t BBL_size (BBL bbl)
- uint64_t BBL_addr (BBL bbl)
- uint32_t BBL_countInsn (BBL bbl)
- INSN BBL_findInsn (BBL bbl, uint64_t addr)
- INSN BBL_firstInsn (BBL bbl)
- INSN BBL_nextInsn (BBL bbl, INSN insn)
- bool BBL_isLastInsn (BBL bbl, INSN insn)
- uint32_t BBL_countTargets (BBL bbl)
- uint32_t BBL_targets (BBL bbl, std::vector< EPAX::BBL > &bblList)
- bool BBL_hasFallthroughTarget (BBL bbl)
- BBL BBL_fallthroughTarget (BBL bbl)
- uint32_t BBL_countJumpTargets (BBL bbl)
- uint32_t BBL_jumpTargets (BBL bbl, std::vector< EPAX::BBL > &bblList)
- uint32_t BBL_countSources (BBL bbl)
- uint32_t BBL_sources (BBL bbl, std::vector< EPAX::BBL > &bblList)
- uint32_t INSN_targets (INSN insn, std::vector< uint64_t > &tlist)
- BBL INSN_bbl (INSN insn)
- FUNC INSN_func (INSN insn)
- LOOP INSN_loop (INSN insn)
- uint64_t INSN_addr (INSN insn)
- std::string INSN_string (INSN insn)
- uint64_t INSN_callTarget (INSN insn)
- bool INSN_isBranch (INSN insn)
- bool INSN_isFpop (INSN insn)
- bool INSN_isMemop (INSN insn)
- uint32_t INSN_size (INSN insn)
- std::string INSN_condName (INSN insn)
- bool INSN_fallsThrough (INSN insn)
- uint32_t INSN_sourceRegisterSizeInBits (INSN insn)
- uint32_t INSN_sourceDatatypeSizeInBits (INSN insn)
- void BIN_run (BIN bin, int argc, char **argv)
- void FUNC_Destroy (FUNC func)
- uint32_t LOOP_exits (LOOP loop, std::vector< INSN > &insnList)
- uint32_t BBL_targets (BBL bbl, std::vector< BBL > &bblList)
- uint32_t BBL_jumpTargets (BBL bbl, std::vector< BBL > &bblList)
- uint32_t BBL_sources (BBL bbl, std::vector< BBL > &bblList)

5.2.1 Typedef Documentation

5.2.1.1 typedef BasicBlock* EPAX::BBL

Definition at line 52 of file Interface.hpp.

5.2.1.2 typedef Binary* EPAX::BIN

Definition at line 43 of file Interface.hpp.

5.2.1.3 typedef ControlFlow* EPAX::CFG

Definition at line 50 of file Interface.hpp.

5.2.1.4 typedef FlowEquation* EPAX::FLOW

Definition at line 55 of file Interface.hpp.

5.2.1.5 typedef Function* EPAX::FUNC

Definition at line 49 of file Interface.hpp.

5.2.1.6 typedef Instruction* EPAX::INSN

Definition at line 53 of file Interface.hpp.

5.2.1.7 typedef Loop* EPAX::LOOP

Definition at line 51 of file Interface.hpp.

5.2.1.8 typedef Section* EPAX::SECT

Definition at line 48 of file Interface.hpp.

5.2.1.9 typedef Symbol* EPAX::SYM

Definition at line 54 of file Interface.hpp.

5.2.2 Enumeration Type Documentation

5.2.2.1 enum EPAX::BinaryFormat

Binary format

Enumerator:

BinaryFormat_undefined

BinaryFormat_Elf32

BinaryFormat_Elf64

BinaryFormat_MachO32

BinaryFormat_MachO64

BinaryFormat_PE

BinaryFormat_total

Definition at line 34 of file Binary.hpp.

5.2.3 Function Documentation

5.2.3.1 uint64_t EPAX::BBL_addr (BBL *bbl*)

Definition at line 624 of file Interface.cpp.

5.2.3.2 uint32_t EPAX::BBL_countInsn (BBL *bbl*)

Get the number of INSNs in a BBL

Parameters

bbl	a BBL object

Returns

the number of INSNs in a BBL

Definition at line 629 of file Interface.cpp.

5.2.3.3 uint32_t EPAX::BBL_countJumpTargets (BBL *bbl*)

Counts the number of non-fallthrough targets for a BBL

Parameters

bbl	a BBL object

Returns

the number of targets for bbl that are not fallthrough targets

Definition at line 680 of file Interface.cpp.

5.2.3.4 uint32_t EPAX::BBL_countSources (BBL *bbl*)

Counts the number of control source blocks for a BBL

Parameters

bbl	a BBL object

Returns

the number of control source blocks for bbl

Definition at line 690 of file Interface.cpp.

5.2.3.5 uint32_t EPAX::BBL_countTargets (BBL *bbl*)

Gets the number of control flow targets for a BBL

Parameters

bbl	a BBL object

Returns

the number of BBLs that are control flow targets for bbl

Definition at line 656 of file Interface.cpp.

5.2.3.6 BBL EPAX::BBL_fallthroughTarget (BBL *bbl*)

Gets the fallthrough target for a BBL

Parameters

bbl	a BBL object

Returns

the BBL that is the fallthrough target of bbl, or NULL if no such BBL exists

Definition at line 675 of file Interface.cpp.

5.2.3.7 INSN EPAX::BBL_findInsn (BBL *bbl,* uint64_t *addr*)

Find the INSN within a BBL at a given address

Parameters

bbl	a BBL object
addr	a virtual address

Returns

the INSN within BBL that intersects with addr, or NULL if no such INSN exists

Definition at line 634 of file Interface.cpp.

5.2.3.8 INSN EPAX::BBL_firstInsn (BBL *bbl*)

Get the first INSN object in a BBL

Parameters

bbl	a BBL object

Returns

the first INSN in bbl

Definition at line 639 of file Interface.cpp.

5.2.3.9 FUNC EPAX::BBL_func (BBL *bbl*)

Get the function containing a BBL

Parameters

bbl	a BBL object

Returns

the FUNC containing bbl

Definition at line 609 of file Interface.cpp.

5.2.3.10 bool EPAX::BBL_hasFallthroughTarget (BBL *bbl*)

Tells whether control can fall through the end of a BBL

Parameters

bbl	a BBL object

Returns

true iff control can fall through the end of bbl

Definition at line 670 of file Interface.cpp.

5.2.3.11 INSN EPAX::BBL_head (BBL *bbl*)

Get the head INSN of a BBL

Parameters

bbl	a BBL object

Returns

the head INSN of bbl

Definition at line 599 of file Interface.cpp.

5.2.3.12 bool EPAX::BBL_isHead (BBL *bbl,* INSN *insn*)

Is an insn the head of a BBL

Parameters

bbl	a BBL object
insn	an INSN object

Returns

true iff insn is the head of bbl

Definition at line 563 of file Interface.cpp.

5.2.3.13 bool EPAX::BBL_isLastInsn (BBL *bbl,* INSN *insn*)

Tests whether a INSN is the last in a BBL

Parameters

bbl	a BBL object
insn	a INSN object

Returns

true iff insn is the last INSN object in bbl, false otherwise

Definition at line 650 of file Interface.cpp.

5.2.3.14 bool EPAX::BBL_isTail (BBL *bbl,* INSN *insn*)

Is an insn the tail of a BBL

Parameters

bbl	a BBL object
insn	an INSN object

Returns

true iff insn is the tail of bbl

Definition at line 581 of file Interface.cpp.

5.2.3.15 uint32_t EPAX::BBL_jumpTargets (BBL *bbl,* std::vector< EPAX::BBL > & *bblList*)

Definition at line 685 of file Interface.cpp.

5.2.3.16 uint32_t EPAX::BBL_jumpTargets (BBL *bbl,* std::vector< BBL > & *bblList*)

Gets the non-fallthrough targets for a BBL

Parameters

bbl	a BBL object
(out)	the non-fallthrough targets for bbl

Returns

the number of non-fallthrough targets for bbl

5.2.3.17 LOOP EPAX::BBL_loop (BBL *bbl*)

Get the loop containing a BBL

Parameters

bbl	a BBL object

Returns

the LOOP containing bbl, of NULL if no such LOOP exists

Definition at line 614 of file Interface.cpp.

5.2.3.18 INSN EPAX::BBL_nextInsn (BBL *bbl,* INSN *insn*)

Get the next INSN object in a BBL

Parameters

bbl	a BBL object
insn	a INSN object

Returns

the INSN from bbl that is subsequent to insn, or NULL if no such INSN exists

Definition at line 644 of file Interface.cpp.

5.2.3.19 uint32_t EPAX::BBL_size (BBL *bbl*)

Get the size of a BBL

Parameters

bbl	a BBL object

Returns

the size in bytes of bbl

Definition at line 619 of file Interface.cpp.

5.2.3.20 uint32_t EPAX::BBL_sources (BBL *bbl,* std::vector< EPAX::BBL > & *bblList*)

Definition at line 695 of file Interface.cpp.

5.2.3.21 uint32_t EPAX::BBL_sources (BBL *bbl,* std::vector< BBL > & *bblList*)

Gets the control source blocks for a BBL

Parameters

bbl	a BBL object
(out)	bblList the control source blocks for bbl

Returns

the number of control source blocks for bbl

5.2.3.22 INSN EPAX::BBL_tail (BBL *bbl*)

Get the tail INSN of a BBL

Parameters

bbl	a BBL object

Returns

the tail INSN of bbl

Definition at line 604 of file Interface.cpp.

5.2.3.23 uint32_t EPAX::BBL_targets (BBL *bbl,* std::vector< EPAX::BBL > & *bblList*)

Definition at line 661 of file Interface.cpp.

5.2.3.24 uint32_t EPAX::BBL_targets (BBL *bbl,* std::vector< BBL > & *bblList*)

Gets the control flow targets for a BBL

Parameters

bbl	a BBL object
(out)	the BBLs that are control flow targets for bbl

Returns

the number of BBLs that are control targets for bbl

5.2.3.25 uint32_t EPAX::BIN_countFunc (BIN *bin*)

Count the functions in a BIN

Parameters

bin	a BIN

Returns

the number of FUNCs in bin

Definition at line 94 of file Interface.cpp.

5.2.3.26 BIN EPAX::BIN_create (std::string *fileName*)

Creates a BIN object

Parameters

fileName	The name of a binary file. Allowed formats are: ELF, MachO

Returns

a BIN object created using the input parameter

Definition at line 43 of file Interface.cpp.

5.2.3.27 void EPAX::BIN_destroy (BIN *bin*)

frees all memory associated with a BIN object

Parameters

bin	a BIN object, which is set to NULL during this operation.

Returns

none

Definition at line 52 of file Interface.cpp.

5.2.3.28 uint32_t EPAX::BIN_fileSize (BIN *bin*)

Find the file size of a BIN

Parameters

bin	a BIN

Returns

the size of the file used to create bin

Definition at line 104 of file Interface.cpp.

5.2.3.29 FUNC EPAX::BIN_findFunc (BIN *bin,* uint64_t *addr*)

Find the function at a given virtual address

Parameters

bin	a BIN
addr	a virtual address

Returns

the FUNC at addr in bin

Definition at line 262 of file Interface.cpp.

5.2.3.30 FUNC EPAX::BIN_firstFunc (BIN *bin*)

Gets the first function in a BIN object

Parameters

bin	a BIN object

Returns

the first logical function in binary

Definition at line 69 of file Interface.cpp.

5.2.3.31 std::string EPAX::BIN_getName (BIN *bin*)

returns the name of a BIN object

Parameters

bin	a BIN

Returns

the name of the file used to create bin

Definition at line 47 of file Interface.cpp.

5.2.3.32 bool EPAX::BIN_isExecutable (BIN *bin*)

Is the BIN executable

Parameters

bin	a BIN

Returns

true iff bin is an executable file

Definition at line 99 of file Interface.cpp.

5.2.3.33 bool EPAX::BIN_isLastFunc (BIN *bin*, FUNC *func*)

Is a FUNC the last logical function in its BIN

Parameters

bin	a BIN
func	a FUNC from bin

Returns

true iff func is the last logical function in bin

Definition at line 88 of file Interface.cpp.

5.2.3.34 FUNC EPAX::BIN_nextFunc (BIN *bin*, FUNC *func*)

Gets the next logical function in a BIN object

Parameters

bin	a BIN object
func	a FUNC from binary

Returns

the logical function following func from bin, or NULL if func is the last such function

Definition at line 78 of file Interface.cpp.

5.2.3.35 void EPAX::BIN_printStaticFile (BIN *bin*, std::string *fname*)

Print a static file containing detailed information about the structures found in a BIN

Parameters

bin	a BIN
fname	the name of the output file to catch static analysis

Returns

none

Definition at line 109 of file Interface.cpp.

5.2.3.36 void EPAX::BIN_run (BIN *bin*, int *argc*, char * *argv[]*)

Definition at line 60 of file Interface.cpp.

5.2.3.37 void EPAX::BIN_run (BIN *bin*, int *argc*, char ** *argv*)

Runs a the program represented by BIN with arguments; does not return.

Parameters

bin	a BIN object for which BIN_isExecutable returns true
argc	the number of program arguments
argv	the program arguments

Returns

none

5.2.3.38 uint32_t EPAX::CFG_countLoop (CFG *cfg*)

Count the number of loops in a CFG

Parameters

cfg	a CFG object

Returns

the number of loops in cfg

Definition at line 399 of file Interface.cpp.

5.2.3.39 LOOP EPAX::CFG_findLoop (CFG *cfg*, uint64_t *addr*)

Find the LOOP within a CFG ad a given address

Parameters

cfg	a CFG object
addr	a virtual address

Returns

the loop within cfg at addr, or NULL if no such loop exists

Definition at line 404 of file Interface.cpp.

5.2.3.40 LOOP EPAX::CFG_firstLoop (CFG *cfg*)

Get the first loop in a CFG

Parameters

cfg	a CFG object

Returns

the first loop in cfg

Definition at line 409 of file Interface.cpp.

5.2.3.41 bool EPAX::CFG_isLastLoop (CFG *cfg,* LOOP *loop*)

Tests whether a LOOP is the last in a CFG

Parameters

cfg	a CFG object
loop	a LOOP object

Returns

true iff loop is the last LOOP in cfg

Definition at line 420 of file Interface.cpp.

5.2.3.42 LOOP EPAX::CFG_nextLoop (CFG *cfg,* LOOP *loop*)

Get the next loop in a CFG

Parameters

cfg	a CFG object
loop	a LOOP object

Returns

the successor to loop within cfg, or NULL if no such LOOP exists

Definition at line 414 of file Interface.cpp.

5.2.3.43 bool EPAX::compareMemory (MemoryBase * *m1*, MemoryBase * *m2*)

Definition at line 35 of file BaseClass.cpp.

5.2.3.44 void EPAX::DFS (std::vector< BasicBlock * > & *backedg*, BasicBlock * *start*, dyn_bitset & *v*, dyn_bitset & *c*)

Definition at line 58 of file ControlFlow.cpp.

5.2.3.45 void EPAX::findBackEdges (std::vector< BasicBlock * > & *backedg*, BasicBlock * *start*, std::vector< BasicBlock * > *bbs*)

Definition at line 131 of file ControlFlow.cpp.

5.2.3.46 void EPAX::findDominators (std::vector< dyn_bitset * > & *dominators*, BasicBlock * *start*, std::vector< BasicBlock * > & *bbs*)

Definition at line 78 of file ControlFlow.cpp.

5.2.3.47 uint64_t EPAX::FUNC_addr (FUNC *func*)

Get the virtual address of a FUNC

Parameters

func	a FUNC object

Returns

the virtual address of func

Definition at line 291 of file Interface.cpp.

5.2.3.48 BIN EPAX::FUNC_bin (FUNC *func*)

Get the BIN object that contains a FUNC

Parameters

func	a FUNC object

Returns

the BIN object associated with func

Definition at line 301 of file Interface.cpp.

5.2.3.49 CFG EPAX::FUNC_cfg (FUNC *func*)

Get the CFG attached to a FUNC

Parameters

func	a FUNC object

Returns

the CFG attached to func

Definition at line 384 of file Interface.cpp.

5.2.3.50 uint32_t EPAX::FUNC_countBbl (FUNC *func*)

Get the number of BBL objects in a FUNC

Parameters

func	a FUNC object

Returns

the number of BBL objects in func

Definition at line 306 of file Interface.cpp.

5.2.3.51 uint32_t EPAX::FUNC_countInsn (FUNC *func*)

Get the number of INSNs in a FUNC

Parameters

func	a FUNC object

Returns

the number of INSNs in a FUNC

Definition at line 333 of file Interface.cpp.

5.2.3.52 uint32_t EPAX::FUNC_countTargets (FUNC *func*)

Get the number of targets of (functions called by) a FUNC

Parameters

func	a FUNC object

Returns

the number of unique targets of the text of func

Definition at line 389 of file Interface.cpp.

5.2.3.53 FUNC EPAX::FUNC_create (uint8_t * *bytes*, uint32_t *size*)

Generate a function using the supplied bytes. Note that the size of the function found may be smaller than the size of the input buffer supplied. Use FUNC_size on the returned FUNC to find its size.

Parameters

bytes	a buffer of raw instruction bytes
size	the size of the buffer

Returns

a FUNC generated using the bytes supplied in buf

Definition at line 267 of file Interface.cpp.

5.2.3.54 void EPAX::FUNC_Destroy (FUNC *func*)

Destroy a function; note that it is an error to destroy a function that was not created with FUNC_create

Parameters

func	a FUNC object that was created with FUNC_Create

Returns

none

5.2.3.55 void EPAX::FUNC_destroy (FUNC *func*)

Definition at line 271 of file Interface.cpp.

5.2.3.56 BBL EPAX::FUNC_findBbl (FUNC *func,* uint64_t *addr*)

Find the BBL within a FUNC at a given address

Parameters

func	a FUNC object
addr	a virtual address

Returns

the BBL within FUNC that intersects with addr, or NULL if no such BBL exists

Definition at line 311 of file Interface.cpp.

5.2.3.57 INSN EPAX::FUNC_findInsn (FUNC *func,* uint64_t *addr*)

Find the INSN within a FUNC at a given address

Parameters

func	a FUNC object
addr	a virtual address

Returns

the INSN within FUNC that intersects with addr, or NULL if no such INSN exists

Definition at line 338 of file Interface.cpp.

5.2.3.58 BBL EPAX::FUNC_firstBbl (FUNC *func*)

Get the first BBL object in a FUNC

Parameters

func	a FUNC object

Returns

the first BBL in func

Definition at line 316 of file Interface.cpp.

5.2.3.59 INSN EPAX::FUNC_firstInsn (FUNC *func*)

Get the first INSN object in a FUNC

Parameters

func	a FUNC object

Returns

the first INSN in func

Definition at line 343 of file Interface.cpp.

5.2.3.60 bool EPAX::FUNC_isLastBbl (FUNC *func,* BBL *bbl*)

Tests whether a BBL is the last in a FUNC

Parameters

func	a FUNC object
bbl	a BBL object

Returns

true iff bbl is the last BBL object in func, false otherwise

Definition at line 327 of file Interface.cpp.

5.2.3.61 bool EPAX::FUNC_isLastInsn (FUNC *func,* INSN *insn*)

Tests whether a INSN is the last in a FUNC

Parameters

func	a FUNC object
insn	a INSN object

Returns

true iff insn is the last INSN object in func, false otherwise

Definition at line 368 of file Interface.cpp.

5.2.3.62 std::string EPAX::FUNC_name (FUNC *func*)

Get the name of a FUNC

Parameters

func	a FUNC object

Returns

the name of func, or NULL if no name can be found

Definition at line 281 of file Interface.cpp.

5.2.3.63 BBL EPAX::FUNC_nextBbl (FUNC *func,* BBL *bbl*)

Get the next BBL object in a FUNC

Parameters

func	a FUNC object
bbl	a BBL object

Returns

the BBL from func that is subsequent to bbl, or NULL if no such BBL exists

Definition at line 321 of file Interface.cpp.

5.2.3.64 INSN EPAX::FUNC_nextInsn (FUNC *func,* INSN *insn*)

Get the next INSN object in a FUNC

Parameters

func	a FUNC object
insn	a INSN object

Returns

the INSN from func that is subsequent to insn, or NULL if no such INSN exists

Definition at line 348 of file Interface.cpp.

5.2.3.65 void EPAX::FUNC_print (FUNC *func*)

Print a FUNC

Parameters

func	a FUNC object

Returns

none

Definition at line 276 of file Interface.cpp.

5.2.3.66 std::string EPAX::FUNC_secName (FUNC *func*)

Get the name of the section that contains a FUNC

Parameters

func	a FUNC object

Returns

the name of the section containing func, or NULL if it is unknown

Definition at line 296 of file Interface.cpp.

5.2.3.67 uint32_t EPAX::FUNC_size (FUNC *func*)

Get the size of a FUNC

Parameters

func	a FUNC object

Returns

the size of func in bytes

Definition at line 286 of file Interface.cpp.

5.2.3.68 uint32_t EPAX::FUNC_targets (FUNC *func,* std::vector< FUNC > & *funcList*)

Get the unique targets of (functions called by) a FUNC

Parameters

func	a FUNC object
(out)	funcList the unique targets of func

Returns

the number of unique targets of func

Definition at line 394 of file Interface.cpp.

5.2.3.69 uint64_t EPAX::INSN_addr (INSN *insn*)

Get the virtual address of an INSN

Parameters

insn	an INSN object

Returns

the virtual address of insn

Definition at line 731 of file Interface.cpp.

5.2.3.70 BBL EPAX::INSN_bbl (INSN *insn*)

Get the basic block of an INSN

Parameters

insn	an INSN object

Returns

the BBL that contains insn, or NULL if no such BBL exists

Definition at line 711 of file Interface.cpp.

5.2.3.71 uint64_t EPAX::INSN_callTarget (INSN *insn*)

Get the call target of an INSN

Parameters

insn	an INSN object

Returns

the address of the call target of insn, or 0 if the target cannot be found

Definition at line 741 of file Interface.cpp.

5.2.3.72 std::string EPAX::INSN_condName (INSN *insn*)

Get the string rep of the predicate condition of an INSN

Parameters

insn	an INSN object

Returns

the string representation of the predicate condition of insn

Definition at line 774 of file Interface.cpp.

5.2.3.73 bool EPAX::INSN_fallsThrough (INSN *insn*)

Can control fall through an INSN

Parameters

insn	an INSN object

Returns

true iff control can fall through insn

Definition at line 780 of file Interface.cpp.

5.2.3.74 FUNC EPAX::INSN_func (INSN *insn*)

Get the function of an INSN

Parameters

insn	an INSN object

Returns

the FUNC that contains insn, or NULL if no such FUNC exists

Definition at line 716 of file Interface.cpp.

5.2.3.75 bool EPAX::INSN_isBranch (INSN *insn*)

Is an INSN a branch

Parameters

insn	an INSN object

Returns

true iff insn is a branch instruction of any kind

Definition at line 754 of file Interface.cpp.

5.2.3.76 bool EPAX::INSN_isFpop (INSN *insn*)

Is an INSN an fp op

Parameters

insn	an INSN object

Returns

true iff either source or destination operands is fp data

Definition at line 759 of file Interface.cpp.

5.2.3.77 bool EPAX::INSN_isMemop (INSN *insn*)

Is an INSN a mem op

Parameters

insn	an INSN object

Returns

true iff the insns touches memory

Definition at line 764 of file Interface.cpp.

5.2.3.78 LOOP EPAX::INSN_loop (INSN *insn*)

Get the loop of an INSN

Parameters

insn	an INSN object

Returns

the LOOP that contains insn, or NULL if no such LOOP exists

Definition at line 726 of file Interface.cpp.

5.2.3.79 uint32_t EPAX::INSN_size (INSN *insn*)

Get the size in of an INSN in bytes

Parameters

insn	an INSN object

Returns

the size (in bytes) of insn

Definition at line 769 of file Interface.cpp.

5.2.3.80 uint32_t EPAX::INSN_sourceDatatypeSizeInBits (INSN *insn*)

Size of source datatype in bits

Parameters

insn	an INSN object

Returns

the number of bits in a source operand

Definition at line 790 of file Interface.cpp.

5.2.3.81 uint32_t EPAX::INSN_sourceRegisterSizeInBits (INSN *insn*)

Size of a source register in bits

Parameters

insn	an INSN object

Returns

the number of bits in a source register

Definition at line 785 of file Interface.cpp.

5.2.3.82 std::string EPAX::INSN_string (INSN *insn*)

Get a string representation of an INSN

Parameters

insn	an INSN object

Returns

the decoded string representation of insn

Definition at line 736 of file Interface.cpp.

5.2.3.83 uint32_t EPAX::INSN_targets (INSN *insn*, std::vector< uint64_t > & *tlist*)

Get the control target INSNs for an INSN

Parameters

insn	an INSN object
tlist	(out) the target INSNs of insn

Returns

the number of control targets of insn

Definition at line 704 of file Interface.cpp.

5.2.3.84 CFG EPAX::LOOP_cfg (LOOP *loop*)

Get the CFG associated with a LOOP

Parameters

loop	a LOOP object

Returns

the CFG associated with loop

Definition at line 426 of file Interface.cpp.

5.2.3.85 uint32_t EPAX::LOOP_countBbl (LOOP *loop*)

Get the number of BBL objects in a LOOP

Parameters

loop	a LOOP object

Returns

the number of BBL objects in loop

Definition at line 445 of file Interface.cpp.

5.2.3.86 uint32_t EPAX::LOOP_countExits (LOOP *loop*)

Get the number of exit points from a LOOP

Parameters

loop	a LOOP object

Returns

the number of exit points in loop

Definition at line 532 of file Interface.cpp.

5.2.3.87 uint32_t EPAX::LOOP_countInsn (LOOP *loop*)

Get the number of INSNs in a LOOP

Parameters

loop	a LOOP object

Returns

the number of INSNs in a LOOP

Definition at line 472 of file Interface.cpp.

5.2.3.88 uint32_t EPAX::LOOP_depth (LOOP *loop*)

Get the depth of a LOOP

Parameters

loop	a LOOP object

Returns

the depth of loop

Definition at line 558 of file Interface.cpp.

5.2.3.89 uint32_t EPAX::LOOP_exits (LOOP *loop,* std::vector< INSN > & *insnList*)

Get the instructions that are exit points from a particular LOOP

Parameters

loop	a LOOP object
(out)	insnList loop's exit points

Returns

the number of exit points in loop

5.2.3.90 uint32_t EPAX::LOOP_exits (LOOP *loop,* std::vector< EPAX::INSN > & *insnList*)

Definition at line 537 of file Interface.cpp.

5.2.3.91 BBL EPAX::LOOP_findBbl (LOOP *loop,* uint64_t *addr*)

Find the BBL within a LOOP at a given address

Parameters

loop	a LOOP object
addr	a virtual address

Returns

the BBL within LOOP that intersects with addr, or NULL if no such BBL exists

Definition at line 450 of file Interface.cpp.

5.2.3.92 INSN EPAX::LOOP_findInsn (LOOP *loop,* uint64_t *addr*)

Find the INSN within a LOOP at a given address

Parameters

loop	a LOOP object
addr	a virtual address

Returns

the INSN within LOOP that intersects with addr, or NULL if no such INSN exists

Definition at line 477 of file Interface.cpp.

5.2.3.93 BBL EPAX::LOOP_firstBbl (LOOP *loop*)

Get the first BBL object in a LOOP

Parameters

loop	a LOOP object

Returns

the first BBL in loop

Definition at line 455 of file Interface.cpp.

5.2.3.94 INSN EPAX::LOOP_firstInsn (LOOP *loop*)

Get the first INSN object in a LOOP

Parameters

loop	a LOOP object

Returns

the first INSN in loop

Definition at line 482 of file Interface.cpp.

5.2.3.95 FUNC EPAX::LOOP_func (LOOP *loop*)

Get the FUNC associated with a LOOP

Parameters

loop	a LOOP object

Returns

the FUNC associated with loop

Definition at line 431 of file Interface.cpp.

5.2.3.96 BBL EPAX::LOOP_head (LOOP *loop*)

Get the head basic block from a LOOP

Parameters

loop	a LOOP object

Returns

the head (target of the back edge) BBL in loop

Definition at line 522 of file Interface.cpp.

5.2.3.97 uint32_t EPAX::LOOP_index (LOOP *loop*)

Get the index of a LOOP

Parameters

loop	a LOOP object

Returns

the index of loop, which is unique within the containing FUNC/CFG

Definition at line 553 of file Interface.cpp.

5.2.3.98 bool EPAX::LOOP_isInnerLoop (LOOP *loop1,* LOOP *loop2*)

Find out whether a LOOP is an inner loop of another LOOP

Parameters

loop	a LOOP object
other	a LOOP object

Returns

true iff loop2 is an inner loop of loop1

Definition at line 542 of file Interface.cpp.

5.2.3.99 bool EPAX::LOOP_isLastBbl (LOOP *loop,* BBL *bbl*)

Tests whether a BBL is the last in a LOOP

Parameters

loop	a LOOP object
bbl	a BBL object

Returns

true iff bbl is the last BBL object in loop, false otherwise

Definition at line 466 of file Interface.cpp.

5.2.3.100 bool EPAX::LOOP_isLastInsn (LOOP *loop,* INSN *insn*)

Tests whether a INSN is the last in a LOOP

Parameters

loop	a LOOP object
insn	a INSN object

Returns

true iff insn is the last INSN object in loop, false otherwise

Definition at line 510 of file Interface.cpp.

5.2.3.101 BBL EPAX::LOOP_nextBbl (LOOP *loop,* BBL *bbl*)

Get the next BBL object in a LOOP

Parameters

loop	a LOOP object
bbl	a BBL object

Returns

the BBL from loop that is subsequent to bbl, or NULL if no such BBL exists

Definition at line 460 of file Interface.cpp.

5.2.3.102 INSN EPAX::LOOP_nextInsn (LOOP *loop,* INSN *insn*)

Get the next INSN object in a LOOP

Parameters

loop	a LOOP object
insn	a INSN object

Returns

the INSN from loop that is subsequent to insn, or NULL if no such INSN exists

Definition at line 491 of file Interface.cpp.

5.2.3.103 LOOP EPAX::LOOP_parent (LOOP *loop*)

Get the parent LOOP of a LOOP

Parameters

loop	a LOOP object

Returns

the parent LOOP of loop, or NULL no such loop exists

Definition at line 547 of file Interface.cpp.

5.2.3.104 uint32_t EPAX::LOOP_size (LOOP *loop*)

Get the size of a LOOP

Parameters

loop	a LOOP object

Returns

the size in bytes of loop

Definition at line 440 of file Interface.cpp.

5.2.3.105 BBL EPAX::LOOP_tail (LOOP *loop*)

Get the tail basic block from a LOOP

Parameters

loop	a LOOP object

Returns

the tail (source of the back edge) BBL in loop

Definition at line 527 of file Interface.cpp.

5.3 EPAX::Elf Namespace Reference

Classes

- class ElfBinary
- class ElfBinary32
- class ElfBinary64
- class FileHeader
- class FileHeader32
- class FileHeader64
- class ElfSymbol
- class ElfSymbol32
- class ElfSymbol64
- class ElfStringTable
- class ElfSymbolTable
- class SectionHeader
- class SectionHeader32
- class SectionHeader64
- class ProgramHeader
- class ProgramHeader32
- class ProgramHeader64

5.4 EPAX::MachO Namespace Reference

Classes

- class MachOBinary
- class MachOBinary32
- class MachOBinary64
- class MachHeader
- class MachHeader32
- class MachHeader64

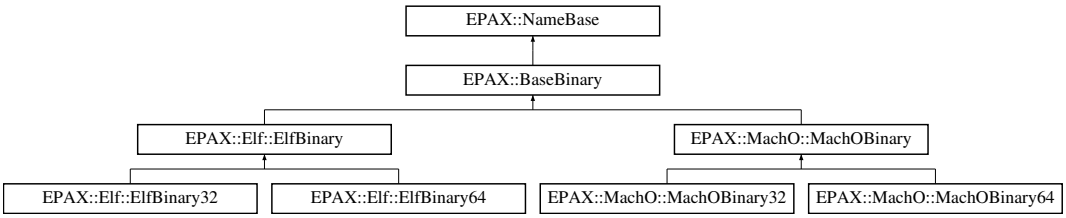
Chapter 6

Class Documentation

6.1 EPAX::BaseBinary Class Reference

`#include <BaseClass.hpp>`

Inheritance diagram for EPAX::BaseBinary:



Public Member Functions

- BaseBinary (std::string n)
- virtual ∼BaseBinary ()
- virtual BinaryFormat getFormat ()=0
- virtual uint64_t getStartAddr ()=0
- virtual void emit (std::string n)=0
- virtual bool verify ()=0
- virtual bool isARM ()=0
- virtual void describe ()=0
- virtual bool is32Bit ()=0
- virtual bool is64Bit ()=0
- virtual bool isExecutable ()=0
- virtual uint64_t functionEndAddress (Function ∗f, Function ∗nextf)=0
- uint32_t countFunctions ()
- Function ∗ getFirstFunction ()
- Function ∗ getNextFunction (Function ∗f)

- bool isLastFunction (Function *f)
- Function * findFunction (uint64_t addr)
- InputFile * getInputFile ()
- virtual uint64_t getFileSize ()
- virtual uint32_t getID ()
- virtual bool insideTextRange (uint64_t a)=0
- virtual void printSections (std::ostream &stream=std::cout)=0
- virtual void printFunctions (std::ostream &stream=std::cout)=0

Static Public Member Functions

- static const char * getFormatName (BinaryFormat f)

Protected Member Functions

- virtual void findFunctions ()=0
- void lazyFunctions ()
- virtual void findSymbols ()=0
- void lazySymbols ()

Protected Attributes

- InputFile * inputfile
- bool foundfunctions
- std::vector< Function * > * functions
- bool foundsymbols
- std::vector< SymbolTable * > * symtabs
- std::vector< StringTable * > * strtabs

6.1.1 Detailed Description

Definition at line 132 of file BaseClass.hpp.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 EPAX::BaseBinary::BaseBinary (std::string *n*)

Definition at line 58 of file BaseClass.cpp.

6.1.2.2 EPAX::BaseBinary::~BaseBinary ( ) [virtual]

Definition at line 67 of file BaseClass.cpp.

6.1.3 Member Function Documentation

6.1.3.1 uint32_t EPAX::BaseBinary::countFunctions ( )

Definition at line 125 of file BaseClass.cpp.

6.1.3.2 virtual void EPAX::BaseBinary::describe ( ) [pure virtual]

Implemented in EPAX::Elf::ElfBinary, and EPAX::MachO::MachOBinary.

6.1.3.3 virtual void EPAX::BaseBinary::emit (std::string *n*) [pure virtual]

Implemented in EPAX::Elf::ElfBinary, and EPAX::MachO::MachOBinary.

6.1.3.4 Function * EPAX::BaseBinary::findFunction (uint64_t *addr*)

Definition at line 153 of file BaseClass.cpp.

6.1.3.5 virtual void EPAX::BaseBinary::findFunctions ( ) [protected, pure virtual]

Finds and internally stores all functions in the image

Returns

none

Implemented in EPAX::Elf::ElfBinary, and EPAX::MachO::MachOBinary.

6.1.3.6 virtual void EPAX::BaseBinary::findSymbols ( ) [protected, pure virtual]

Finds and internally stores all symbols in the image

Returns

none

Implemented in EPAX::Elf::ElfBinary, and EPAX::MachO::MachOBinary.

6.1.3.7 virtual uint64_t EPAX::BaseBinary::functionEndAddress (Function * *f*, Function * *nextf*) [pure virtual]

Implemented in EPAX::Elf::ElfBinary, and EPAX::MachO::MachOBinary.

6.1.3.8 uint64_t EPAX::BaseBinary::getFileSize ( ) [virtual]

Definition at line 149 of file BaseClass.cpp.

6.1.3.9 Function * EPAX::BaseBinary::getFirstFunction ( )

Definition at line 115 of file BaseClass.cpp.

6.1.3.10 virtual BinaryFormat EPAX::BaseBinary::getFormat ( ) [pure virtual]

Implemented in EPAX::Elf::ElfBinary64, EPAX::Elf::ElfBinary32, EPAX::MachO::MachOBinary64, EPAX::MachO::MachOBinary32, EPAX::Elf::ElfBinary, and EPAX::MachO::MachOBinary.

6.1.3.11 const char * EPAX::BaseBinary::getFormatName (BinaryFormat *f*) [static]

Definition at line 98 of file BaseClass.cpp.

6.1.3.12 virtual uint32_t EPAX::BaseBinary::getID ( ) [inline, virtual]

Definition at line 190 of file BaseClass.hpp.

6.1.3.13 InputFile* EPAX::BaseBinary::getInputFile ( ) [inline]

Definition at line 186 of file BaseClass.hpp.

6.1.3.14 Function * EPAX::BaseBinary::getNextFunction (Function * *f*)

Definition at line 130 of file BaseClass.cpp.

6.1.3.15 virtual uint64_t EPAX::BaseBinary::getStartAddr ( ) [pure virtual]

Implemented in EPAX::Elf::ElfBinary, and EPAX::MachO::MachOBinary.

6.1.3.16 virtual bool EPAX::BaseBinary::insideTextRange (uint64_t *a*) [pure virtual]

Implemented in EPAX::Elf::ElfBinary, and EPAX::MachO::MachOBinary.

6.1.3.17 virtual bool EPAX::BaseBinary::is32Bit ( ) [pure virtual]

Implemented in EPAX::Elf::ElfBinary, and EPAX::MachO::MachOBinary.

6.1.3.18 virtual bool EPAX::BaseBinary::is64Bit ( ) [pure virtual]

Implemented in EPAX::Elf::ElfBinary, and EPAX::MachO::MachOBinary.

6.1.3.19 virtual bool EPAX::BaseBinary::isARM ( ) [pure virtual]

Implemented in EPAX::Elf::ElfBinary, and EPAX::MachO::MachOBinary.

6.1.3.20 virtual bool EPAX::BaseBinary::isExecutable ( ) [pure virtual]

Implemented in EPAX::Elf::ElfBinary, and EPAX::MachO::MachOBinary.

6.1.3.21 bool EPAX::BaseBinary::isLastFunction (Function * *f*)

Definition at line 138 of file BaseClass.cpp.

6.1.3.22 void EPAX::BaseBinary::lazyFunctions ( ) [protected]

Definition at line 103 of file BaseClass.cpp.

6.1.3.23 void EPAX::BaseBinary::lazySymbols ( ) [protected]

Definition at line 109 of file BaseClass.cpp.

6.1.3.24 virtual void EPAX::BaseBinary::printFunctions (std::ostream & *stream* = std::cout) [pure virtual]

Implemented in EPAX::Elf::ElfBinary, and EPAX::MachO::MachOBinary.

6.1.3.25 virtual void EPAX::BaseBinary::printSections (std::ostream & *stream* = std::cout) [pure virtual]

Implemented in EPAX::Elf::ElfBinary, and EPAX::MachO::MachOBinary.

6.1.3.26 virtual bool EPAX::BaseBinary::verify ( ) [pure virtual]

Implemented in EPAX::Elf::ElfBinary, and EPAX::MachO::MachOBinary.

6.1.4 Member Data Documentation

6.1.4.1 bool EPAX::BaseBinary::foundfunctions [protected]

Definition at line 147 of file BaseClass.hpp.

6.1.4.2 bool EPAX::BaseBinary::foundsymbols [protected]

Definition at line 157 of file BaseClass.hpp.

6.1.4.3 std::vector<Function*>* EPAX::BaseBinary::functions [protected]

Definition at line 148 of file BaseClass.hpp.

6.1.4.4 InputFile* EPAX::BaseBinary::inputfile [protected]

The image file

Definition at line 138 of file BaseClass.hpp.

6.1.4.5 std::vector<StringTable*>* EPAX::BaseBinary::strtabs [protected]

Definition at line 159 of file BaseClass.hpp.

6.1.4.6 std::vector<SymbolTable*>* EPAX::BaseBinary::symtabs [protected]

Definition at line 158 of file BaseClass.hpp.

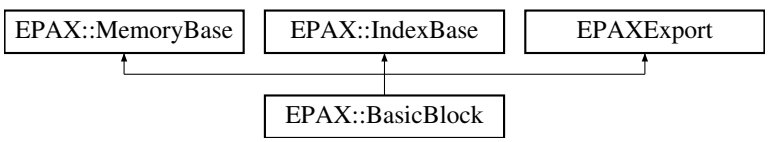
The documentation for this class was generated from the following files:

- BaseClass.hpp
- BaseClass.cpp

6.2 EPAX::BasicBlock Class Reference

`#include <BasicBlock.hpp>`

Inheritance diagram for EPAX::BasicBlock:



Public Member Functions

- BasicBlock (Function *f, uint64_t a, uint32_t i)
- virtual ~BasicBlock ()
- uint32_t countInstructions ()
- void addInstruction (Instruction *insn)
- Instruction * getInstruction (uint32_t idx)
- Instruction * head ()
- Instruction * tail ()
- Instruction * findInstruction (uint64_t addr)
- void addSource (BasicBlock *bb)
- void addTarget (BasicBlock *bb)
- uint32_t countSources ()
- uint32_t countTargets ()
- BasicBlock * getSource (uint32_t idx)
- BasicBlock * getTarget (uint32_t idx)
- bool isReachable ()
- void setUnreachable ()
- Function * getFunction ()
- Loop * getLoop ()
- void setLoop (Loop *l)
- ControlFlow * getControlFlow ()
- bool isFallThrough ()
- void print (std::ostream &stream=std::cout)

6.2.1 Detailed Description

Definition at line 38 of file BasicBlock.hpp.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 EPAX::BasicBlock::BasicBlock (Function * *f*, uint64_t *a*, uint32_t *i*)

Definition at line 33 of file BasicBlock.cpp.

6.2.2.2 EPAX::BasicBlock::~BasicBlock ( ) [virtual]

Definition at line 43 of file BasicBlock.cpp.

6.2.3 Member Function Documentation

6.2.3.1 void EPAX::BasicBlock::addInstruction (Instruction * *insn*)

Definition at line 138 of file BasicBlock.cpp.

6.2.3.2 void EPAX::BasicBlock::addSource (BasicBlock * *bb*)

Definition at line 50 of file BasicBlock.cpp.

6.2.3.3 void EPAX::BasicBlock::addTarget (BasicBlock * *bb*)

Definition at line 54 of file BasicBlock.cpp.

6.2.3.4 uint32_t EPAX::BasicBlock::countInstructions ( )

Definition at line 134 of file BasicBlock.cpp.

6.2.3.5 uint32_t EPAX::BasicBlock::countSources ( ) [inline]

Definition at line 64 of file BasicBlock.hpp.

6.2.3.6 uint32_t EPAX::BasicBlock::countTargets ( ) [inline]

Definition at line 65 of file BasicBlock.hpp.

6.2.3.7 Instruction * EPAX::BasicBlock::findInstruction (uint64_t *addr*)

Definition at line 86 of file BasicBlock.cpp.

6.2.3.8 ControlFlow * EPAX::BasicBlock::getControlFlow ( )

Definition at line 150 of file BasicBlock.cpp.

6.2.3.9 Function* EPAX::BasicBlock::getFunction ( ) [inline]

Definition at line 72 of file BasicBlock.hpp.

6.2.3.10 Instruction * EPAX::BasicBlock::getInstruction (uint32_t *idx*)

Definition at line 143 of file BasicBlock.cpp.

6.2.3.11 Loop* EPAX::BasicBlock::getLoop ( ) [inline]

Definition at line 74 of file BasicBlock.hpp.

6.2.3.12 BasicBlock * EPAX::BasicBlock::getSource (uint32_t *idx*)

Definition at line 58 of file BasicBlock.cpp.

6.2.3.13 BasicBlock * EPAX::BasicBlock::getTarget (uint32_t *idx*)

Definition at line 65 of file BasicBlock.cpp.

6.2.3.14 Instruction * EPAX::BasicBlock::head ( )

Definition at line 72 of file BasicBlock.cpp.

6.2.3.15 bool EPAX::BasicBlock::isFallThrough ( )

Definition at line 157 of file BasicBlock.cpp.

6.2.3.16 bool EPAX::BasicBlock::isReachable ( ) [inline]

Definition at line 69 of file BasicBlock.hpp.

6.2.3.17 void EPAX::BasicBlock::print (std::ostream & *stream* = `std::cout`)

Definition at line 100 of file BasicBlock.cpp.

6.2.3.18 void EPAX::BasicBlock::setLoop (Loop * *l*) [inline]

Definition at line 75 of file BasicBlock.hpp.

6.2.3.19 void EPAX::BasicBlock::setUnreachable ( ) [inline]

Definition at line 70 of file BasicBlock.hpp.

6.2.3.20 Instruction * EPAX::BasicBlock::tail ( )

Definition at line 79 of file BasicBlock.cpp.

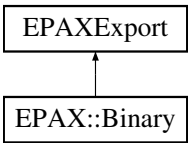
The documentation for this class was generated from the following files:

- BasicBlock.hpp
- BasicBlock.cpp

6.3 EPAX::Binary Class Reference

`#include <Binary.hpp>`

Inheritance diagram for EPAX::Binary:



Public Member Functions

- Binary (std::string n)
- Binary (std::string n, BinaryFormat f)
- virtual ~Binary ()
- void runBasic (int argc, char *argv[])
- uint64_t getStartAddr ()
- std::string getName ()
- BinaryFormat getFormat ()
- const char * getFormatName ()
- uint32_t countFunctions ()
- Function * getFirstFunction ()
- Function * getNextFunction (Function *f)
- bool isLastFunction (Function *f)
- Function * findFunction (uint64_t addr)
- bool isExecutable ()
- void printStaticFile (std::string &fname)
- void printStaticFile (const char *fname)
- uint32_t getFileSize ()

Static Public Member Functions

- static BinaryFormat detectFormat (std::string n)

6.3.1 Detailed Description

A thin wrapper around the classes which will hold all of the useful information about program binaryies. The idea is that this class will provide a single interface on top of any number of different formats.

Definition at line 51 of file Binary.hpp.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 EPAX::Binary::Binary (std::string *n*)

Constructs an Binary object.

Parameters

n	The name of a file. Format will be set based on the file's contents.

Definition at line 35 of file Binary.cpp.

6.3.2.2 EPAX::Binary::Binary (std::string *n*, BinaryFormat *f*)

Constructs an Binary object.

Parameters

n	The name of a file.
f	The format of the file.

Definition at line 41 of file Binary.cpp.

6.3.2.3 EPAX::Binary::~Binary ( ) [virtual]

Destroys an Binary instance. Should not be called directly.

Definition at line 47 of file Binary.cpp.

6.3.3 Member Function Documentation

6.3.3.1 uint32_t EPAX::Binary::countFunctions ( )

Counts the functions in the binary

Returns

the number of functions in the binary

Definition at line 225 of file Binary.cpp.

6.3.3.2 BinaryFormat EPAX::Binary::detectFormat (std::string *n*) [static]

Attempts to guess the format of an binary file

Returns

the format of the binary file, or BinaryFormat_undefined(0) if the format cannot be found

Definition at line 160 of file Binary.cpp.

6.3.3.3 Function * EPAX::Binary::findFunction (uint64_t *addr*)

Definition at line 221 of file Binary.cpp.

6.3.3.4 uint32_t EPAX::Binary::getFileSize ( )

Definition at line 233 of file Binary.cpp.

6.3.3.5 Function * EPAX::Binary::getFirstFunction ( )

Gets the first function in the binary

Returns

the first function in the binary

Definition at line 209 of file Binary.cpp.

6.3.3.6 BinaryFormat EPAX::Binary::getFormat ( ) [inline]

Gets the format of the binary.

Returns

the format of this binary

Definition at line 106 of file Binary.hpp.

6.3.3.7 const char * EPAX::Binary::getFormatName ( )

Gets a string representation of the format of this binary.

@ return a string representation of the format of this binary

Definition at line 123 of file Binary.cpp.

6.3.3.8 std::string EPAX::Binary::getName ( )

Definition at line 205 of file Binary.cpp.

6.3.3.9 Function * EPAX::Binary::getNextFunction (Function * *f*)

Gets the next function in the binary

Parameters

f	a function in the binary

Returns

the function following f

Definition at line 213 of file Binary.cpp.

6.3.3.10 uint64_t EPAX::Binary::getStartAddr ( )

Definition at line 201 of file Binary.cpp.

6.3.3.11 bool EPAX::Binary::isExecutable ( )

Definition at line 229 of file Binary.cpp.

6.3.3.12 bool EPAX::Binary::isLastFunction (Function * *f*)

Tells whether this is the last function in the binary

Parameters

f	a function in the binary

Returns

true iff f is the last function in the binary

Definition at line 217 of file Binary.cpp.

6.3.3.13 void EPAX::Binary::printStaticFile (std::string & *fname*)

Definition at line 53 of file Binary.cpp.

6.3.3.14 void EPAX::Binary::printStaticFile (const char * *fname*)

Definition at line 57 of file Binary.cpp.

6.3.3.15 void EPAX::Binary::runBasic (int *argc*, char * *argv[]*)

Definition at line 69 of file Binary.cpp.

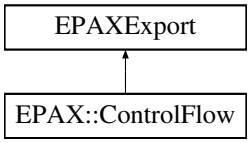
The documentation for this class was generated from the following files:

- Binary.hpp
- Binary.cpp

6.4 EPAX::ControlFlow Class Reference

`#include <ControlFlow.hpp>`

Inheritance diagram for EPAX::ControlFlow:



Public Member Functions

- ControlFlow (Function *f, std::vector< BasicBlock * > bbs)
- virtual ~ControlFlow ()
- Function * getFunction ()
- void print (std::ostream &stream=std::cout)
- void dot_print (std::ostream &stream=std::cout)
- uint32_t countBasicBlocks ()
- BasicBlock * findBasicBlock (uint64_t addr)
- BasicBlock * getBasicBlock (uint32_t idx)
- uint32_t countInstructions ()
- Instruction * findInstruction (uint64_t addr)
- Instruction * getInstruction (uint32_t idx)
- uint32_t countLoops ()
- Loop * findLoop (uint64_t addr)
- Loop * getLoop (uint32_t idx)
- Loop * getParentOf (Loop *loop)

6.4.1 Detailed Description

Definition at line 37 of file ControlFlow.hpp.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 EPAX::ControlFlow::ControlFlow (Function * *f*, std::vector< BasicBlock * > *bbs*)

Definition at line 35 of file ControlFlow.cpp.

6.4.2.2 EPAX::ControlFlow::~ControlFlow ( ) [virtual]

Definition at line 42 of file ControlFlow.cpp.

6.4.3 Member Function Documentation

6.4.3.1 uint32_t EPAX::ControlFlow::countBasicBlocks ( )

Definition at line 300 of file ControlFlow.cpp.

6.4.3.2 uint32_t EPAX::ControlFlow::countInstructions ( )

Definition at line 322 of file ControlFlow.cpp.

6.4.3.3 uint32_t EPAX::ControlFlow::countLoops ( )

Definition at line 344 of file ControlFlow.cpp.

6.4.3.4 void EPAX::ControlFlow::dot_print (std::ostream & *stream* = std::cout)

Definition at line 275 of file ControlFlow.cpp.

6.4.3.5 BasicBlock * EPAX::ControlFlow::findBasicBlock (uint64_t *addr*)

Definition at line 304 of file ControlFlow.cpp.

6.4.3.6 Instruction * EPAX::ControlFlow::findInstruction (uint64_t *addr*)

Definition at line 326 of file ControlFlow.cpp.

6.4.3.7 Loop * EPAX::ControlFlow::findLoop (uint64_t *addr*)

Definition at line 348 of file ControlFlow.cpp.

6.4.3.8 BasicBlock * EPAX::ControlFlow::getBasicBlock (uint32_t *idx*)

Definition at line 315 of file ControlFlow.cpp.

6.4.3.9 Function* EPAX::ControlFlow::getFunction ( ) [inline]

Definition at line 52 of file ControlFlow.hpp.

6.4.3.10 Instruction * EPAX::ControlFlow::getInstruction (uint32_t *idx*)

Definition at line 337 of file ControlFlow.cpp.

6.4.3.11 Loop * EPAX::ControlFlow::getLoop (uint32_t *idx*)

Definition at line 368 of file ControlFlow.cpp.

6.4.3.12 Loop * EPAX::ControlFlow::getParentOf (Loop * *loop*)

Definition at line 291 of file ControlFlow.cpp.

6.4.3.13 void EPAX::ControlFlow::print (std::ostream & *stream* = std::cout)

Definition at line 267 of file ControlFlow.cpp.

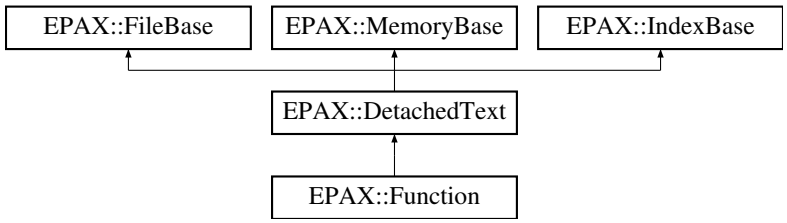
The documentation for this class was generated from the following files:

- ControlFlow.hpp
- ControlFlow.cpp

6.5 EPAX::DetachedText Class Reference

`#include <Function.hpp>`

Inheritance diagram for EPAX::DetachedText:



Public Member Functions

- DetachedText (BaseBinary *b, uint64_t o, uint64_t s, uint64_t a, uint32_t i)
- virtual ~DetachedText ()
- virtual void print (std::ostream &stream=std::cout)

Protected Attributes

- std::vector< void * > instructions

6.5.1 Detailed Description

Definition at line 43 of file Function.hpp.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 EPAX::DetachedText::DetachedText (BaseBinary * *b*, uint64_t *o*, uint64_t *s*, uint64_t *a*, uint32_t *i*)

Definition at line 36 of file Function.cpp.

6.5.2.2 virtual EPAX::DetachedText::~DetachedText ( ) [inline, virtual]

Definition at line 49 of file Function.hpp.

6.5.3 Member Function Documentation

6.5.3.1 virtual void EPAX::DetachedText::print (std::ostream & *stream* = `std::cout`) [inline, virtual]

Reimplemented in EPAX::Function.

Definition at line 51 of file Function.hpp.

6.5.4 Member Data Documentation

6.5.4.1 std::vector<void*> EPAX::DetachedText::instructions [protected]

Definition at line 45 of file Function.hpp.

The documentation for this class was generated from the following files:

- Function.hpp
- Function.cpp

6.6 EPAX::dyn_bitset Class Reference

`#include <DataStruct.hpp>`

Public Member Functions

- dyn_bitset (uint32_t s)
- ~dyn_bitset ()
- uint32_t size ()
- void clear ()
- void set (uint32_t idx)
- void set ()
- bool has (uint32_t idx)
- const dyn_bitset & operator&= (const dyn_bitset &a)
- const dyn_bitset & operator|= (const dyn_bitset &a)
- const dyn_bitset & operator= (const dyn_bitset &a)
- bool operator== (const dyn_bitset &a)
- bool operator!= (const dyn_bitset &a)
- void print ()

Public Attributes

- uint8_t * _elements
- uint32_t _size

6.6.1 Detailed Description

Definition at line 32 of file DataStruct.hpp.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 EPAX::dyn_bitset::dyn_bitset (uint32_t *s*) [inline]

Definition at line 49 of file DataStruct.hpp.

6.6.2.2 EPAX::dyn_bitset::~dyn_bitset ( ) [inline]

Definition at line 54 of file DataStruct.hpp.

6.6.3 Member Function Documentation

6.6.3.1 void EPAX::dyn_bitset::clear ( ) [inline]

Definition at line 64 of file DataStruct.hpp.

6.6.3.2 bool EPAX::dyn_bitset::has (uint32_t *idx*) [inline]

Definition at line 81 of file DataStruct.hpp.

6.6.3.3 bool EPAX::dyn_bitset::operator!= (const dyn_bitset & *a*) [inline]

Definition at line 120 of file DataStruct.hpp.

6.6.3.4 const dyn_bitset& EPAX::dyn_bitset::operator&= (const dyn_bitset & *a*) [inline]

Definition at line 86 of file DataStruct.hpp.

6.6.3.5 const dyn_bitset& EPAX::dyn_bitset::operator= (const dyn_bitset & *a*) [inline]

Definition at line 102 of file DataStruct.hpp.

6.6.3.6 bool EPAX::dyn_bitset::operator== (const dyn_bitset & *a*) [inline]

Definition at line 110 of file DataStruct.hpp.

6.6.3.7 const dyn_bitset& EPAX::dyn_bitset::operator|= (const dyn_bitset & *a*) [inline]

Definition at line 94 of file DataStruct.hpp.

6.6.3.8 void EPAX::dyn_bitset::print ( ) [inline]

Definition at line 124 of file DataStruct.hpp.

6.6.3.9 void EPAX::dyn_bitset::set (uint32_t *idx*) [inline]

Definition at line 70 of file DataStruct.hpp.

6.6.3.10 void EPAX::dyn_bitset::set ( ) [inline]

Definition at line 75 of file DataStruct.hpp.

6.6.3.11 uint32_t EPAX::dyn_bitset::size ( ) [inline]

Definition at line 60 of file DataStruct.hpp.

6.6.4 Member Data Documentation

6.6.4.1 uint8_t* EPAX::dyn_bitset::_elements

Definition at line 34 of file DataStruct.hpp.

6.6.4.2 uint32_t EPAX::dyn_bitset::_size

Definition at line 35 of file DataStruct.hpp.

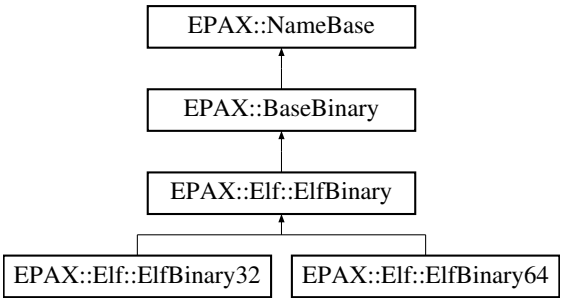
The documentation for this class was generated from the following file:

- DataStruct.hpp

6.7 EPAX::Elf::ElfBinary Class Reference

`#include <ElfBinary.hpp>`

Inheritance diagram for EPAX::Elf::ElfBinary:



Public Member Functions

- ElfBinary (std::string n)
- virtual ~ElfBinary ()
- virtual BinaryFormat getFormat ()=0
- uint64_t getStartAddr ()
- void emit (std::string n)
- bool verify ()
- bool isARM ()
- void describe ()
- void findFunctions ()
- void findSymbols ()
- void findSections ()
- void findSegments ()
- bool is32Bit ()

- bool is64Bit ()
- bool isExecutable ()
- ElfStringTable * findStringtable (uint32_t i)
- bool insideTextRange (uint64_t a)
- void printSections (std::ostream &stream=std::cout)
- void printFunctions (std::ostream &stream=std::cout)
- uint64_t vaddrToFile (uint64_t v)
- uint64_t functionEndAddress (Function *f, Function *nextf)

Protected Attributes

- FileHeader * fileheader
- bool foundsections
- std::vector< SectionHeader * > * sections
- bool foundsegments
- std::vector< ProgramHeader * > * segments

6.7.1 Detailed Description

Definition at line 42 of file ElfBinary.hpp.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 EPAX::Elf::ElfBinary::ElfBinary (std::string *n*)

Definition at line 47 of file ElfBinary.cpp.

6.7.2.2 EPAX::Elf::ElfBinary::~ElfBinary ( ) [virtual]

Definition at line 55 of file ElfBinary.cpp.

6.7.3 Member Function Documentation

6.7.3.1 void EPAX::Elf::ElfBinary::describe ( ) [virtual]

Implements EPAX::BaseBinary.

Definition at line 120 of file ElfBinary.cpp.

6.7.3.2 void EPAX::Elf::ElfBinary::emit (std::string *n*) [virtual]

Implements EPAX::BaseBinary.

Definition at line 144 of file ElfBinary.cpp.

6.7.3.3 void EPAX::Elf::ElfBinary::findFunctions ( ) [virtual]

Finds and internally stores all functions in the image

Returns

none

Implements EPAX::BaseBinary.

Definition at line 158 of file ElfBinary.cpp.

6.7.3.4 void EPAX::Elf::ElfBinary::findSections ( )

Definition at line 278 of file ElfBinary.cpp.

6.7.3.5 void EPAX::Elf::ElfBinary::findSegments ( )

Definition at line 299 of file ElfBinary.cpp.

6.7.3.6 ElfStringTable * EPAX::Elf::ElfBinary::findStringtable (uint32_t *i*)

Definition at line 220 of file ElfBinary.cpp.

6.7.3.7 void EPAX::Elf::ElfBinary::findSymbols ( ) [virtual]

Finds and internally stores all symbols in the image

Returns

none

Implements EPAX::BaseBinary.

Definition at line 230 of file ElfBinary.cpp.

6.7.3.8 uint64_t EPAX::Elf::ElfBinary::functionEndAddress (Function * *f*, Function * *nextf*) [virtual]

Implements EPAX::BaseBinary.

Definition at line 88 of file ElfBinary.cpp.

6.7.3.9 virtual BinaryFormat EPAX::Elf::ElfBinary::getFormat ( ) [pure virtual]

Implements EPAX::BaseBinary.

Implemented in EPAX::Elf::ElfBinary64, and EPAX::Elf::ElfBinary32.

6.7.3.10 uint64_t EPAX::Elf::ElfBinary::getStartAddr ( ) [virtual]

Implements EPAX::BaseBinary.

Definition at line 108 of file ElfBinary.cpp.

6.7.3.11 bool EPAX::Elf::ElfBinary::insideTextRange (uint64_t *a*) [virtual]

Implements EPAX::BaseBinary.

Definition at line 78 of file ElfBinary.cpp.

6.7.3.12 bool EPAX::Elf::ElfBinary::is32Bit ( ) [virtual]

Implements EPAX::BaseBinary.

Definition at line 136 of file ElfBinary.cpp.

6.7.3.13 bool EPAX::Elf::ElfBinary::is64Bit ( ) [virtual]

Implements EPAX::BaseBinary.

Definition at line 140 of file ElfBinary.cpp.

6.7.3.14 bool EPAX::Elf::ElfBinary::isARM ( ) [virtual]

Implements EPAX::BaseBinary.

Definition at line 116 of file ElfBinary.cpp.

6.7.3.15 bool EPAX::Elf::ElfBinary::isExecutable ( ) [virtual]

Implements EPAX::BaseBinary.

Definition at line 458 of file ElfBinary.cpp.

6.7.3.16 void EPAX::Elf::ElfBinary::printFunctions (std::ostream & *stream* = std::cout) [virtual]

Implements EPAX::BaseBinary.

Definition at line 210 of file ElfBinary.cpp.

6.7.3.17 void EPAX::Elf::ElfBinary::printSections (std::ostream & *stream* = std::cout) [virtual]

Implements EPAX::BaseBinary.

Definition at line 704 of file ElfBinary.cpp.

6.7.3.18 uint64_t EPAX::Elf::ElfBinary::vaddrToFile (uint64_t *v*)

Definition at line 148 of file ElfBinary.cpp.

6.7.3.19 bool EPAX::Elf::ElfBinary::verify ( ) [virtual]

Implements EPAX::BaseBinary.

Definition at line 112 of file ElfBinary.cpp.

6.7.4 Member Data Documentation

6.7.4.1 FileHeader* EPAX::Elf::ElfBinary::fileheader [protected]

Definition at line 44 of file ElfBinary.hpp.

6.7.4.2 bool EPAX::Elf::ElfBinary::foundsections [protected]

Definition at line 46 of file ElfBinary.hpp.

6.7.4.3 bool EPAX::Elf::ElfBinary::foundsegments [protected]

Definition at line 48 of file ElfBinary.hpp.

6.7.4.4 std::vector<SectionHeader*>* EPAX::Elf::ElfBinary::sections [protected]

Definition at line 47 of file ElfBinary.hpp.

6.7.4.5 std::vector<ProgramHeader*>* EPAX::Elf::ElfBinary::segments [protected]

Definition at line 49 of file ElfBinary.hpp.

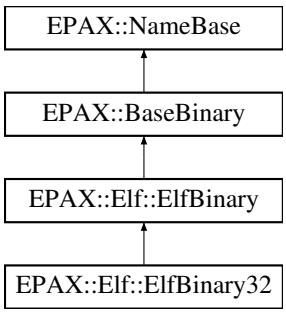
The documentation for this class was generated from the following files:

- ElfBinary.hpp
- ElfBinary.cpp

6.8 EPAX::Elf::ElfBinary32 Class Reference

`#include <ElfBinary.hpp>`

Inheritance diagram for EPAX::Elf::ElfBinary32:



Public Member Functions

- ElfBinary32 (std::string n)
- virtual ~ElfBinary32 ()
- BinaryFormat getFormat ()

6.8.1 Detailed Description

Definition at line 84 of file ElfBinary.hpp.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 EPAX::Elf::ElfBinary32::ElfBinary32 (std::string *n*)

Definition at line 124 of file ElfBinary.cpp.

6.8.2.2 virtual EPAX::Elf::ElfBinary32::~ElfBinary32 ( ) [inline, virtual]

Definition at line 87 of file ElfBinary.hpp.

6.8.3 Member Function Documentation

6.8.3.1 BinaryFormat EPAX::Elf::ElfBinary32::getFormat ( ) [inline, virtual]

Implements EPAX::Elf::ElfBinary.

Definition at line 89 of file ElfBinary.hpp.

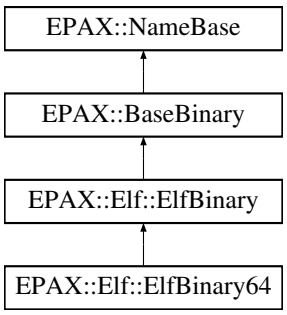
The documentation for this class was generated from the following files:

- ElfBinary.hpp
- ElfBinary.cpp

6.9 EPAX::Elf::ElfBinary64 Class Reference

`#include <ElfBinary.hpp>`

Inheritance diagram for EPAX::Elf::ElfBinary64:



Public Member Functions

- ElfBinary64 (std::string n)
- virtual ~ElfBinary64 ()
- BinaryFormat getFormat ()

6.9.1 Detailed Description

Definition at line 92 of file ElfBinary.hpp.

6.9.2 Constructor & Destructor Documentation

6.9.2.1 EPAX::Elf::ElfBinary64::ElfBinary64 (std::string *n*)

Definition at line 130 of file ElfBinary.cpp.

6.9.2.2 virtual EPAX::Elf::ElfBinary64::~ElfBinary64 ( ) [inline, virtual]

Definition at line 95 of file ElfBinary.hpp.

6.9.3 Member Function Documentation

6.9.3.1 BinaryFormat EPAX::Elf::ElfBinary64::getFormat ( ) [inline, virtual]

Implements EPAX::Elf::ElfBinary.

Definition at line 97 of file ElfBinary.hpp.

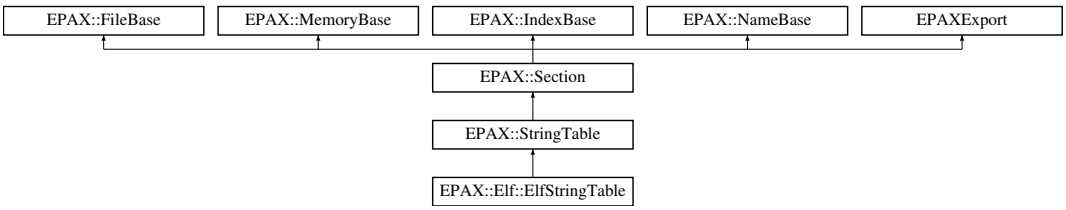
The documentation for this class was generated from the following files:

- ElfBinary.hpp
- ElfBinary.cpp

6.10 EPAX::Elf::ElfStringTable Class Reference

`#include <ElfBinary.hpp>`

Inheritance diagram for EPAX::Elf::ElfStringTable:



Public Member Functions

- ElfStringTable (BaseBinary *b, uint64_t o, uint64_t fs, uint64_t ma, uint64_t ms, uint32_t i, std::string n)
- ~ElfStringTable ()
- char * getStringAt (uint32_t i)
- void print (std::ostream &stream=std::cout)

6.10.1 Detailed Description

Definition at line 223 of file ElfBinary.hpp.

6.10.2 Constructor & Destructor Documentation

6.10.2.1 EPAX::Elf::ElfStringTable::ElfStringTable (BaseBinary * *b*, uint64_t *o*, uint64_t *fs*, uint64_t *ma*, uint64_t *ms*, uint32_t *i*, std::string *n*)

Definition at line 629 of file ElfBinary.cpp.

6.10.2.2 EPAX::Elf::ElfStringTable::~ElfStringTable ( )

Definition at line 638 of file ElfBinary.cpp.

6.10.3 Member Function Documentation

6.10.3.1 char * EPAX::Elf::ElfStringTable::getStringAt (uint32_t *i*) [virtual]

Implements EPAX::StringTable.

Definition at line 644 of file ElfBinary.cpp.

6.10.3.2 void EPAX::Elf::ElfStringTable::print (std::ostream & *stream* = std::cout) [virtual]

Reimplemented from EPAX::Section.

Definition at line 508 of file ElfBinary.cpp.

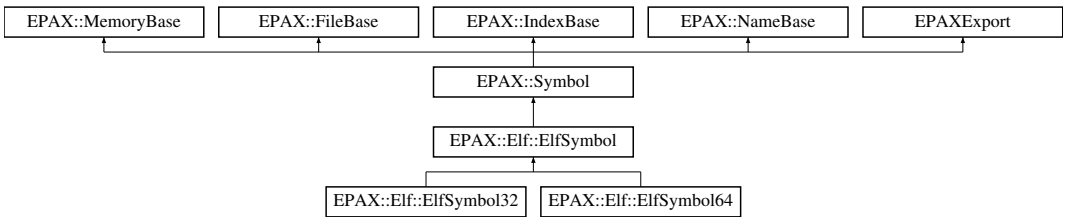
The documentation for this class was generated from the following files:

- ElfBinary.hpp
- ElfBinary.cpp

6.11 EPAX::Elf::ElfSymbol Class Reference

`#include <ElfBinary.hpp>`

Inheritance diagram for EPAX::Elf::ElfSymbol:



Public Member Functions

- ElfSymbol (BaseBinary *b, uint64_t o, uint64_t s, uint32_t i)
- virtual ~ElfSymbol ()
- void print (std::ostream &stream=std::cout)
- virtual uint64_t getNameIndex ()=0
- virtual uint64_t getValue ()=0
- virtual uint32_t getSection ()=0
- virtual uint32_t getSize ()=0
- virtual uint32_t getType ()=0
- virtual uint32_t getBinding ()=0
- virtual uint64_t getVisibility ()=0

- bool isFunction ()
- bool isThumbFunction ()
- uint64_t getFunctionAddress ()

Protected Attributes

- rawbyte_t * entry

6.11.1 Detailed Description

Definition at line 170 of file ElfBinary.hpp.

6.11.2 Constructor & Destructor Documentation

6.11.2.1 EPAX::Elf::ElfSymbol::ElfSymbol (BaseBinary * *b*, uint64_t *o*, uint64_t *s*, uint32_t *i*)

Definition at line 496 of file ElfBinary.cpp.

6.11.2.2 EPAX::Elf::ElfSymbol::~ElfSymbol ( ) [virtual]

Definition at line 502 of file ElfBinary.cpp.

6.11.3 Member Function Documentation

6.11.3.1 virtual uint32_t EPAX::Elf::ElfSymbol::getBinding ( ) [pure virtual]

Implemented in EPAX::Elf::ElfSymbol64, and EPAX::Elf::ElfSymbol32.

6.11.3.2 uint64_t EPAX::Elf::ElfSymbol::getFunctionAddress ( )

Definition at line 620 of file ElfBinary.cpp.

6.11.3.3 virtual uint64_t EPAX::Elf::ElfSymbol::getNameIndex ( ) [pure virtual]

Implemented in EPAX::Elf::ElfSymbol64, and EPAX::Elf::ElfSymbol32.

6.11.3.4 virtual uint32_t EPAX::Elf::ElfSymbol::getSection ( ) [pure virtual]

Implemented in EPAX::Elf::ElfSymbol64, and EPAX::Elf::ElfSymbol32.

6.11.3.5 virtual uint32_t EPAX::Elf::ElfSymbol::getSize ( ) [pure virtual]

Implemented in EPAX::Elf::ElfSymbol64, and EPAX::Elf::ElfSymbol32.

6.11.3.6 virtual uint32_t EPAX::Elf::ElfSymbol::getType ( ) [pure virtual]

Implemented in EPAX::Elf::ElfSymbol64, and EPAX::Elf::ElfSymbol32.

6.11.3.7 virtual uint64_t EPAX::Elf::ElfSymbol::getValue ( ) [pure virtual]

Implemented in EPAX::Elf::ElfSymbol64, and EPAX::Elf::ElfSymbol32.

6.11.3.8 virtual uint64_t EPAX::Elf::ElfSymbol::getVisibility ( ) [pure virtual]

Implemented in EPAX::Elf::ElfSymbol64, and EPAX::Elf::ElfSymbol32.

6.11.3.9 bool EPAX::Elf::ElfSymbol::isFunction ( ) [virtual]

Implements EPAX::Symbol.

Definition at line 612 of file ElfBinary.cpp.

6.11.3.10 bool EPAX::Elf::ElfSymbol::isThumbFunction ( ) [virtual]

Implements EPAX::Symbol.

Definition at line 616 of file ElfBinary.cpp.

6.11.3.11 void EPAX::Elf::ElfSymbol::print (std::ostream & *stream* = std::cout)

Definition at line 526 of file ElfBinary.cpp.

6.11.4 Member Data Documentation

6.11.4.1 rawbyte_t* EPAX::Elf::ElfSymbol::entry [protected]

Definition at line 172 of file ElfBinary.hpp.

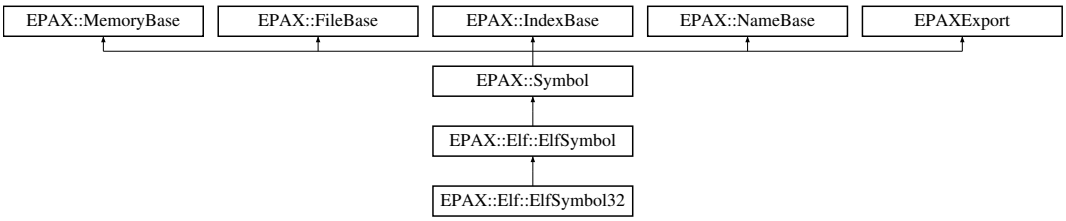
The documentation for this class was generated from the following files:

- ElfBinary.hpp
- ElfBinary.cpp

6.12 EPAX::Elf::ElfSymbol32 Class Reference

`#include <ElfBinary.hpp>`

Inheritance diagram for EPAX::Elf::ElfSymbol32:



Public Member Functions

- ElfSymbol32 (BaseBinary ∗b, uint64_t o, uint32_t i)
- virtual ∼ElfSymbol32 ()
- uint64_t getNameIndex ()
- uint64_t getValue ()
- uint32_t getSection ()
- uint32_t getSize ()
- uint32_t getType ()
- uint32_t getBinding ()
- uint64_t getVisibility ()

6.12.1 Detailed Description

Definition at line 193 of file ElfBinary.hpp.

6.12.2 Constructor & Destructor Documentation

6.12.2.1 EPAX::Elf::ElfSymbol32::ElfSymbol32 (BaseBinary ∗ *b*, uint64_t *o*, uint32_t *i*)

Definition at line 542 of file ElfBinary.cpp.

6.12.2.2 virtual EPAX::Elf::ElfSymbol32::∼ElfSymbol32 ( ) [inline, virtual]

Definition at line 196 of file ElfBinary.hpp.

6.12.3 Member Function Documentation

6.12.3.1 uint32_t EPAX::Elf::ElfSymbol32::getBinding ( ) [virtual]

Implements EPAX::Elf::ElfSymbol.

Definition at line 596 of file ElfBinary.cpp.

6.12.3.2 uint64_t EPAX::Elf::ElfSymbol32::getNameIndex ( ) [virtual]

Implements EPAX::Elf::ElfSymbol.

Definition at line 556 of file ElfBinary.cpp.

6.12.3.3 uint32_t EPAX::Elf::ElfSymbol32::getSection ( ) [virtual]

Implements EPAX::Elf::ElfSymbol.

Definition at line 580 of file ElfBinary.cpp.

6.12.3.4 uint32_t EPAX::Elf::ElfSymbol32::getSize ( ) [virtual]

Implements EPAX::Elf::ElfSymbol.

Definition at line 572 of file ElfBinary.cpp.

6.12.3.5 uint32_t EPAX::Elf::ElfSymbol32::getType ( ) [virtual]

Implements EPAX::Elf::ElfSymbol.

Definition at line 588 of file ElfBinary.cpp.

6.12.3.6 uint64_t EPAX::Elf::ElfSymbol32::getValue ( ) [virtual]

Implements EPAX::Elf::ElfSymbol.

Definition at line 564 of file ElfBinary.cpp.

6.12.3.7 uint64_t EPAX::Elf::ElfSymbol32::getVisibility ( ) [virtual]

Implements EPAX::Elf::ElfSymbol.

Definition at line 604 of file ElfBinary.cpp.

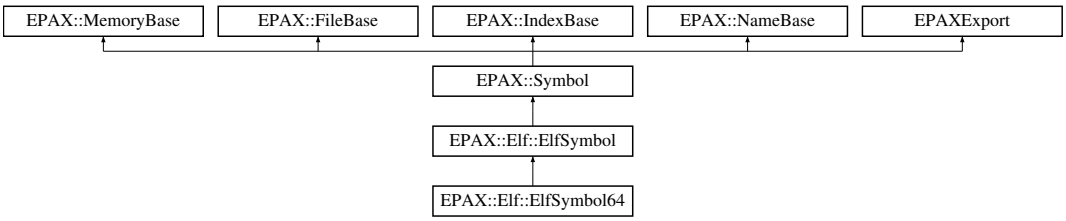
The documentation for this class was generated from the following files:

- ElfBinary.hpp
- ElfBinary.cpp

6.13 EPAX::Elf::ElfSymbol64 Class Reference

`#include <ElfBinary.hpp>`

Inheritance diagram for EPAX::Elf::ElfSymbol64:



Public Member Functions

- ElfSymbol64 (BaseBinary *b, uint64_t o, uint32_t i)
- virtual ~ElfSymbol64 ()
- uint64_t getNameIndex ()
- uint64_t getValue ()
- uint32_t getSection ()
- uint32_t getSize ()
- uint32_t getType ()
- uint32_t getBinding ()
- uint64_t getVisibility ()

6.13.1 Detailed Description

Definition at line 208 of file ElfBinary.hpp.

6.13.2 Constructor & Destructor Documentation

6.13.2.1 EPAX::Elf::ElfSymbol64::ElfSymbol64 (BaseBinary * *b*, uint64_t *o*, uint32_t *i*)

Definition at line 549 of file ElfBinary.cpp.

6.13.2.2 virtual EPAX::Elf::ElfSymbol64::~ElfSymbol64 ( ) [inline, virtual]

Definition at line 211 of file ElfBinary.hpp.

6.13.3 Member Function Documentation

6.13.3.1 uint32_t EPAX::Elf::ElfSymbol64::getBinding ( ) [virtual]

Implements EPAX::Elf::ElfSymbol.

Definition at line 600 of file ElfBinary.cpp.

6.13.3.2 uint64_t EPAX::Elf::ElfSymbol64::getNameIndex ( ) [virtual]

Implements EPAX::Elf::ElfSymbol.

Definition at line 560 of file ElfBinary.cpp.

6.13.3.3 uint32_t EPAX::Elf::ElfSymbol64::getSection ( ) [virtual]

Implements EPAX::Elf::ElfSymbol.

Definition at line 584 of file ElfBinary.cpp.

6.13.3.4 uint32_t EPAX::Elf::ElfSymbol64::getSize ( ) [virtual]

Implements EPAX::Elf::ElfSymbol.

Definition at line 576 of file ElfBinary.cpp.

6.13.3.5 uint32_t EPAX::Elf::ElfSymbol64::getType ( ) [virtual]

Implements EPAX::Elf::ElfSymbol.

Definition at line 592 of file ElfBinary.cpp.

6.13.3.6 uint64_t EPAX::Elf::ElfSymbol64::getValue ( ) [virtual]

Implements EPAX::Elf::ElfSymbol.

Definition at line 568 of file ElfBinary.cpp.

6.13.3.7 uint64_t EPAX::Elf::ElfSymbol64::getVisibility ( ) [virtual]

Implements EPAX::Elf::ElfSymbol.

Definition at line 608 of file ElfBinary.cpp.

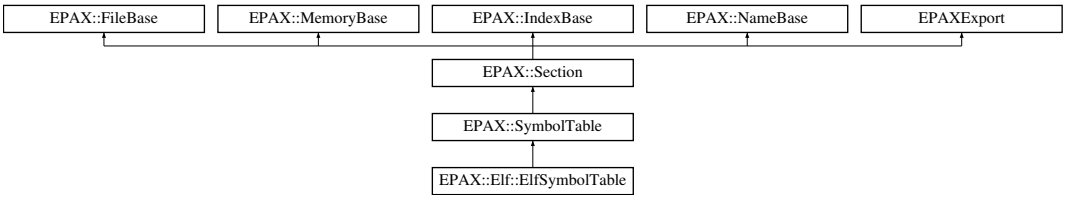
The documentation for this class was generated from the following files:

- ElfBinary.hpp
- ElfBinary.cpp

6.14 EPAX::Elf::ElfSymbolTable Class Reference

`#include <ElfBinary.hpp>`

Inheritance diagram for EPAX::Elf::ElfSymbolTable:



Public Member Functions

- ElfSymbolTable (BaseBinary ∗b, uint64_t o, uint64_t fs, uint64_t ma, uint64_t ms, uint32_t i, std::string n, ElfStringTable ∗st)
- ∼ElfSymbolTable ()
- void print (std::ostream &stream=std::cout)

6.14.1 Detailed Description

Definition at line 236 of file ElfBinary.hpp.

6.14.2 Constructor & Destructor Documentation

6.14.2.1 EPAX::Elf::ElfSymbolTable::ElfSymbolTable (BaseBinary ∗ *b,* uint64_t *o,* uint64_t *fs,* uint64_t *ma,* uint64_t *ms,* uint32_t *i,* std::string *n,* ElfStringTable ∗ *st*)

Definition at line 649 of file ElfBinary.cpp.

6.14.2.2 EPAX::Elf::ElfSymbolTable::∼ElfSymbolTable ( ) [inline]

Definition at line 242 of file ElfBinary.hpp.

6.14.3 Member Function Documentation

6.14.3.1 void EPAX::Elf::ElfSymbolTable::print (std::ostream & *stream* = `std::cout`) [virtual]

Implements EPAX::SymbolTable.

Definition at line 518 of file ElfBinary.cpp.

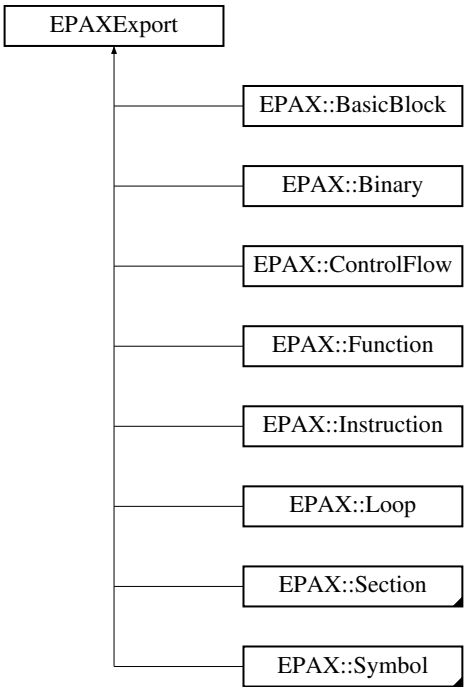
The documentation for this class was generated from the following files:

- ElfBinary.hpp
- ElfBinary.cpp

6.15 EPAXExport Class Reference

`#include <EPAXCommonInternal.hpp>`

Inheritance diagram for EPAXExport:



Public Member Functions

- EPAXExport (EPAXExportClass cls)
- virtual ~EPAXExport ()
- EPAXExportClass getClass ()

Protected Attributes

- EPAXExportClass expclass

6.15.1 Detailed Description

Definition at line 112 of file EPAXCommonInternal.hpp.

6.15.2 Constructor & Destructor Documentation

6.15.2.1 EPAXExport::EPAXExport (EPAXExportClass *cls*) [inline]

Definition at line 116 of file EPAXCommonInternal.hpp.

6.15.2.2 virtual EPAXExport::~EPAXExport ( ) [inline, virtual]

Definition at line 117 of file EPAXCommonInternal.hpp.

6.15.3 Member Function Documentation

6.15.3.1 EPAXExportClass EPAXExport::getClass ( ) [inline]

Definition at line 119 of file EPAXCommonInternal.hpp.

6.15.4 Member Data Documentation

6.15.4.1 EPAXExportClass EPAXExport::expclass [protected]

Definition at line 114 of file EPAXCommonInternal.hpp.

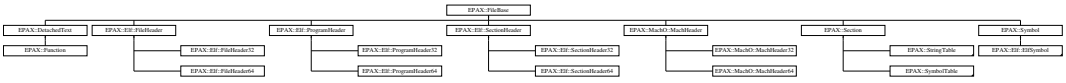
The documentation for this class was generated from the following file:

- EPAXCommonInternal.hpp

6.16 EPAX::FileBase Class Reference

`#include <BaseClass.hpp>`

Inheritance diagram for EPAX::FileBase:



Public Member Functions

- FileBase (BaseBinary *b, uint64_t o, uint64_t s)
- virtual ~FileBase ()
- BaseBinary * getBinary ()
- InputFile * getInputFile ()
- bool is32Bit ()
- uint64_t getFileOffset ()

- uint64_t getFileSize ()
- void setFileSize (uint64_t s)

6.16.1 Detailed Description

Definition at line 42 of file BaseClass.hpp.

6.16.2 Constructor & Destructor Documentation

6.16.2.1 EPAX::FileBase::FileBase (BaseBinary * *b*, uint64_t *o*, uint64_t *s*) [inline]

Definition at line 49 of file BaseClass.hpp.

6.16.2.2 virtual EPAX::FileBase::~FileBase ( ) [inline, virtual]

Definition at line 51 of file BaseClass.hpp.

6.16.3 Member Function Documentation

6.16.3.1 BaseBinary* EPAX::FileBase::getBinary ( ) [inline]

Definition at line 53 of file BaseClass.hpp.

6.16.3.2 uint64_t EPAX::FileBase::getFileOffset ( ) [inline]

Reimplemented in EPAX::Elf::SectionHeader64, EPAX::Elf::SectionHeader32, and EPAX::Elf::SectionHeader.

Definition at line 56 of file BaseClass.hpp.

6.16.3.3 uint64_t EPAX::FileBase::getFileSize ( ) [inline]

Definition at line 57 of file BaseClass.hpp.

6.16.3.4 InputFile * EPAX::FileBase::getInputFile ( )

Definition at line 46 of file BaseClass.cpp.

6.16.3.5 bool EPAX::FileBase::is32Bit ( )

Definition at line 50 of file BaseClass.cpp.

6.16.3.6 void EPAX::FileBase::setFileSize (uint64_t *s*) [inline]

Definition at line 58 of file BaseClass.hpp.

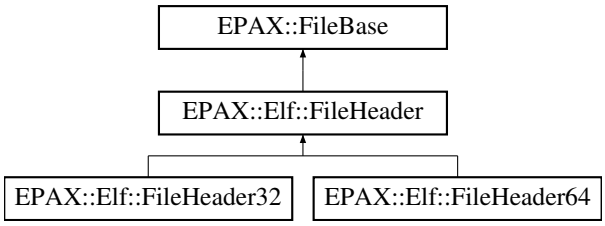
The documentation for this class was generated from the following files:

- BaseClass.hpp
- BaseClass.cpp

6.17 EPAX::Elf::FileHeader Class Reference

`#include <ElfBinary.hpp>`

Inheritance diagram for EPAX::Elf::FileHeader:



Public Member Functions

- FileHeader (BaseBinary ∗b, uint64_t o, uint64_t s)
- virtual ∼FileHeader ()
- virtual uint64_t getStartAddr ()=0
- virtual bool verify ()=0
- virtual bool isARM ()=0
- virtual void describe ()=0
- virtual uint32_t getSectionCount ()=0
- virtual uint64_t getSecTableOffset ()=0
- virtual uint32_t getShdrSize ()=0
- virtual uint32_t getShdrStringIndex ()=0
- virtual uint32_t getSegmentCount ()=0
- virtual uint64_t getSegTableOffset ()=0
- virtual uint32_t getPhdrSize ()=0
- virtual uint32_t getFileType ()=0

Static Protected Member Functions

- static void describeISA (uint32_t ctype)

Protected Attributes

- rawbyte_t * entry

6.17.1 Detailed Description

Definition at line 101 of file ElfBinary.hpp.

6.17.2 Constructor & Destructor Documentation

6.17.2.1 EPAX::Elf::FileHeader::FileHeader (BaseBinary * *b*, uint64_t *o*, uint64_t *s*)

Definition at line 321 of file ElfBinary.cpp.

6.17.2.2 EPAX::Elf::FileHeader::~FileHeader ( ) [virtual]

Definition at line 327 of file ElfBinary.cpp.

6.17.3 Member Function Documentation

6.17.3.1 void EPAX::Elf::FileHeader::describe ( ) [pure virtual]

Implemented in EPAX::Elf::FileHeader64, and EPAX::Elf::FileHeader32.

Definition at line 462 of file ElfBinary.cpp.

6.17.3.2 void EPAX::Elf::FileHeader::describeISA (uint32_t *ctype*) [static, protected]

Definition at line 466 of file ElfBinary.cpp.

6.17.3.3 virtual uint32_t EPAX::Elf::FileHeader::getFileType ( ) [pure virtual]

Implemented in EPAX::Elf::FileHeader64, and EPAX::Elf::FileHeader32.

6.17.3.4 virtual uint32_t EPAX::Elf::FileHeader::getPhdrSize ( ) [pure virtual]

Implemented in EPAX::Elf::FileHeader64, and EPAX::Elf::FileHeader32.

6.17.3.5 virtual uint64_t EPAX::Elf::FileHeader::getSecTableOffset ( ) [pure virtual]

Implemented in EPAX::Elf::FileHeader64, and EPAX::Elf::FileHeader32.

6.17.3.6 virtual uint32_t EPAX::Elf::FileHeader::getSectionCount ( ) [pure virtual]

Implemented in EPAX::Elf::FileHeader64, and EPAX::Elf::FileHeader32.

6.17.3.7 virtual uint32_t EPAX::Elf::FileHeader::getSegmentCount ( ) [pure virtual]

Implemented in EPAX::Elf::FileHeader64, and EPAX::Elf::FileHeader32.

6.17.3.8 virtual uint64_t EPAX::Elf::FileHeader::getSegTableOffset ( ) [pure virtual]

Implemented in EPAX::Elf::FileHeader64, and EPAX::Elf::FileHeader32.

6.17.3.9 virtual uint32_t EPAX::Elf::FileHeader::getShdrSize ( ) [pure virtual]

Implemented in EPAX::Elf::FileHeader64, and EPAX::Elf::FileHeader32.

6.17.3.10 virtual uint32_t EPAX::Elf::FileHeader::getShdrStringIndex ( ) [pure virtual]

Implemented in EPAX::Elf::FileHeader64, and EPAX::Elf::FileHeader32.

6.17.3.11 virtual uint64_t EPAX::Elf::FileHeader::getStartAddr ( ) [pure virtual]

Implemented in EPAX::Elf::FileHeader64, and EPAX::Elf::FileHeader32.

6.17.3.12 virtual bool EPAX::Elf::FileHeader::isARM ( ) [pure virtual]

Implemented in EPAX::Elf::FileHeader64, and EPAX::Elf::FileHeader32.

6.17.3.13 virtual bool EPAX::Elf::FileHeader::verify ( ) [pure virtual]

Implemented in EPAX::Elf::FileHeader64, and EPAX::Elf::FileHeader32.

6.17.4 Member Data Documentation

6.17.4.1 rawbyte_t* EPAX::Elf::FileHeader::entry [protected]

Definition at line 103 of file ElfBinary.hpp.

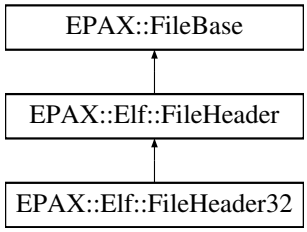
The documentation for this class was generated from the following files:

- ElfBinary.hpp
- ElfBinary.cpp

6.18 EPAX::Elf::FileHeader32 Class Reference

`#include <ElfBinary.hpp>`

Inheritance diagram for EPAX::Elf::FileHeader32:



Public Member Functions

- FileHeader32 (BaseBinary *b, uint64_t o)
- virtual ~FileHeader32 ()
- uint64_t getStartAddr ()
- bool verify ()
- bool isARM ()
- void describe ()
- uint32_t getSectionCount ()
- uint64_t getSecTableOffset ()
- uint32_t getShdrSize ()
- uint32_t getShdrStringIndex ()
- uint32_t getSegmentCount ()
- uint64_t getSegTableOffset ()
- uint32_t getPhdrSize ()
- uint32_t getFileType ()

6.18.1 Detailed Description

Definition at line 127 of file ElfBinary.hpp.

6.18.2 Constructor & Destructor Documentation

6.18.2.1 EPAX::Elf::FileHeader32::FileHeader32 (BaseBinary * *b*, uint64_t *o*)

Definition at line 397 of file ElfBinary.cpp.

6.18.2.2 virtual EPAX::Elf::FileHeader32::~FileHeader32 ( ) [inline, virtual]

Definition at line 131 of file ElfBinary.hpp.

6.18.3 Member Function Documentation

6.18.3.1 void EPAX::Elf::FileHeader32::describe ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 484 of file ElfBinary.cpp.

6.18.3.2 uint32_t EPAX::Elf::FileHeader32::getFileType ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 389 of file ElfBinary.cpp.

6.18.3.3 uint32_t EPAX::Elf::FileHeader32::getPhdrSize ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 357 of file ElfBinary.cpp.

6.18.3.4 uint64_t EPAX::Elf::FileHeader32::getSecTableOffset ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 333 of file ElfBinary.cpp.

6.18.3.5 uint32_t EPAX::Elf::FileHeader32::getSectionCount ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 373 of file ElfBinary.cpp.

6.18.3.6 uint32_t EPAX::Elf::FileHeader32::getSegmentCount ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 381 of file ElfBinary.cpp.

6.18.3.7 uint64_t EPAX::Elf::FileHeader32::getSegTableOffset ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 349 of file ElfBinary.cpp.

6.18.3.8 uint32_t EPAX::Elf::FileHeader32::getShdrSize ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 341 of file ElfBinary.cpp.

6.18.3.9 uint32_t EPAX::Elf::FileHeader32::getShdrStringIndex ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 365 of file ElfBinary.cpp.

6.18.3.10 uint64_t EPAX::Elf::FileHeader32::getStartAddr ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 411 of file ElfBinary.cpp.

6.18.3.11 bool EPAX::Elf::FileHeader32::isARM ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 447 of file ElfBinary.cpp.

6.18.3.12 bool EPAX::Elf::FileHeader32::verify ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 419 of file ElfBinary.cpp.

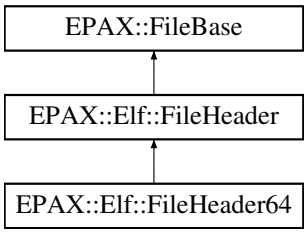
The documentation for this class was generated from the following files:

- ElfBinary.hpp
- ElfBinary.cpp

6.19 EPAX::Elf::FileHeader64 Class Reference

```
#include <ElfBinary.hpp>
```

Inheritance diagram for EPAX::Elf::FileHeader64:



Public Member Functions

- FileHeader64 (BaseBinary *b, uint64_t o)
- virtual ~FileHeader64 ()
- uint64_t getStartAddr ()
- bool verify ()
- bool isARM ()
- void describe ()
- uint32_t getSectionCount ()
- uint64_t getSecTableOffset ()
- uint32_t getShdrSize ()
- uint32_t getShdrStringIndex ()
- uint32_t getSegmentCount ()
- uint64_t getSegTableOffset ()
- uint32_t getPhdrSize ()
- uint32_t getFileType ()

6.19.1 Detailed Description

Definition at line 149 of file ElfBinary.hpp.

6.19.2 Constructor & Destructor Documentation

6.19.2.1 EPAX::Elf::FileHeader64::FileHeader64 (BaseBinary * *b*, uint64_t *o*)

Definition at line 404 of file ElfBinary.cpp.

6.19.2.2 virtual EPAX::Elf::FileHeader64::~FileHeader64 ( ) [inline, virtual]

Definition at line 152 of file ElfBinary.hpp.

6.19.3 Member Function Documentation

6.19.3.1 void EPAX::Elf::FileHeader64::describe ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 490 of file ElfBinary.cpp.

6.19.3.2 uint32_t EPAX::Elf::FileHeader64::getFileType ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 393 of file ElfBinary.cpp.

6.19.3.3 uint32_t EPAX::Elf::FileHeader64::getPhdrSize ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 361 of file ElfBinary.cpp.

6.19.3.4 uint64_t EPAX::Elf::FileHeader64::getSecTableOffset ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 337 of file ElfBinary.cpp.

6.19.3.5 uint32_t EPAX::Elf::FileHeader64::getSectionCount ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 377 of file ElfBinary.cpp.

6.19.3.6 uint32_t EPAX::Elf::FileHeader64::getSegmentCount ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 385 of file ElfBinary.cpp.

6.19.3.7 uint64_t EPAX::Elf::FileHeader64::getSegTableOffset ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 353 of file ElfBinary.cpp.

6.19.3.8 uint32_t EPAX::Elf::FileHeader64::getShdrSize ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 345 of file ElfBinary.cpp.

6.19.3.9 uint32_t EPAX::Elf::FileHeader64::getShdrStringIndex ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 369 of file ElfBinary.cpp.

6.19.3.10 uint64_t EPAX::Elf::FileHeader64::getStartAddr ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 415 of file ElfBinary.cpp.

6.19.3.11 bool EPAX::Elf::FileHeader64::isARM ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 453 of file ElfBinary.cpp.

6.19.3.12 bool EPAX::Elf::FileHeader64::verify ( ) [virtual]

Implements EPAX::Elf::FileHeader.

Definition at line 433 of file ElfBinary.cpp.

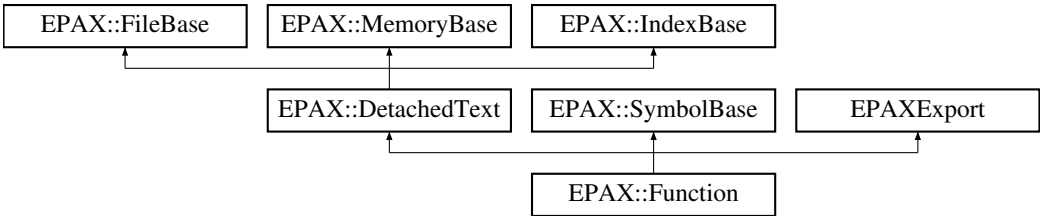
The documentation for this class was generated from the following files:

- ElfBinary.hpp
- ElfBinary.cpp

6.20 EPAX::Function Class Reference

`#include <Function.hpp>`

Inheritance diagram for EPAX::Function:



Public Member Functions

- Function (BaseBinary *b, uint64_t o, uint64_t s, uint64_t a, uint32_t i, Symbol *y)
- virtual ~Function ()
- void print (std::ostream &stream=std::cout)
- ControlFlow * getControlFlow ()
- uint32_t countBasicBlocks ()
- BasicBlock * findBasicBlock (uint64_t addr)
- BasicBlock * getBasicBlock (uint32_t idx)
- uint32_t countInstructions ()
- Instruction * findInstruction (uint64_t addr)
- Instruction * getInstruction (uint32_t idx)
- void disassemble ()

Static Public Member Functions

- static void printHeader (std::ostream &stream=std::cout)

6.20.1 Detailed Description

Definition at line 55 of file Function.hpp.

6.20.2 Constructor & Destructor Documentation

6.20.2.1 EPAX::Function::Function (BaseBinary * *b*, uint64_t *o*, uint64_t *s*, uint64_t *a*, uint32_t *i*, Symbol * *y*)

Definition at line 154 of file Function.cpp.

6.20.2.2 EPAX::Function::~Function ( ) [virtual]

Definition at line 166 of file Function.cpp.

6.20.3 Member Function Documentation

6.20.3.1 uint32_t EPAX::Function::countBasicBlocks ( )

Definition at line 198 of file Function.cpp.

6.20.3.2 uint32_t EPAX::Function::countInstructions ( )

Definition at line 220 of file Function.cpp.

6.20.3.3 void EPAX::Function::disassemble ( )

Definition at line 172 of file Function.cpp.

6.20.3.4 BasicBlock * EPAX::Function::findBasicBlock (uint64_t *addr*)

Definition at line 205 of file Function.cpp.

6.20.3.5 Instruction * EPAX::Function::findInstruction (uint64_t *addr*)

Definition at line 227 of file Function.cpp.

6.20.3.6 BasicBlock * EPAX::Function::getBasicBlock (uint32_t *idx*)

Definition at line 213 of file Function.cpp.

6.20.3.7 ControlFlow* EPAX::Function::getControlFlow ( ) [inline]

Definition at line 68 of file Function.hpp.

6.20.3.8 Instruction * EPAX::Function::getInstruction (uint32_t *idx*)

Definition at line 234 of file Function.cpp.

6.20.3.9 void EPAX::Function::print (std::ostream & *stream* = `std::cout`) [virtual]

Reimplemented from EPAX::DetachedText.

Definition at line 186 of file Function.cpp.

6.20.3.10 void EPAX::Function::printHeader (std::ostream & *stream* = `std::cout`) [static]

Definition at line 178 of file Function.cpp.

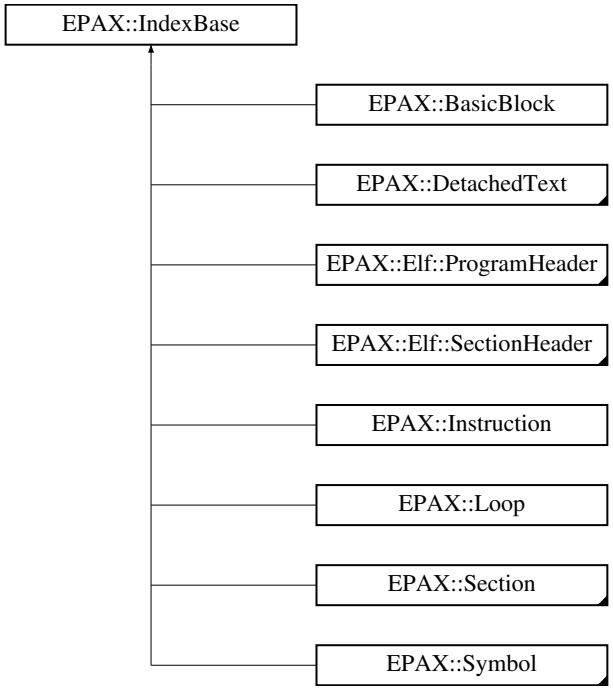
The documentation for this class was generated from the following files:

- Function.hpp
- Function.cpp

6.21 EPAX::IndexBase Class Reference

`#include <BaseClass.hpp>`

Inheritance diagram for EPAX::IndexBase:



Public Member Functions

- IndexBase (uint32_t i)
- virtual ~IndexBase ()
- uint32_t getIndex ()
- void setIndex (uint32_t i)

6.21.1 Detailed Description

Definition at line 94 of file BaseClass.hpp.

6.21.2 Constructor & Destructor Documentation

6.21.2.1 EPAX::IndexBase::IndexBase (uint32_t *i*) [inline]

Definition at line 99 of file BaseClass.hpp.

6.21.2.2 virtual EPAX::IndexBase::~IndexBase ( ) [inline, virtual]

Definition at line 101 of file BaseClass.hpp.

6.21.3 Member Function Documentation

6.21.3.1 uint32_t EPAX::IndexBase::getIndex ( ) [inline]

Definition at line 103 of file BaseClass.hpp.

6.21.3.2 void EPAX::IndexBase::setIndex (uint32_t *i*) [inline]

Definition at line 104 of file BaseClass.hpp.

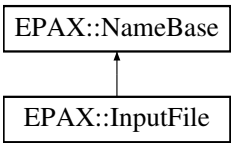
The documentation for this class was generated from the following file:

- BaseClass.hpp

6.22 EPAX::InputFile Class Reference

`#include <InputFile.hpp>`

Inheritance diagram for EPAX::InputFile:



Public Member Functions

- InputFile (std::string n)
- virtual ~InputFile ()
- uint64_t getBytes (uint64_t offset, uint64_t size, rawbyte_t *buffer)
- uint64_t getFileSize ()

6.22.1 Detailed Description

Definition at line 32 of file InputFile.hpp.

6.22.2 Constructor & Destructor Documentation

6.22.2.1 EPAX::InputFile::InputFile (std::string *n*)

Definition at line 30 of file InputFile.cpp.

6.22.2.2 EPAX::InputFile::~InputFile ( ) [virtual]

Definition at line 38 of file InputFile.cpp.

6.22.3 Member Function Documentation

6.22.3.1 uint64_t EPAX::InputFile::getBytes (uint64_t *offset,* uint64_t *size,* rawbyte_t * *buffer*)

Definition at line 54 of file InputFile.cpp.

6.22.3.2 uint64_t EPAX::InputFile::getFileSize ( )

Definition at line 44 of file InputFile.cpp.

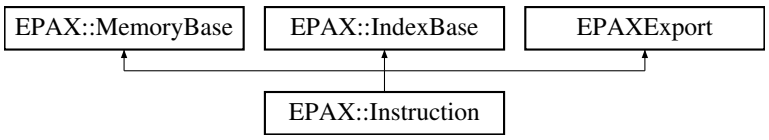
The documentation for this class was generated from the following files:

- InputFile.hpp
- InputFile.cpp

6.23 EPAX::Instruction Class Reference

`#include <Instruction.hpp>`

Inheritance diagram for EPAX::Instruction:



Public Member Functions

- Instruction (const uint64_t a, const rawbyte_t *r, Function *func, const darm_-mode_t mode)
- virtual ~Instruction ()
- darm_t * source ()

- void setBasicBlock (BasicBlock *bb)
- BasicBlock * getBasicBlock ()
- darm_cond_t getCondition ()
- bool isBranch ()
- bool isConditionalBranch ()
- bool isUnconditionalBranch ()
- bool hasFallthrough ()
- bool touchesPC ()
- bool isCall ()
- uint64_t fallthroughTarget ()
- uint64_t getBranchTarget ()
- uint32_t getControlTargets (std::vector< uint64_t > &tgts)
- bool isFpop ()
- bool isLoad ()
- bool isStore ()
- bool isMemop ()
- uint32_t getSourceRegisterSizeInBits ()
- uint32_t getSourceDatatypeSizeInBits ()
- std::string stringRep ()
- void print (std::ostream &stream=std::cout)

6.23.1 Detailed Description

Definition at line 40 of file Instruction.hpp.

6.23.2 Constructor & Destructor Documentation

6.23.2.1 EPAX::Instruction::Instruction (const uint64_t *a*, const rawbyte_t * *r*, Function * *func*, const darm_mode_t *mode*)

Definition at line 38 of file Instruction.cpp.

6.23.2.2 EPAX::Instruction::~Instruction ( ) [virtual]

Definition at line 106 of file Instruction.cpp.

6.23.3 Member Function Documentation

6.23.3.1 uint64_t EPAX::Instruction::fallthroughTarget ( )

Definition at line 257 of file Instruction.cpp.

6.23.3.2 BasicBlock* EPAX::Instruction::getBasicBlock ( ) [inline]

Definition at line 55 of file Instruction.hpp.

6.23.3.3 uint64_t EPAX::Instruction::getBranchTarget ( )

Definition at line 217 of file Instruction.cpp.

6.23.3.4 darm_cond_t EPAX::Instruction::getCondition ( )

Definition at line 117 of file Instruction.cpp.

6.23.3.5 uint32_t EPAX::Instruction::getControlTargets (std::vector< uint64_t > & *tgts*)

Definition at line 262 of file Instruction.cpp.

6.23.3.6 uint32_t EPAX::Instruction::getSourceDatatypeSizeInBits ( )

Definition at line 351 of file Instruction.cpp.

6.23.3.7 uint32_t EPAX::Instruction::getSourceRegisterSizeInBits ( )

Definition at line 335 of file Instruction.cpp.

6.23.3.8 bool EPAX::Instruction::hasFallthrough ( )

Definition at line 244 of file Instruction.cpp.

6.23.3.9 bool EPAX::Instruction::isBranch ( )

Definition at line 175 of file Instruction.cpp.

6.23.3.10 bool EPAX::Instruction::isCall ( )

Definition at line 209 of file Instruction.cpp.

6.23.3.11 bool EPAX::Instruction::isConditionalBranch ( )

Definition at line 179 of file Instruction.cpp.

6.23.3.12 bool EPAX::Instruction::isFpop ( )

Definition at line 273 of file Instruction.cpp.

6.23.3.13 bool EPAX::Instruction::isLoad ( )

Definition at line 293 of file Instruction.cpp.

6.23.3.14 bool EPAX::Instruction::isMemop ( )

Definition at line 331 of file Instruction.cpp.

6.23.3.15 bool EPAX::Instruction::isStore ( )

Definition at line 312 of file Instruction.cpp.

6.23.3.16 bool EPAX::Instruction::isUnconditionalBranch ( )

Definition at line 202 of file Instruction.cpp.

6.23.3.17 void EPAX::Instruction::print (std::ostream & *stream* = `std::cout`)

Definition at line 121 of file Instruction.cpp.

6.23.3.18 void EPAX::Instruction::setBasicBlock (BasicBlock * *bb*) `[inline]`

Definition at line 54 of file Instruction.hpp.

6.23.3.19 darm_t* EPAX::Instruction::source ( ) `[inline]`

Definition at line 52 of file Instruction.hpp.

6.23.3.20 std::string EPAX::Instruction::stringRep ( )

Definition at line 109 of file Instruction.cpp.

6.23.3.21 bool EPAX::Instruction::touchesPC ( )

Definition at line 186 of file Instruction.cpp.

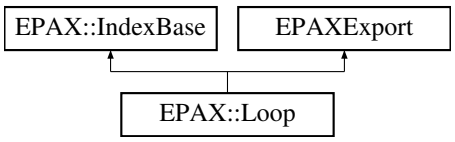
The documentation for this class was generated from the following files:

- Instruction.hpp
- Instruction.cpp

6.24 EPAX::Loop Class Reference

`#include <Loop.hpp>`

Inheritance diagram for EPAX::Loop:



Public Member Functions

- Loop (ControlFlow ∗c, uint32_t h, uint32_t t, uint32_t d, dyn_bitset ∗m, uint32_t i)
- virtual ∼Loop ()
- ControlFlow ∗ getControlFlow ()
- uint32_t countBasicBlocks ()
- uint32_t countInstructions ()
- uint32_t getSize ()
- BasicBlock ∗ findBasicBlock (uint64_t addr)
- bool hasBasicBlock (uint32_t idx)
- BasicBlock ∗ getBasicBlock (uint32_t idx)
- BasicBlock ∗ getNextBasicBlock (uint32_t idx)
- bool isLastBasicBlock (uint32_t idx)
- BasicBlock ∗ head ()
- BasicBlock ∗ tail ()
- Instruction ∗ findInstruction (uint64_t addr)
- Instruction ∗ getInstruction (uint32_t idx)
- bool isLastInstruction (uint32_t idx)
- void setDepth (uint32_t d)
- uint32_t getDepth ()
- bool isChildOf (Loop ∗lp)

6.24.1 Detailed Description

Definition at line 37 of file Loop.hpp.

6.24.2 Constructor & Destructor Documentation

6.24.2.1 EPAX::Loop::Loop (ControlFlow ∗ *c*, uint32_t *h*, uint32_t *t*, uint32_t *d*, dyn_bitset ∗ *m*, uint32_t *i*)

Definition at line 34 of file Loop.cpp.

6.24.2.2 EPAX::Loop::~Loop ( ) [virtual]

Definition at line 50 of file Loop.cpp.

6.24.3 Member Function Documentation

6.24.3.1 uint32_t EPAX::Loop::countBasicBlocks ( )

Definition at line 98 of file Loop.cpp.

6.24.3.2 uint32_t EPAX::Loop::countInstructions ( )

Definition at line 86 of file Loop.cpp.

6.24.3.3 BasicBlock * EPAX::Loop::findBasicBlock (uint64_t *addr*)

Definition at line 120 of file Loop.cpp.

6.24.3.4 Instruction * EPAX::Loop::findInstruction (uint64_t *addr*)

Definition at line 78 of file Loop.cpp.

6.24.3.5 BasicBlock * EPAX::Loop::getBasicBlock (uint32_t *idx*)

Definition at line 139 of file Loop.cpp.

6.24.3.6 ControlFlow* EPAX::Loop::getControlFlow ( ) [inline]

Definition at line 50 of file Loop.hpp.

6.24.3.7 uint32_t EPAX::Loop::getDepth ( )

Definition at line 201 of file Loop.cpp.

6.24.3.8 Instruction* EPAX::Loop::getInstruction (uint32_t *idx*)

6.24.3.9 BasicBlock * EPAX::Loop::getNextBasicBlock (uint32_t *idx*)

Definition at line 149 of file Loop.cpp.

6.24.3.10 uint32_t EPAX::Loop::getSize ( )

Definition at line 108 of file Loop.cpp.

6.24.3.11 bool EPAX::Loop::hasBasicBlock (uint32_t *idx*)

Definition at line 135 of file Loop.cpp.

6.24.3.12 BasicBlock * EPAX::Loop::head ( )

Definition at line 56 of file Loop.cpp.

6.24.3.13 bool EPAX::Loop::isChildOf (Loop * *lp*)

Definition at line 182 of file Loop.cpp.

6.24.3.14 bool EPAX::Loop::isLastBasicBlock (uint32_t *idx*)

Definition at line 165 of file Loop.cpp.

6.24.3.15 bool EPAX::Loop::isLastInstruction (uint32_t *idx*)

6.24.3.16 void EPAX::Loop::setDepth (uint32_t *d*)

Definition at line 197 of file Loop.cpp.

6.24.3.17 BasicBlock * EPAX::Loop::tail ( )

Definition at line 67 of file Loop.cpp.

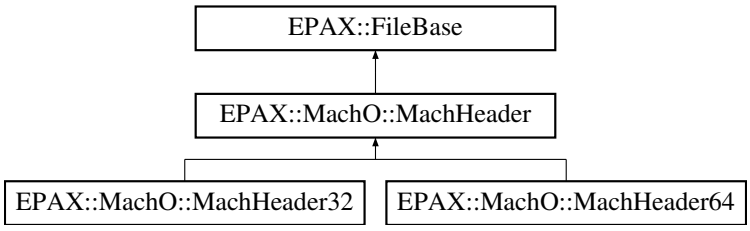
The documentation for this class was generated from the following files:

- Loop.hpp
- Loop.cpp

6.25 EPAX::MachO::MachHeader Class Reference

`#include <MachOBinary.hpp>`

Inheritance diagram for EPAX::MachO::MachHeader:



Public Member Functions

- MachHeader (BaseBinary *b, uint64_t o, uint64_t s)
- virtual ~MachHeader ()
- virtual uint64_t getStartAddr ()=0
- virtual bool verify ()=0
- virtual bool isARM ()=0
- virtual void describe ()=0
- virtual uint32_t getFileType ()=0

Static Protected Member Functions

- static void describeISA (int32_t ctype, int32_t stype)

Protected Attributes

- rawbyte_t * entry

6.25.1 Detailed Description

Definition at line 85 of file MachOBinary.hpp.

6.25.2 Constructor & Destructor Documentation

6.25.2.1 EPAX::MachO::MachHeader::MachHeader (BaseBinary * *b,* uint64_t *o,* uint64_t *s*)

Definition at line 119 of file MachOBinary.cpp.

6.25.2.2 EPAX::MachO::MachHeader::~MachHeader ( ) [virtual]

Definition at line 125 of file MachOBinary.cpp.

6.25.3 Member Function Documentation

6.25.3.1 void EPAX::MachO::MachHeader::describe ( ) [pure virtual]

Implemented in EPAX::MachO::MachHeader64, and EPAX::MachO::MachHeader32.

Definition at line 186 of file MachOBinary.cpp.

6.25.3.2 void EPAX::MachO::MachHeader::describeISA (int32_t *ctype,* int32_t *stype*) [static, protected]

Definition at line 194 of file MachOBinary.cpp.

6.25.3.3 virtual uint32_t EPAX::MachO::MachHeader::getFileType ( ) [pure virtual]

Implemented in EPAX::MachO::MachHeader64, and EPAX::MachO::MachHeader32.

6.25.3.4 virtual uint64_t EPAX::MachO::MachHeader::getStartAddr ( ) [pure virtual]

Implemented in EPAX::MachO::MachHeader64, and EPAX::MachO::MachHeader32.

6.25.3.5 virtual bool EPAX::MachO::MachHeader::isARM ( ) [pure virtual]

Implemented in EPAX::MachO::MachHeader64, and EPAX::MachO::MachHeader32.

6.25.3.6 virtual bool EPAX::MachO::MachHeader::verify ( ) [pure virtual]

Implemented in EPAX::MachO::MachHeader64, and EPAX::MachO::MachHeader32.

6.25.4 Member Data Documentation

6.25.4.1 rawbyte_t* EPAX::MachO::MachHeader::entry [protected]

Definition at line 87 of file MachOBinary.hpp.

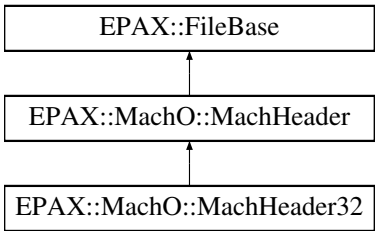
The documentation for this class was generated from the following files:

- MachOBinary.hpp
- MachOBinary.cpp

6.26 EPAX::MachO::MachHeader32 Class Reference

`#include <MachOBinary.hpp>`

Inheritance diagram for EPAX::MachO::MachHeader32:



Public Member Functions

- MachHeader32 (BaseBinary ∗b, uint64_t o)
- virtual ∼MachHeader32 ()
- uint64_t getStartAddr ()
- bool verify ()
- bool isARM ()
- void describe ()
- uint32_t getFileType ()

6.26.1 Detailed Description

Definition at line 102 of file MachOBinary.hpp.

6.26.2 Constructor & Destructor Documentation

6.26.2.1 EPAX::MachO::MachHeader32::MachHeader32 (BaseBinary ∗ *b*, uint64_t *o*)

Definition at line 131 of file MachOBinary.cpp.

6.26.2.2 virtual EPAX::MachO::MachHeader32::∼MachHeader32 ( ) [inline, virtual]

Definition at line 106 of file MachOBinary.hpp.

6.26.3 Member Function Documentation

6.26.3.1 void EPAX::MachO::MachHeader32::describe ( ) [virtual]

Implements EPAX::MachO::MachHeader.

Definition at line 228 of file MachOBinary.cpp.

6.26.3.2 uint32_t EPAX::MachO::MachHeader32::getFileType ( ) [virtual]

Implements EPAX::MachO::MachHeader.

Definition at line 178 of file MachOBinary.cpp.

6.26.3.3 uint64_t EPAX::MachO::MachHeader32::getStartAddr ( ) [virtual]

Implements EPAX::MachO::MachHeader.

Definition at line 145 of file MachOBinary.cpp.

6.26.3.4 bool EPAX::MachO::MachHeader32::isARM ( ) [virtual]

Implements EPAX::MachO::MachHeader.

Definition at line 167 of file MachOBinary.cpp.

6.26.3.5 bool EPAX::MachO::MachHeader32::verify ( ) [virtual]

Implements EPAX::MachO::MachHeader.

Definition at line 153 of file MachOBinary.cpp.

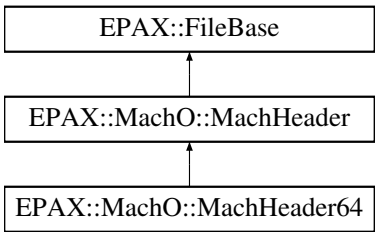
The documentation for this class was generated from the following files:

- MachOBinary.hpp
- MachOBinary.cpp

6.27 EPAX::MachO::MachHeader64 Class Reference

`#include <MachOBinary.hpp>`

Inheritance diagram for EPAX::MachO::MachHeader64:



Public Member Functions

- MachHeader64 (BaseBinary ∗b, uint64_t o)
- virtual ∼MachHeader64 ()
- uint64_t getStartAddr ()
- bool verify ()
- bool isARM ()
- void describe ()
- uint32_t getFileType ()

6.27.1 Detailed Description

Definition at line 115 of file MachOBinary.hpp.

6.27.2 Constructor & Destructor Documentation

6.27.2.1 EPAX::MachO::MachHeader64::MachHeader64 (BaseBinary * *b*, uint64_t *o*)

Definition at line 138 of file MachOBinary.cpp.

6.27.2.2 virtual EPAX::MachO::MachHeader64::~MachHeader64 ( ) [inline, virtual]

Definition at line 118 of file MachOBinary.hpp.

6.27.3 Member Function Documentation

6.27.3.1 void EPAX::MachO::MachHeader64::describe ( ) [virtual]

Implements EPAX::MachO::MachHeader.

Definition at line 234 of file MachOBinary.cpp.

6.27.3.2 uint32_t EPAX::MachO::MachHeader64::getFileType ( ) [virtual]

Implements EPAX::MachO::MachHeader.

Definition at line 182 of file MachOBinary.cpp.

6.27.3.3 uint64_t EPAX::MachO::MachHeader64::getStartAddr ( ) [virtual]

Implements EPAX::MachO::MachHeader.

Definition at line 149 of file MachOBinary.cpp.

6.27.3.4 bool EPAX::MachO::MachHeader64::isARM ( ) [virtual]

Implements EPAX::MachO::MachHeader.

Definition at line 173 of file MachOBinary.cpp.

6.27.3.5 bool EPAX::MachO::MachHeader64::verify ( ) [virtual]

Implements EPAX::MachO::MachHeader.

Definition at line 160 of file MachOBinary.cpp.

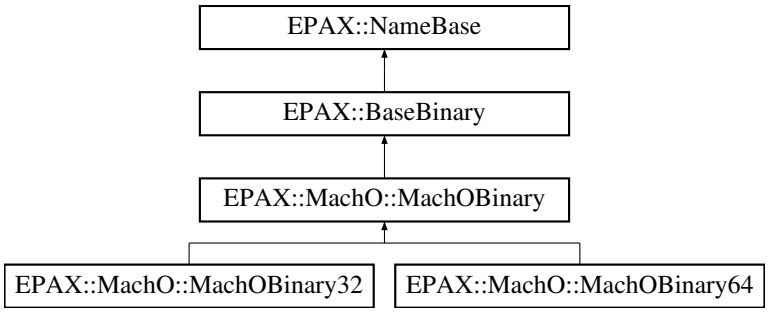
The documentation for this class was generated from the following files:

- MachOBinary.hpp
- MachOBinary.cpp

6.28 EPAX::MachO::MachOBinary Class Reference

`#include <MachOBinary.hpp>`

Inheritance diagram for EPAX::MachO::MachOBinary:



Public Member Functions

- MachOBinary (std::string n)
- virtual ~MachOBinary ()
- virtual BinaryFormat getFormat ()=0
- uint64_t getStartAddr ()
- void emit (std::string n)
- bool verify ()
- bool isARM ()
- void describe ()
- bool is32Bit ()
- bool is64Bit ()
- bool isExecutable ()
- void findFunctions ()
- void findSymbols ()
- void findSections ()
- bool insideTextRange (uint64_t a)
- uint64_t functionEndAddress (Function *f, Function *nextf)
- void printSections (std::ostream &stream=std::cout)
- void printFunctions (std::ostream &stream=std::cout)

Protected Attributes

- MachHeader * machheader

6.28.1 Detailed Description

Definition at line 37 of file MachOBinary.hpp.

6.28.2 Constructor & Destructor Documentation

6.28.2.1 EPAX::MachO::MachOBinary::MachOBinary (std::string *n*)

Definition at line 40 of file MachOBinary.cpp.

6.28.2.2 EPAX::MachO::MachOBinary::~MachOBinary ( ) [virtual]

Definition at line 45 of file MachOBinary.cpp.

6.28.3 Member Function Documentation

6.28.3.1 void EPAX::MachO::MachOBinary::describe ( ) [virtual]

Implements EPAX::BaseBinary.

Definition at line 95 of file MachOBinary.cpp.

6.28.3.2 void EPAX::MachO::MachOBinary::emit (std::string *n*) [virtual]

Implements EPAX::BaseBinary.

Definition at line 83 of file MachOBinary.cpp.

6.28.3.3 void EPAX::MachO::MachOBinary::findFunctions ( ) [virtual]

Finds and internally stores all functions in the image

Returns

none

Implements EPAX::BaseBinary.

Definition at line 99 of file MachOBinary.cpp.

6.28.3.4 void EPAX::MachO::MachOBinary::findSections ( )

Definition at line 107 of file MachOBinary.cpp.

6.28.3.5 void EPAX::MachO::MachOBinary::findSymbols ( ) [virtual]

Finds and internally stores all symbols in the image

Returns

none

Implements EPAX::BaseBinary.

Definition at line 103 of file MachOBinary.cpp.

6.28.3.6 uint64_t EPAX::MachO::MachOBinary::functionEndAddress (Function * *f*, Function * *nextf*) [virtual]

Implements EPAX::BaseBinary.

Definition at line 51 of file MachOBinary.cpp.

6.28.3.7 virtual BinaryFormat EPAX::MachO::MachOBinary::getFormat ( ) [pure virtual]

Implements EPAX::BaseBinary.

Implemented in EPAX::MachO::MachOBinary64, and EPAX::MachO::MachOBinary32.

6.28.3.8 uint64_t EPAX::MachO::MachOBinary::getStartAddr ( ) [virtual]

Implements EPAX::BaseBinary.

Definition at line 67 of file MachOBinary.cpp.

6.28.3.9 bool EPAX::MachO::MachOBinary::insideTextRange (uint64_t *a*) [virtual]

Implements EPAX::BaseBinary.

Definition at line 55 of file MachOBinary.cpp.

6.28.3.10 bool EPAX::MachO::MachOBinary::is32Bit ( ) [virtual]

Implements EPAX::BaseBinary.

Definition at line 59 of file MachOBinary.cpp.

6.28.3.11 bool EPAX::MachO::MachOBinary::is64Bit ( ) [virtual]

Implements EPAX::BaseBinary.

Definition at line 63 of file MachOBinary.cpp.

6.28.3.12 bool EPAX::MachO::MachOBinary::isARM ( ) [virtual]

Implements EPAX::BaseBinary.

Definition at line 91 of file MachOBinary.cpp.

6.28.3.13 bool EPAX::MachO::MachOBinary::isExecutable ( ) [virtual]

Implements EPAX::BaseBinary.

Definition at line 190 of file MachOBinary.cpp.

6.28.3.14 void EPAX::MachO::MachOBinary::printFunctions (std::ostream & *stream* = std::cout) [virtual]

Implements EPAX::BaseBinary.

Definition at line 115 of file MachOBinary.cpp.

6.28.3.15 void EPAX::MachO::MachOBinary::printSections (std::ostream & *stream* = std::cout) [virtual]

Implements EPAX::BaseBinary.

Definition at line 111 of file MachOBinary.cpp.

6.28.3.16 bool EPAX::MachO::MachOBinary::verify ( ) [virtual]

Implements EPAX::BaseBinary.

Definition at line 87 of file MachOBinary.cpp.

6.28.4 Member Data Documentation

6.28.4.1 MachHeader* EPAX::MachO::MachOBinary::machheader [protected]

Definition at line 39 of file MachOBinary.hpp.

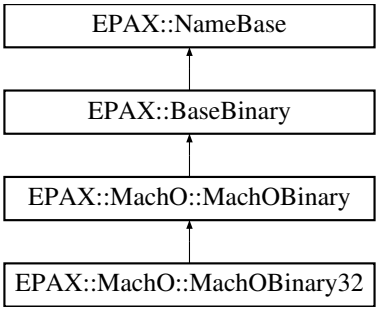
The documentation for this class was generated from the following files:

- MachOBinary.hpp
- MachOBinary.cpp

6.29 EPAX::MachO::MachOBinary32 Class Reference

```
#include <MachOBinary.hpp>
```

Inheritance diagram for EPAX::MachO::MachOBinary32:



Public Member Functions

- MachOBinary32 (std::string n)
- virtual ~MachOBinary32 ()
- BinaryFormat getFormat ()

6.29.1 Detailed Description

Definition at line 69 of file MachOBinary.hpp.

6.29.2 Constructor & Destructor Documentation

6.29.2.1 EPAX::MachO::MachOBinary32::MachOBinary32 (std::string *n*)

Definition at line 71 of file MachOBinary.cpp.

6.29.2.2 virtual EPAX::MachO::MachOBinary32::~MachOBinary32 ( ) [inline, virtual]

Definition at line 72 of file MachOBinary.hpp.

6.29.3 Member Function Documentation

6.29.3.1 BinaryFormat EPAX::MachO::MachOBinary32::getFormat ( ) [inline, virtual]

Implements EPAX::MachO::MachOBinary.

Definition at line 74 of file MachOBinary.hpp.

The documentation for this class was generated from the following files:

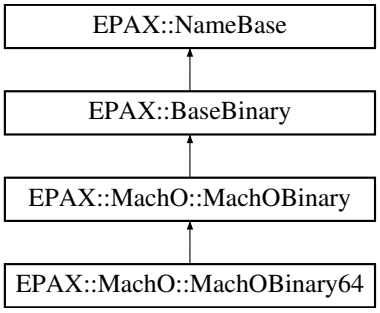
- MachOBinary.hpp

- MachOBinary.cpp

6.30 EPAX::MachO::MachOBinary64 Class Reference

`#include <MachOBinary.hpp>`

Inheritance diagram for EPAX::MachO::MachOBinary64:



Public Member Functions

- MachOBinary64 (std::string n)
- virtual ~MachOBinary64 ()
- BinaryFormat getFormat ()

6.30.1 Detailed Description

Definition at line 77 of file MachOBinary.hpp.

6.30.2 Constructor & Destructor Documentation

6.30.2.1 EPAX::MachO::MachOBinary64::MachOBinary64 (std::string *n*)

Definition at line 77 of file MachOBinary.cpp.

6.30.2.2 virtual EPAX::MachO::MachOBinary64::~MachOBinary64 ( ) [inline, virtual]

Definition at line 80 of file MachOBinary.hpp.

6.30.3 Member Function Documentation

6.30.3.1 BinaryFormat EPAX::MachO::MachOBinary64::getFormat ( ) [inline, virtual]

Implements EPAX::MachO::MachOBinary.

Definition at line 82 of file MachOBinary.hpp.

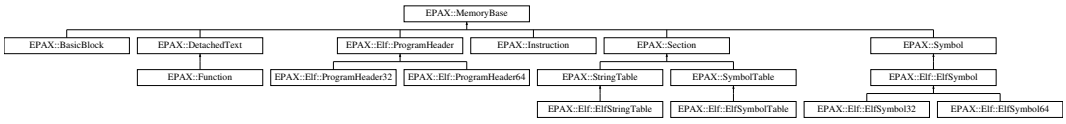
The documentation for this class was generated from the following files:

- MachOBinary.hpp
- MachOBinary.cpp

6.31 EPAX::MemoryBase Class Reference

`#include <BaseClass.hpp>`

Inheritance diagram for EPAX::MemoryBase:



Public Member Functions

- MemoryBase (uint64_t a, uint64_t s)
- MemoryBase ()
- virtual ~MemoryBase ()
- uint64_t getMemoryAddress ()
- uint64_t getMemorySize ()
- void setMemorySize (uint64_t s)
- bool inRange (uint64_t a)

6.31.1 Detailed Description

Definition at line 61 of file BaseClass.hpp.

6.31.2 Constructor & Destructor Documentation

6.31.2.1 EPAX::MemoryBase::MemoryBase (uint64_t *a*, uint64_t *s*) [inline]

Definition at line 67 of file BaseClass.hpp.

6.31.2.2 EPAX::MemoryBase::MemoryBase ( ) [inline]

Definition at line 69 of file BaseClass.hpp.

6.31.2.3 virtual EPAX::MemoryBase::~MemoryBase ( ) [inline, virtual]

Definition at line 71 of file BaseClass.hpp.

6.31.3 Member Function Documentation

6.31.3.1 uint64_t EPAX::MemoryBase::getMemoryAddress ( ) [inline]

Definition at line 73 of file BaseClass.hpp.

6.31.3.2 uint64_t EPAX::MemoryBase::getMemorySize ( ) [inline]

Definition at line 74 of file BaseClass.hpp.

6.31.3.3 bool EPAX::MemoryBase::inRange (uint64_t *a*)

Definition at line 54 of file BaseClass.cpp.

6.31.3.4 void EPAX::MemoryBase::setMemorySize (uint64_t *s*) [inline]

Definition at line 75 of file BaseClass.hpp.

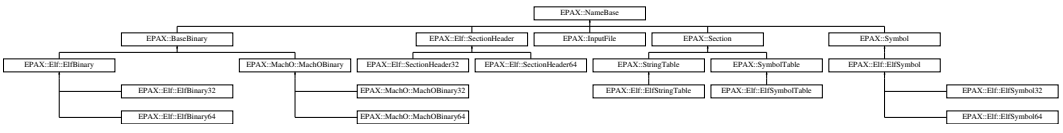
The documentation for this class was generated from the following files:

- BaseClass.hpp
- BaseClass.cpp

6.32 EPAX::NameBase Class Reference

`#include <BaseClass.hpp>`

Inheritance diagram for EPAX::NameBase:



Public Member Functions

- NameBase (std::string n)
- NameBase ()
- virtual ~NameBase ()

- std::string getName ()
- void setName (std::string n)

6.32.1 Detailed Description

Definition at line 107 of file BaseClass.hpp.

6.32.2 Constructor & Destructor Documentation

6.32.2.1 EPAX::NameBase::NameBase (std::string *n*) [inline]

Definition at line 112 of file BaseClass.hpp.

6.32.2.2 EPAX::NameBase::NameBase ( ) [inline]

Definition at line 113 of file BaseClass.hpp.

6.32.2.3 virtual EPAX::NameBase::~NameBase ( ) [inline, virtual]

Definition at line 114 of file BaseClass.hpp.

6.32.3 Member Function Documentation

6.32.3.1 std::string EPAX::NameBase::getName ( ) [inline]

Definition at line 116 of file BaseClass.hpp.

6.32.3.2 void EPAX::NameBase::setName (std::string *n*) [inline]

Definition at line 117 of file BaseClass.hpp.

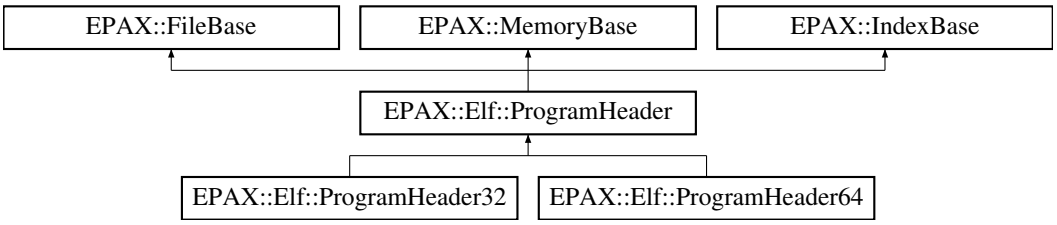
The documentation for this class was generated from the following file:

- BaseClass.hpp

6.33 EPAX::Elf::ProgramHeader Class Reference

`#include <ElfBinary.hpp>`

Inheritance diagram for EPAX::Elf::ProgramHeader:



Public Member Functions

- ProgramHeader (BaseBinary *b, uint64_t o, uint64_t s, uint32_t i)
- virtual ~ProgramHeader ()
- bool isValidVaddr (uint64_t v)
- uint64_t vaddrToFileaddr (uint64_t v)
- virtual uint64_t getVaddr ()=0
- virtual uint64_t getPaddr ()=0
- virtual uint64_t getFSize ()=0
- virtual uint64_t getMSize ()=0
- virtual uint32_t getSegmentType ()=0
- virtual uint64_t getFlags ()=0
- virtual uint32_t getAlignment ()=0
- virtual uint64_t getFOffset ()=0

Protected Attributes

- rawbyte_t * entry

6.33.1 Detailed Description

Definition at line 319 of file ElfBinary.hpp.

6.33.2 Constructor & Destructor Documentation

6.33.2.1 EPAX::Elf::ProgramHeader::ProgramHeader (BaseBinary * *b*, uint64_t *o*, uint64_t *s*, uint32_t *i*)

Definition at line 691 of file ElfBinary.cpp.

6.33.2.2 EPAX::Elf::ProgramHeader::~ProgramHeader ( ) [virtual]

Definition at line 698 of file ElfBinary.cpp.

6.33.3 Member Function Documentation

6.33.3.1 virtual uint32_t EPAX::Elf::ProgramHeader::getAlignment ( ) [pure virtual]

Implemented in EPAX::Elf::ProgramHeader64, and EPAX::Elf::ProgramHeader32.

6.33.3.2 virtual uint64_t EPAX::Elf::ProgramHeader::getFlags ( ) [pure virtual]

Implemented in EPAX::Elf::ProgramHeader64, and EPAX::Elf::ProgramHeader32.

6.33.3.3 virtual uint64_t EPAX::Elf::ProgramHeader::getFOffset ( ) [pure virtual]

Implemented in EPAX::Elf::ProgramHeader64, and EPAX::Elf::ProgramHeader32.

6.33.3.4 virtual uint64_t EPAX::Elf::ProgramHeader::getFSize ( ) [pure virtual]

Implemented in EPAX::Elf::ProgramHeader64, and EPAX::Elf::ProgramHeader32.

6.33.3.5 virtual uint64_t EPAX::Elf::ProgramHeader::getMSize ( ) [pure virtual]

Implemented in EPAX::Elf::ProgramHeader64, and EPAX::Elf::ProgramHeader32.

6.33.3.6 virtual uint64_t EPAX::Elf::ProgramHeader::getPaddr ( ) [pure virtual]

Implemented in EPAX::Elf::ProgramHeader64, and EPAX::Elf::ProgramHeader32.

6.33.3.7 virtual uint32_t EPAX::Elf::ProgramHeader::getSegmentType ( ) [pure virtual]

Implemented in EPAX::Elf::ProgramHeader64, and EPAX::Elf::ProgramHeader32.

6.33.3.8 virtual uint64_t EPAX::Elf::ProgramHeader::getVaddr ( ) [pure virtual]

Implemented in EPAX::Elf::ProgramHeader64, and EPAX::Elf::ProgramHeader32.

6.33.3.9 bool EPAX::Elf::ProgramHeader::isValidVaddr (uint64_t *v*)

Definition at line 873 of file ElfBinary.cpp.

6.33.3.10 uint64_t EPAX::Elf::ProgramHeader::vaddrToFileaddr (uint64_t *v*)

Definition at line 877 of file ElfBinary.cpp.

6.33.4 Member Data Documentation

6.33.4.1 rawbyte_t* EPAX::Elf::ProgramHeader::entry [protected]

Definition at line 321 of file ElfBinary.hpp.

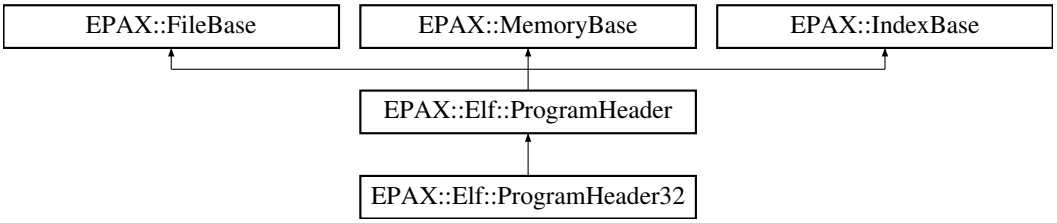
The documentation for this class was generated from the following files:

- ElfBinary.hpp
- ElfBinary.cpp

6.34 EPAX::Elf::ProgramHeader32 Class Reference

`#include <ElfBinary.hpp>`

Inheritance diagram for EPAX::Elf::ProgramHeader32:



Public Member Functions

- ProgramHeader32 (BaseBinary *b, uint64_t o, uint64_t s, uint32_t i)
- virtual ~ProgramHeader32 ()
- uint64_t getVaddr ()
- uint64_t getPaddr ()
- uint64_t getFSize ()
- uint64_t getMSize ()
- uint32_t getSegmentType ()
- uint64_t getFlags ()
- uint32_t getAlignment ()
- uint64_t getFOffset ()

6.34.1 Detailed Description

Definition at line 340 of file ElfBinary.hpp.

6.34.2 Constructor & Destructor Documentation

6.34.2.1 EPAX::Elf::ProgramHeader32::ProgramHeader32 (BaseBinary * *b*, uint64_t *o*, uint64_t *s*, uint32_t *i*)

Definition at line 884 of file ElfBinary.cpp.

6.34.2.2 virtual EPAX::Elf::ProgramHeader32::~ProgramHeader32 ( ) [inline, virtual]

Definition at line 343 of file ElfBinary.hpp.

6.34.3 Member Function Documentation

6.34.3.1 uint32_t EPAX::Elf::ProgramHeader32::getAlignment ( ) [virtual]

Implements EPAX::Elf::ProgramHeader.

Definition at line 938 of file ElfBinary.cpp.

6.34.3.2 uint64_t EPAX::Elf::ProgramHeader32::getFlags ( ) [virtual]

Implements EPAX::Elf::ProgramHeader.

Definition at line 930 of file ElfBinary.cpp.

6.34.3.3 uint64_t EPAX::Elf::ProgramHeader32::getFOffset ( ) [virtual]

Implements EPAX::Elf::ProgramHeader.

Definition at line 954 of file ElfBinary.cpp.

6.34.3.4 uint64_t EPAX::Elf::ProgramHeader32::getFSize ( ) [virtual]

Implements EPAX::Elf::ProgramHeader.

Definition at line 914 of file ElfBinary.cpp.

6.34.3.5 uint64_t EPAX::Elf::ProgramHeader32::getMSize ( ) [virtual]

Implements EPAX::Elf::ProgramHeader.

Definition at line 922 of file ElfBinary.cpp.

6.34.3.6 uint64_t EPAX::Elf::ProgramHeader32::getPaddr ( ) [virtual]

Implements EPAX::Elf::ProgramHeader.

Definition at line 906 of file ElfBinary.cpp.

6.34.3.7 uint32_t EPAX::Elf::ProgramHeader32::getSegmentType ( ) [virtual]

Implements EPAX::Elf::ProgramHeader.

Definition at line 946 of file ElfBinary.cpp.

6.34.3.8 uint64_t EPAX::Elf::ProgramHeader32::getVaddr ( ) [virtual]

Implements EPAX::Elf::ProgramHeader.

Definition at line 898 of file ElfBinary.cpp.

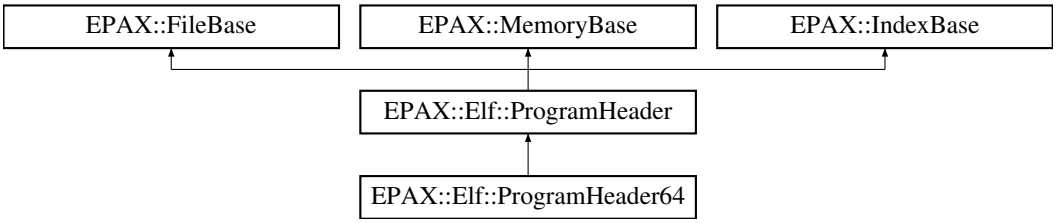
The documentation for this class was generated from the following files:

- ElfBinary.hpp
- ElfBinary.cpp

6.35 EPAX::Elf::ProgramHeader64 Class Reference

`#include <ElfBinary.hpp>`

Inheritance diagram for EPAX::Elf::ProgramHeader64:



Public Member Functions

- ProgramHeader64 (BaseBinary *b, uint64_t o, uint64_t s, uint32_t i)
- virtual ~ProgramHeader64 ()
- uint64_t getVaddr ()
- uint64_t getPaddr ()
- uint64_t getFSize ()

- uint64_t getMSize ()
- uint32_t getSegmentType ()
- uint64_t getFlags ()
- uint32_t getAlignment ()
- uint64_t getFOffset ()

6.35.1 Detailed Description

Definition at line 355 of file ElfBinary.hpp.

6.35.2 Constructor & Destructor Documentation

6.35.2.1 EPAX::Elf::ProgramHeader64::ProgramHeader64 (BaseBinary * *b*, uint64_t *o*, uint64_t *s*, uint32_t *i*)

Definition at line 891 of file ElfBinary.cpp.

6.35.2.2 virtual EPAX::Elf::ProgramHeader64::~ProgramHeader64 ( ) [inline, virtual]

Definition at line 358 of file ElfBinary.hpp.

6.35.3 Member Function Documentation

6.35.3.1 uint32_t EPAX::Elf::ProgramHeader64::getAlignment ( ) [virtual]

Implements EPAX::Elf::ProgramHeader.

Definition at line 942 of file ElfBinary.cpp.

6.35.3.2 uint64_t EPAX::Elf::ProgramHeader64::getFlags ( ) [virtual]

Implements EPAX::Elf::ProgramHeader.

Definition at line 934 of file ElfBinary.cpp.

6.35.3.3 uint64_t EPAX::Elf::ProgramHeader64::getFOffset ( ) [virtual]

Implements EPAX::Elf::ProgramHeader.

Definition at line 958 of file ElfBinary.cpp.

6.35.3.4 uint64_t EPAX::Elf::ProgramHeader64::getFSize ( ) [virtual]

Implements EPAX::Elf::ProgramHeader.

Definition at line 918 of file ElfBinary.cpp.

6.35.3.5 uint64_t EPAX::Elf::ProgramHeader64::getMSize ( ) [virtual]

Implements EPAX::Elf::ProgramHeader.

Definition at line 926 of file ElfBinary.cpp.

6.35.3.6 uint64_t EPAX::Elf::ProgramHeader64::getPaddr ( ) [virtual]

Implements EPAX::Elf::ProgramHeader.

Definition at line 910 of file ElfBinary.cpp.

6.35.3.7 uint32_t EPAX::Elf::ProgramHeader64::getSegmentType ( ) [virtual]

Implements EPAX::Elf::ProgramHeader.

Definition at line 950 of file ElfBinary.cpp.

6.35.3.8 uint64_t EPAX::Elf::ProgramHeader64::getVaddr ( ) [virtual]

Implements EPAX::Elf::ProgramHeader.

Definition at line 902 of file ElfBinary.cpp.

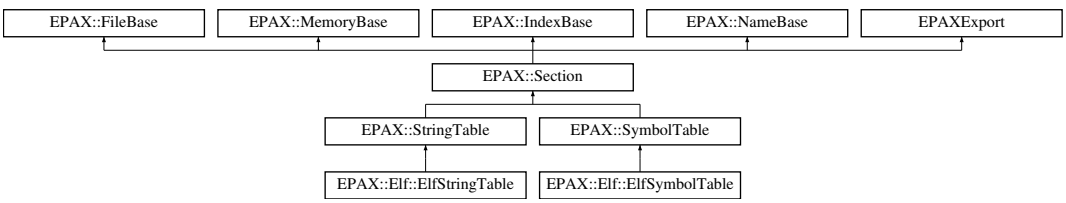
The documentation for this class was generated from the following files:

- ElfBinary.hpp
- ElfBinary.cpp

6.36 EPAX::Section Class Reference

`#include <Section.hpp>`

Inheritance diagram for EPAX::Section:



Public Member Functions

- Section (BaseBinary *b, uint64_t o, uint64_t fs, uint64_t ma, uint64_t ms, uint32_t i, std::string n)
- virtual ~Section ()
- virtual void print (std::ostream &stream=std::cout)
- virtual bool isText ()
- virtual bool isData ()
- virtual bool isBSS ()
- virtual bool isDebug ()
- virtual bool isString ()
- virtual bool isSymbol ()

6.36.1 Detailed Description

Definition at line 35 of file Section.hpp.

6.36.2 Constructor & Destructor Documentation

6.36.2.1 EPAX::Section::Section (BaseBinary * *b*, uint64_t *o*, uint64_t *fs*, uint64_t *ma*, uint64_t *ms*, uint32_t *i*, std::string *n*)

Definition at line 31 of file Section.cpp.

6.36.2.2 virtual EPAX::Section::~Section ( ) [inline, virtual]

Definition at line 38 of file Section.hpp.

6.36.3 Member Function Documentation

6.36.3.1 virtual bool EPAX::Section::isBSS ( ) [inline, virtual]

Definition at line 44 of file Section.hpp.

6.36.3.2 virtual bool EPAX::Section::isData ( ) [inline, virtual]

Definition at line 43 of file Section.hpp.

6.36.3.3 virtual bool EPAX::Section::isDebug ( ) [inline, virtual]

Definition at line 45 of file Section.hpp.

6.36.3.4 virtual bool EPAX::Section::isString ( ) [inline, virtual]

Reimplemented in EPAX::StringTable.

Definition at line 46 of file Section.hpp.

6.36.3.5 virtual bool EPAX::Section::isSymbol ( ) [inline, virtual]

Reimplemented in EPAX::SymbolTable.

Definition at line 47 of file Section.hpp.

6.36.3.6 virtual bool EPAX::Section::isText ( ) [inline, virtual]

Definition at line 42 of file Section.hpp.

6.36.3.7 void EPAX::Section::print (std::ostream & *stream* = std::cout) [virtual]

Reimplemented in EPAX::Elf::ElfSymbolTable, EPAX::Elf::ElfStringTable, and EPAX::SymbolTable.

Definition at line 40 of file Section.cpp.

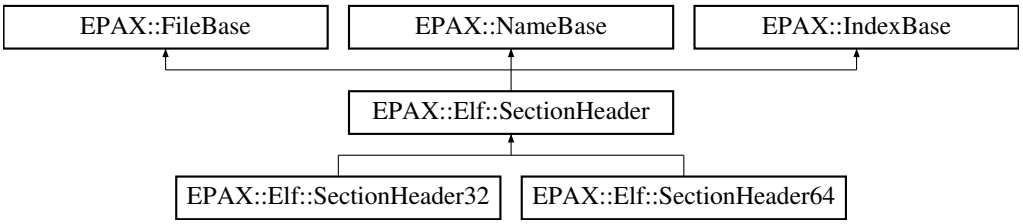
The documentation for this class was generated from the following files:

- Section.hpp
- Section.cpp

6.37 EPAX::Elf::SectionHeader Class Reference

`#include <ElfBinary.hpp>`

Inheritance diagram for EPAX::Elf::SectionHeader:



Public Member Functions

- SectionHeader (BaseBinary *b, uint64_t o, uint64_t s, uint32_t i)

- virtual ~SectionHeader ()
- void print (std::ostream &stream=std::cout)
- bool isText ()
- bool isData ()
- bool isBSS ()
- bool isDebug ()
- bool isString ()
- bool isSymbol ()
- bool isRead ()
- bool isWrite ()
- bool isExec ()
- bool isAlloc ()
- bool isMerge ()
- bool inRange (uint64_t a)
- virtual uint64_t getNameIndex ()=0
- virtual uint64_t getType ()=0
- virtual uint64_t getVirtAddr ()=0
- virtual uint64_t getFileOffset ()=0
- virtual uint64_t getSectionLink ()=0
- virtual uint64_t getAlignment ()=0
- virtual uint64_t getEntrySize ()=0
- virtual uint64_t getFlags ()=0
- virtual uint64_t getSize ()=0

Protected Attributes

- rawbyte_t * entry

6.37.1 Detailed Description

Definition at line 248 of file ElfBinary.hpp.

6.37.2 Constructor & Destructor Documentation

6.37.2.1 EPAX::Elf::SectionHeader::SectionHeader (BaseBinary * *b*, uint64_t *o*, uint64_t *s*, uint32_t *i*)

Definition at line 677 of file ElfBinary.cpp.

6.37.2.2 EPAX::Elf::SectionHeader::~SectionHeader ( ) [virtual]

Definition at line 685 of file ElfBinary.cpp.

6.37.3 Member Function Documentation

6.37.3.1 virtual uint64_t EPAX::Elf::SectionHeader::getAlignment ( ) [pure virtual]

Implemented in EPAX::Elf::SectionHeader64, and EPAX::Elf::SectionHeader32.

6.37.3.2 virtual uint64_t EPAX::Elf::SectionHeader::getEntrySize ( ) [pure virtual]

Implemented in EPAX::Elf::SectionHeader64, and EPAX::Elf::SectionHeader32.

6.37.3.3 virtual uint64_t EPAX::Elf::SectionHeader::getFileOffset ( ) [pure virtual]

Reimplemented from EPAX::FileBase.

Implemented in EPAX::Elf::SectionHeader64, and EPAX::Elf::SectionHeader32.

6.37.3.4 virtual uint64_t EPAX::Elf::SectionHeader::getFlags ( ) [pure virtual]

Implemented in EPAX::Elf::SectionHeader64, and EPAX::Elf::SectionHeader32.

6.37.3.5 virtual uint64_t EPAX::Elf::SectionHeader::getNameIndex ( ) [pure virtual]

Implemented in EPAX::Elf::SectionHeader64, and EPAX::Elf::SectionHeader32.

6.37.3.6 virtual uint64_t EPAX::Elf::SectionHeader::getSectionLink ( ) [pure virtual]

Implemented in EPAX::Elf::SectionHeader64, and EPAX::Elf::SectionHeader32.

6.37.3.7 virtual uint64_t EPAX::Elf::SectionHeader::getSize ( ) [pure virtual]

Implemented in EPAX::Elf::SectionHeader64, and EPAX::Elf::SectionHeader32.

6.37.3.8 virtual uint64_t EPAX::Elf::SectionHeader::getType ( ) [pure virtual]

Implemented in EPAX::Elf::SectionHeader64, and EPAX::Elf::SectionHeader32.

6.37.3.9 virtual uint64_t EPAX::Elf::SectionHeader::getVirtAddr ( ) [pure virtual]

Implemented in EPAX::Elf::SectionHeader64, and EPAX::Elf::SectionHeader32.

6.37.3.10 bool EPAX::Elf::SectionHeader::inRange (uint64_t *a*)

Definition at line 736 of file ElfBinary.cpp.

6.37.3.11 bool EPAX::Elf::SectionHeader::isAlloc ( )

Definition at line 779 of file ElfBinary.cpp.

6.37.3.12 bool EPAX::Elf::SectionHeader::isBSS ( )

Definition at line 751 of file ElfBinary.cpp.

6.37.3.13 bool EPAX::Elf::SectionHeader::isData ( )

Definition at line 747 of file ElfBinary.cpp.

6.37.3.14 bool EPAX::Elf::SectionHeader::isDebug ( )

Definition at line 755 of file ElfBinary.cpp.

6.37.3.15 bool EPAX::Elf::SectionHeader::isExec ( )

Definition at line 775 of file ElfBinary.cpp.

6.37.3.16 bool EPAX::Elf::SectionHeader::isMerge ( )

Definition at line 783 of file ElfBinary.cpp.

6.37.3.17 bool EPAX::Elf::SectionHeader::isRead ( )

Definition at line 767 of file ElfBinary.cpp.

6.37.3.18 bool EPAX::Elf::SectionHeader::isString ( )

Definition at line 759 of file ElfBinary.cpp.

6.37.3.19 bool EPAX::Elf::SectionHeader::isSymbol ( )

Definition at line 763 of file ElfBinary.cpp.

6.37.3.20 bool EPAX::Elf::SectionHeader::isText ( )

Definition at line 743 of file ElfBinary.cpp.

6.37.3.21 bool EPAX::Elf::SectionHeader::isWrite ( )

Definition at line 771 of file ElfBinary.cpp.

6.37.3.22 void EPAX::Elf::SectionHeader::print (std::ostream & *stream* = `std::cout`)

Definition at line 724 of file ElfBinary.cpp.

6.37.4 Member Data Documentation

6.37.4.1 rawbyte_t* EPAX::Elf::SectionHeader::entry [protected]

Definition at line 250 of file ElfBinary.hpp.

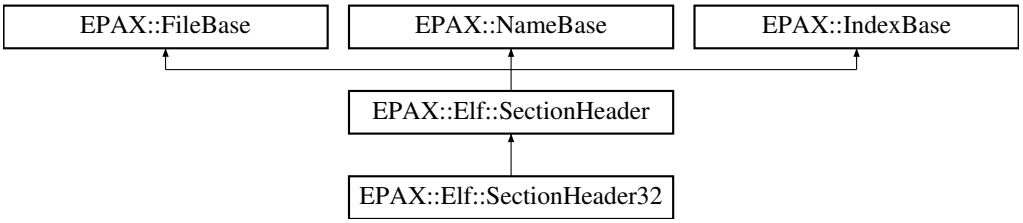
The documentation for this class was generated from the following files:

- ElfBinary.hpp
- ElfBinary.cpp

6.38 EPAX::Elf::SectionHeader32 Class Reference

`#include <ElfBinary.hpp>`

Inheritance diagram for EPAX::Elf::SectionHeader32:



Public Member Functions

- SectionHeader32 (BaseBinary *b, uint64_t o, uint64_t s, uint32_t i)
- virtual ~SectionHeader32 ()
- uint64_t getNameIndex ()
- uint64_t getType ()
- uint64_t getVirtAddr ()
- uint64_t getFileOffset ()
- uint64_t getSectionLink ()
- uint64_t getAlignment ()
- uint64_t getEntrySize ()
- uint64_t getFlags ()
- uint64_t getSize ()

6.38.1 Detailed Description

Definition at line 285 of file ElfBinary.hpp.

6.38.2 Constructor & Destructor Documentation

6.38.2.1 EPAX::Elf::SectionHeader32::SectionHeader32 (BaseBinary * *b*, uint64_t *o*, uint64_t *s*, uint32_t *i*)

Definition at line 787 of file ElfBinary.cpp.

6.38.2.2 virtual EPAX::Elf::SectionHeader32::~SectionHeader32 ( ) [inline, virtual]

Definition at line 288 of file ElfBinary.hpp.

6.38.3 Member Function Documentation

6.38.3.1 uint64_t EPAX::Elf::SectionHeader32::getAlignment ( ) [virtual]

Implements EPAX::Elf::SectionHeader.

Definition at line 849 of file ElfBinary.cpp.

6.38.3.2 uint64_t EPAX::Elf::SectionHeader32::getEntrySize ( ) [virtual]

Implements EPAX::Elf::SectionHeader.

Definition at line 857 of file ElfBinary.cpp.

6.38.3.3 uint64_t EPAX::Elf::SectionHeader32::getFileOffset ( ) [virtual]

Implements EPAX::Elf::SectionHeader.

Definition at line 833 of file ElfBinary.cpp.

6.38.3.4 uint64_t EPAX::Elf::SectionHeader32::getFlags ( ) [virtual]

Implements EPAX::Elf::SectionHeader.

Definition at line 865 of file ElfBinary.cpp.

6.38.3.5 uint64_t EPAX::Elf::SectionHeader32::getNameIndex ( ) [virtual]

Implements EPAX::Elf::SectionHeader.

Definition at line 801 of file ElfBinary.cpp.

6.38.3.6 uint64_t EPAX::Elf::SectionHeader32::getSectionLink ( ) [virtual]

Implements EPAX::Elf::SectionHeader.

Definition at line 841 of file ElfBinary.cpp.

6.38.3.7 uint64_t EPAX::Elf::SectionHeader32::getSize ( ) [virtual]

Implements EPAX::Elf::SectionHeader.

Definition at line 817 of file ElfBinary.cpp.

6.38.3.8 uint64_t EPAX::Elf::SectionHeader32::getType ( ) [virtual]

Implements EPAX::Elf::SectionHeader.

Definition at line 809 of file ElfBinary.cpp.

6.38.3.9 uint64_t EPAX::Elf::SectionHeader32::getVirtAddr ( ) [virtual]

Implements EPAX::Elf::SectionHeader.

Definition at line 825 of file ElfBinary.cpp.

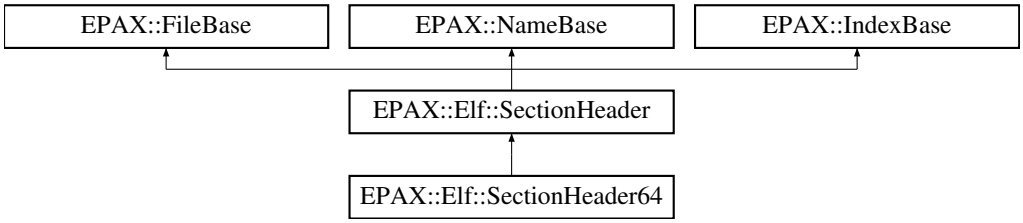
The documentation for this class was generated from the following files:

- ElfBinary.hpp
- ElfBinary.cpp

6.39 EPAX::Elf::SectionHeader64 Class Reference

`#include <ElfBinary.hpp>`

Inheritance diagram for EPAX::Elf::SectionHeader64:



Public Member Functions

- SectionHeader64 (BaseBinary *b, uint64_t o, uint64_t s, uint32_t i)
- virtual ~SectionHeader64 ()
- uint64_t getNameIndex ()
- uint64_t getType ()
- uint64_t getVirtAddr ()
- uint64_t getFileOffset ()
- uint64_t getSectionLink ()
- uint64_t getAlignment ()
- uint64_t getEntrySize ()
- uint64_t getFlags ()
- uint64_t getSize ()

6.39.1 Detailed Description

Definition at line 302 of file ElfBinary.hpp.

6.39.2 Constructor & Destructor Documentation

6.39.2.1 EPAX::Elf::SectionHeader64::SectionHeader64 (BaseBinary * *b*, uint64_t *o*, uint64_t *s*, uint32_t *i*)

Definition at line 794 of file ElfBinary.cpp.

6.39.2.2 virtual EPAX::Elf::SectionHeader64::~SectionHeader64 ( ) [inline, virtual]

Definition at line 305 of file ElfBinary.hpp.

6.39.3 Member Function Documentation

6.39.3.1 uint64_t EPAX::Elf::SectionHeader64::getAlignment ( ) [virtual]

Implements EPAX::Elf::SectionHeader.

Definition at line 853 of file ElfBinary.cpp.

6.39.3.2 uint64_t EPAX::Elf::SectionHeader64::getEntrySize ( ) [virtual]

Implements EPAX::Elf::SectionHeader.

Definition at line 861 of file ElfBinary.cpp.

6.39.3.3 uint64_t EPAX::Elf::SectionHeader64::getFileOffset ( ) [virtual]

Implements EPAX::Elf::SectionHeader.

Definition at line 837 of file ElfBinary.cpp.

6.39.3.4 uint64_t EPAX::Elf::SectionHeader64::getFlags ( ) [virtual]

Implements EPAX::Elf::SectionHeader.

Definition at line 869 of file ElfBinary.cpp.

6.39.3.5 uint64_t EPAX::Elf::SectionHeader64::getNameIndex ( ) [virtual]

Implements EPAX::Elf::SectionHeader.

Definition at line 805 of file ElfBinary.cpp.

6.39.3.6 uint64_t EPAX::Elf::SectionHeader64::getSectionLink ( ) [virtual]

Implements EPAX::Elf::SectionHeader.

Definition at line 845 of file ElfBinary.cpp.

6.39.3.7 uint64_t EPAX::Elf::SectionHeader64::getSize ( ) [virtual]

Implements EPAX::Elf::SectionHeader.

Definition at line 821 of file ElfBinary.cpp.

6.39.3.8 uint64_t EPAX::Elf::SectionHeader64::getType ( ) [virtual]

Implements EPAX::Elf::SectionHeader.

Definition at line 813 of file ElfBinary.cpp.

6.39.3.9 uint64_t EPAX::Elf::SectionHeader64::getVirtAddr ( ) [virtual]

Implements EPAX::Elf::SectionHeader.

Definition at line 829 of file ElfBinary.cpp.

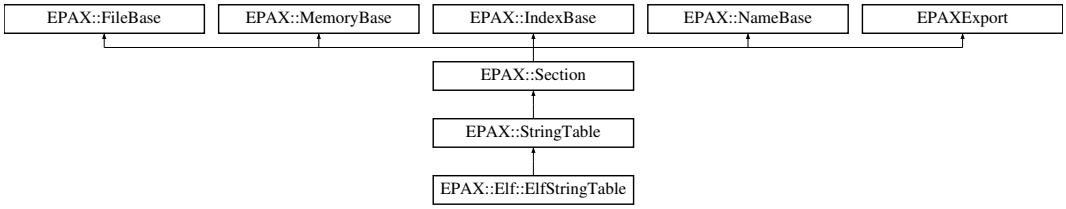
The documentation for this class was generated from the following files:

- ElfBinary.hpp
- ElfBinary.cpp

6.40 EPAX::StringTable Class Reference

`#include <Symbol.hpp>`

Inheritance diagram for EPAX::StringTable:



Public Member Functions

- StringTable (BaseBinary *b, uint64_t o, uint64_t fs, uint64_t ma, uint64_t ms, uint32_t i, std::string n)
- virtual ~StringTable ()
- virtual char * getStringAt (uint32_t i)=0
- bool isString ()

6.40.1 Detailed Description

Definition at line 60 of file Symbol.hpp.

6.40.2 Constructor & Destructor Documentation

6.40.2.1 EPAX::StringTable::StringTable (BaseBinary * *b*, uint64_t *o*, uint64_t *fs*, uint64_t *ma*, uint64_t *ms*, uint32_t *i*, std::string *n*)

Definition at line 61 of file Symbol.cpp.

6.40.2.2 virtual EPAX::StringTable::~StringTable ( ) [inline, virtual]

Definition at line 63 of file Symbol.hpp.

6.40.3 Member Function Documentation

6.40.3.1 virtual char* EPAX::StringTable::getStringAt (uint32_t *i*) [pure virtual]

Implemented in EPAX::Elf::ElfStringTable.

6.40.3.2 bool EPAX::StringTable::isString ( ) [inline, virtual]

Reimplemented from EPAX::Section.

Definition at line 67 of file Symbol.hpp.

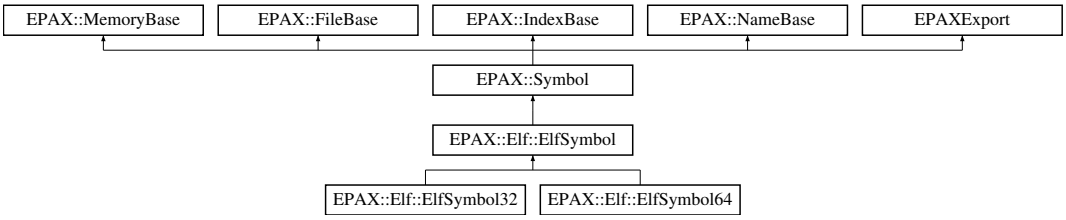
The documentation for this class was generated from the following files:

- Symbol.hpp
- Symbol.cpp

6.41 EPAX::Symbol Class Reference

`#include <Symbol.hpp>`

Inheritance diagram for EPAX::Symbol:



Public Member Functions

- Symbol (BaseBinary *b, uint64_t o, uint64_t s, uint32_t i)
- virtual ~Symbol ()
- virtual bool isFunction ()=0
- virtual bool isThumbFunction ()=0

6.41.1 Detailed Description

Definition at line 35 of file Symbol.hpp.

6.41.2 Constructor & Destructor Documentation

6.41.2.1 EPAX::Symbol::Symbol (BaseBinary * *b*, uint64_t *o*, uint64_t *s*, uint32_t *i*)

Definition at line 31 of file Symbol.cpp.

6.41.2.2 virtual EPAX::Symbol::~Symbol ( ) [inline, virtual]

Definition at line 38 of file Symbol.hpp.

6.41.3 Member Function Documentation

6.41.3.1 virtual bool EPAX::Symbol::isFunction ( ) [pure virtual]

Implemented in EPAX::Elf::ElfSymbol.

6.41.3.2 virtual bool EPAX::Symbol::isThumbFunction ( ) [pure virtual]

Implemented in EPAX::Elf::ElfSymbol.

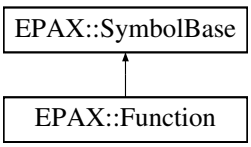
The documentation for this class was generated from the following files:

- Symbol.hpp
- Symbol.cpp

6.42 EPAX::SymbolBase Class Reference

`#include <BaseClass.hpp>`

Inheritance diagram for EPAX::SymbolBase:



Public Member Functions

- SymbolBase (Symbol *s)

- virtual ~SymbolBase ()
- Symbol * getSymbol ()
- void setSymbol (Symbol *s)
- std::string getName ()

6.42.1 Detailed Description

Definition at line 80 of file BaseClass.hpp.

6.42.2 Constructor & Destructor Documentation

6.42.2.1 EPAX::SymbolBase::SymbolBase (Symbol * *s*) [inline]

Definition at line 84 of file BaseClass.hpp.

6.42.2.2 virtual EPAX::SymbolBase::~SymbolBase ( ) [inline, virtual]

Definition at line 86 of file BaseClass.hpp.

6.42.3 Member Function Documentation

6.42.3.1 std::string EPAX::SymbolBase::getName ( )

Definition at line 39 of file BaseClass.cpp.

6.42.3.2 Symbol* EPAX::SymbolBase::getSymbol ( ) [inline]

Definition at line 88 of file BaseClass.hpp.

6.42.3.3 void EPAX::SymbolBase::setSymbol (Symbol * *s*) [inline]

Definition at line 89 of file BaseClass.hpp.

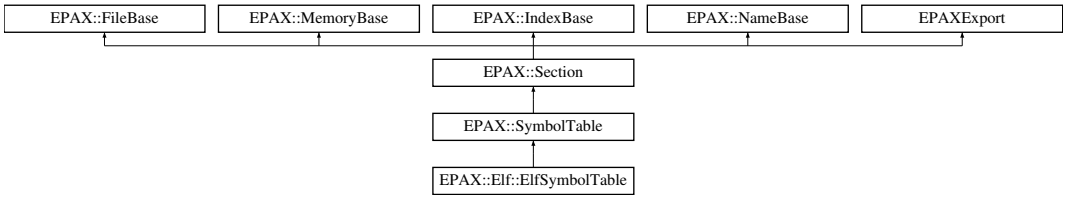
The documentation for this class was generated from the following files:

- BaseClass.hpp
- BaseClass.cpp

6.43 EPAX::SymbolTable Class Reference

`#include <Symbol.hpp>`

Inheritance diagram for EPAX::SymbolTable:



Public Member Functions

- SymbolTable (BaseBinary *b, uint64_t o, uint64_t fs, uint64_t ma, uint64_t ms, uint32_t i, std::string n)
- virtual ~SymbolTable ()
- uint32_t countSymbols ()
- Symbol * getSymbol (uint32_t i)
- virtual void print (std::ostream &stream=std::cout)=0
- bool isSymbol ()

Protected Attributes

- std::vector< Symbol * > * symbols

6.43.1 Detailed Description

Definition at line 44 of file Symbol.hpp.

6.43.2 Constructor & Destructor Documentation

6.43.2.1 EPAX::SymbolTable::SymbolTable (BaseBinary * *b*, uint64_t *o*, uint64_t *fs*, uint64_t *ma*, uint64_t *ms*, uint32_t *i*, std::string *n*)

Definition at line 40 of file Symbol.cpp.

6.43.2.2 EPAX::SymbolTable::~SymbolTable ( ) [virtual]

Definition at line 46 of file Symbol.cpp.

6.43.3 Member Function Documentation

6.43.3.1 uint32_t EPAX::SymbolTable::countSymbols ( ) [inline]

Definition at line 52 of file Symbol.hpp.

6.43.3.2 Symbol * EPAX::SymbolTable::getSymbol (uint32_t *i*)

Definition at line 56 of file Symbol.cpp.

6.43.3.3 bool EPAX::SymbolTable::isSymbol ( ) [inline, virtual]

Reimplemented from EPAX::Section.

Definition at line 57 of file Symbol.hpp.

6.43.3.4 virtual void EPAX::SymbolTable::print (std::ostream & *stream* = std::cout) [pure virtual]

Reimplemented from EPAX::Section.

Implemented in EPAX::Elf::ElfSymbolTable.

6.43.4 Member Data Documentation

6.43.4.1 std::vector<Symbol*>* EPAX::SymbolTable::symbols [protected]

Definition at line 46 of file Symbol.hpp.

The documentation for this class was generated from the following files:

- Symbol.hpp
- Symbol.cpp

Chapter 7

File Documentation

7.1 BaseClass.cpp File Reference

```
#include "EPAXCommonInternal.hpp"    #include "BaseClass.-
hpp"   #include "InputFile.hpp"   #include "Function.hpp" ×
#include "Symbol.hpp" #include "Section.hpp"
```

Namespaces

- namespace EPAX

Functions

- bool EPAX::compareMemory (MemoryBase ∗m1, MemoryBase ∗m2)

7.1.1 Detailed Description

7.1.2 LICENSE

Definition in file BaseClass.cpp.

7.2 BaseClass.hpp File Reference

```
#include "Binary.hpp"
```

Classes

- class EPAX::FileBase
- class EPAX::MemoryBase
- class EPAX::SymbolBase
- class EPAX::IndexBase
- class EPAX::NameBase
- class EPAX::BaseBinary

Namespaces

- namespace EPAX

Functions

- bool EPAX::compareMemory (MemoryBase *m1, MemoryBase *m2)

7.2.1 Detailed Description

7.2.2 LICENSE

Definition in file BaseClass.hpp.

7.3 BasicBlock.cpp File Reference

```
#include "EPAXCommonInternal.hpp"  #include "BasicBlock.-
hpp" #include "Instruction.hpp" #include "Function.hpp"
```

Namespaces

- namespace EPAX

7.3.1 Detailed Description

7.3.2 LICENSE

This file is part of the EPAX toolkit.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file BasicBlock.cpp.

7.4 BasicBlock.hpp File Reference

```
#include "BaseClass.hpp"
```

Classes

- class EPAX::BasicBlock

Namespaces

- namespace EPAX

7.4.1 Detailed Description

7.4.2 LICENSE

This file is part of the EPAX toolkit.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file BasicBlock.hpp.

7.5 Binary.cpp File Reference

```
#include "EPAXCommonInternal.hpp"  #include "Binary.hpp" ×
#include "ElfBinary.hpp" #include "Function.hpp" #include
"Instruction.hpp" #include "MachOBinary.hpp"
```

Namespaces

- namespace EPAX

Defines

- #define PTRACE_AND_CHECK(__opt, __pid, __addr, __data)
- #define VERIFY_SINGLE_FORMAT(__fmt__) EPAXAssert(f == BinaryFormat_-undefined, "This binary appears to be valid for two formats: " << BaseBinary-::getFormatName(__fmt__) << " and " << BaseBinary::getFormatName(f));

7.5.1 Detailed Description

7.5.2 LICENSE

This file is part of the EPAX toolkit.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file Binary.cpp.

7.5.3 Define Documentation

7.5.3.1 #define PTRACE_AND_CHECK(__opt, __pid, __addr, __data)

Value:

```
res = ptrace(__opt, __pid,__addr,__data);                              \
    if (res == -1){                                                        \
        EPAXErr << "ptrace failed with " << DEC(errno) << ENDL;        \
    }                                                                      \
    EPAXAssert(res == 0, "call to ptrace(" << #__opt << ", pid=" << DEC(__pid)
      << ", addr=" << HEX(__addr) << ", data=" << HEX(__data) << ") failed with error
      " << DEC(res));
```

Definition at line 61 of file Binary.cpp.

7.5.3.2 #define VERIFY_SINGLE_FORMAT(__fmt__) EPAXAssert(f == BinaryFormat_undefined, "This binary appears to be valid for two formats: " << BaseBinary::getFormatName(__fmt__) << " and " << BaseBinary::getFormatName(f));

7.6 Binary.hpp File Reference

Classes

- class EPAX::Binary

Namespaces

- namespace EPAX

Enumerations

- enum EPAX::BinaryFormat { EPAX::BinaryFormat_undefined = 0, EPAX::BinaryFormat_Elf32, EPAX::BinaryFormat_Elf64, EPAX::BinaryFormat_MachO32, EPAX::BinaryFormat_MachO64, EPAX::BinaryFormat_PE, EPAX::BinaryFormat_total }

7.6.1 Detailed Description

7.6.2 LICENSE

This file is part of the EPAX toolkit.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file Binary.hpp.

7.7 c_interface.py File Reference

Namespaces

- namespace c_interface

Functions

- def c_interface.file_exists
- def c_interface.error_die
- def c_interface.print_usage
- def c_interface.char_range
- def c_interface.main

Variables

- list c_interface.objs = ['BIN', 'SECT', 'FUNC', 'CFG', 'LOOP', 'BBL', 'INSN', 'SYM', 'FLOW']
- dictionary c_interface.other = {}
- dictionary c_interface.remove = {}

7.8 ControlFlow.cpp File Reference

```
#include "EPAXCommonInternal.hpp"  #include "BasicBlock.-
hpp" #include "ControlFlow.hpp" #include "Function.hpp" ×
#include "Instruction.hpp" #include "Loop.hpp"
```

Namespaces

- namespace EPAX

Functions

- void EPAX::DFS (std::vector< BasicBlock * > &backedg, BasicBlock *start, dyn_bitset &v, dyn_bitset &c)
- void EPAX::findDominators (std::vector< dyn_bitset * > &dominators, BasicBlock *start, std::vector< BasicBlock * > &bbs)
- void EPAX::findBackEdges (std::vector< BasicBlock * > &backedg, BasicBlock *start, std::vector< BasicBlock * > bbs)

7.8.1 Detailed Description

7.8.2 LICENSE

This file is part of the EPAX toolkit.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file ControlFlow.cpp.

7.9 ControlFlow.hpp File Reference

```
#include "BaseClass.hpp"
```

Classes

- class EPAX::ControlFlow

Namespaces

- namespace EPAX

7.9.1 Detailed Description

7.9.2 LICENSE

This file is part of the EPAX toolkit.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file ControlFlow.hpp.

7.10 DataStruct.hpp File Reference

```
#include "EPAXCommonInternal.hpp"
```

Classes

- class EPAX::dyn_bitset

Namespaces

- namespace EPAX

Defines

- #define __get_index(__idx) (__idx >> div)
- #define __has_bit(__idx) ((_elements[__get_index(__idx)] >> (__idx & mask)) & 1)
- #define __set_bit(__idx) _elements[__get_index(__idx)] |= (1 << (__idx & mask))
- #define __internal_size (__get_index(_size) + 1)

7.10.1 Detailed Description

7.10.2 LICENSE

This file is part of the EPAX toolkit.

Definition in file DataStruct.hpp.

7.10.3 Define Documentation

7.10.3.1 #define __get_index(__idx) (__idx >> div)

Definition at line 43 of file DataStruct.hpp.

7.10.3.2 #define __has_bit(__idx) ((_elements[__get_index(__idx)] >> (__idx & mask)) & 1)

Definition at line 44 of file DataStruct.hpp.

7.10.3.3 #define __internal_size (__get_index(_size) + 1)

Definition at line 46 of file DataStruct.hpp.

7.10.3.4 #define __set_bit(__idx) _elements[__get_index(__idx)] |= (1 << (__idx & mask))

Definition at line 45 of file DataStruct.hpp.

7.11 ElfBinary.cpp File Reference

```
#include "EPAXCommonInternal.hpp"   #include "Elf/elf.h" ×
#include "Binary.hpp" #include "ElfBinary.hpp" #include "-
Function.hpp" #include "InputFile.hpp"
```

Namespaces

- namespace EPAX
- namespace EPAX::Elf

Defines

- #define EHDR32_ENTRY ((Elf32_Ehdr*)entry)
- #define EHDR64_ENTRY ((Elf64_Ehdr*)entry)
- #define SYM32_ENTRY ((Elf32_Sym*)entry)
- #define SYM64_ENTRY ((Elf64_Sym*)entry)
- #define SHDR32_ENTRY ((Elf32_Shdr*)entry)
- #define SHDR64_ENTRY ((Elf64_Shdr*)entry)
- #define PHDR32_ENTRY ((Elf32_Phdr*)entry)
- #define PHDR64_ENTRY ((Elf64_Phdr*)entry)
- #define CASE(__typ__) case EM_ ## __typ__: std::cout << "isa=" << #__typ__ ; break

7.11.1 Detailed Description

7.11.2 LICENSE

This file is part of the EPAX toolkit.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file ElfBinary.cpp.

7.11.3 Define Documentation

7.11.3.1 #define CASE(__typ__) case EM_ ## __typ__: std::cout << "isa=" << #__typ__ ; break

7.11.3.2 #define EHDR32_ENTRY ((Elf32_Ehdr*)entry)

Definition at line 38 of file ElfBinary.cpp.

7.11.3.3 #define EHDR64_ENTRY ((Elf64_Ehdr*)entry)

Definition at line 39 of file ElfBinary.cpp.

7.11.3.4 #define PHDR32_ENTRY ((Elf32_Phdr*)entry)

Definition at line 44 of file ElfBinary.cpp.

7.11.3.5 #define PHDR64_ENTRY ((Elf64_Phdr*)entry)

Definition at line 45 of file ElfBinary.cpp.

7.11.3.6 #define SHDR32_ENTRY ((Elf32_Shdr*)entry)

Definition at line 42 of file ElfBinary.cpp.

7.11.3.7 #define SHDR64_ENTRY ((Elf64_Shdr*)entry)

Definition at line 43 of file ElfBinary.cpp.

7.11.3.8 #define SYM32_ENTRY ((Elf32_Sym*)entry)

Definition at line 40 of file ElfBinary.cpp.

7.11.3.9 #define SYM64_ENTRY ((Elf64_Sym*)entry)

Definition at line 41 of file ElfBinary.cpp.

7.12 ElfBinary.hpp File Reference

```
#include "BaseClass.hpp"  #include "Section.hpp"  #include
"Symbol.hpp"
```

Classes

- class EPAX::Elf::ElfBinary
- class EPAX::Elf::ElfBinary32
- class EPAX::Elf::ElfBinary64
- class EPAX::Elf::FileHeader
- class EPAX::Elf::FileHeader32
- class EPAX::Elf::FileHeader64
- class EPAX::Elf::ElfSymbol
- class EPAX::Elf::ElfSymbol32
- class EPAX::Elf::ElfSymbol64
- class EPAX::Elf::ElfStringTable
- class EPAX::Elf::ElfSymbolTable

- class EPAX::Elf::SectionHeader
- class EPAX::Elf::SectionHeader32
- class EPAX::Elf::SectionHeader64
- class EPAX::Elf::ProgramHeader
- class EPAX::Elf::ProgramHeader32
- class EPAX::Elf::ProgramHeader64

Namespaces

- namespace EPAX
- namespace EPAX::Elf

7.12.1 Detailed Description

7.12.2 LICENSE

This file is part of the EPAX toolkit.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file ElfBinary.hpp.

7.13 EPAX.cpp File Reference

`#include "EPAX.hpp" #include <stdlib.h> #include <assert.-h> #include <iostream> #include <fstream>`

Functions

- void error_out (char *prg, const char *msg)
- int main (int argc, char **argv)

7.13.1 Detailed Description

7.13.2 LICENSE

This file is part of the EPAX toolkit.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file EPAX.cpp.

7.13.3 Function Documentation

7.13.3.1 void error_out (char * *prg*, const char * *msg*)

Definition at line 31 of file EPAX.cpp.

7.13.3.2 int main (int *argc*, char ** *argv*)

Definition at line 37 of file EPAX.cpp.

7.14 EPAXCommonInternal.hpp File Reference

```
#include <stdint.h> #include <stdlib.h> #include <stddef.-
h> #include <string.h> #include <sys/ptrace.h> #include
<sys/user.h> #include <errno.h> #include <sys/types.-
h> #include <sys/wait.h> #include <unistd.h> #include
<iostream> #include <iomanip> #include <fstream> #include
<string> #include <vector> #include <map> #include <bitset> ×
#include <stack> #include <set> #include <algorithm>
```

Classes

- class EPAXExport

Defines

- #define INVALID_PTR (NULL)
- #define IS_VALID_PTR(__p__) ((__p__ != INVALID_PTR))

- #define MAX_STRING_SIZE (1024)
- #define NAME_UNKNOWN "__unknown__"
- #define EPAX_PREFACE "-EPAX- "
- #define ENDL "\n"
- #define TAB "\t"
- #define DEC(__n__) std::dec << (__n__)
- #define HEX(__n__) std::hex << "0x" << (__n__)
- #define BACKTRACE_LIMIT 64
- #define EPAXErr std::cerr << EPAX_PREFACE
- #define EPAXOut std::cout << EPAX_PREFACE
- #define EPAXWarn EPAXErr << " warning: "
- #define EPAXAssert(__stmt__, __msg__)
- #define EPAXDie(__msg__) EPAXAssert(false, __msg__)
- #define ShouldNotArrive EPAXDie("This function is not yet implemented")
- #define EPAXVerifyType(__type__, __obj__)
- #define __do_not_call__ EPAXAssert(false, "This function cannot be called.")
- #define ALIGN_PWR2(__addr, __exp) (__addr & ~((1 << __exp) - 1))
- #define INVALID_ADDRESS (0x0)
- #define ADDRESS_IS_THUMB(__a) ((__a & 0x1) == 1)

Typedefs

- typedef char rawbyte_t

Enumerations

- enum EPAXExportClass { EPAXExportClass_undefined = 0, EPAXExportClass_BIN, EPAXExportClass_SECT, EPAXExportClass_FUNC, EPAXExportClass_CFG, EPAXExportClass_LOOP, EPAXExportClass_BBL, EPAXExportClass_INSN, EPAXExportClass_SYM, EPAXExportClass_FLOW, EPAXExportClass_total }

7.14.1 Detailed Description

7.14.2 LICENSE

This file is part of the EPAX toolkit.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file EPAXCommonInternal.hpp.

7.14.3 Define Documentation

7.14.3.1 #define __do_not_call__ EPAXAssert(false, "This function cannot be called.")

Definition at line 126 of file EPAXCommonInternal.hpp.

7.14.3.2 #define ADDRESS_IS_THUMB(__a) ((__a & 0x1) == 1)

Definition at line 132 of file EPAXCommonInternal.hpp.

7.14.3.3 #define ALIGN_PWR2(__addr, __exp) (__addr & ~((1 << __exp) - 1))

Definition at line 127 of file EPAXCommonInternal.hpp.

7.14.3.4 #define BACKTRACE_LIMIT 64

Definition at line 67 of file EPAXCommonInternal.hpp.

7.14.3.5 #define DEC(__n__) std::dec << (__n__)

Definition at line 64 of file EPAXCommonInternal.hpp.

7.14.3.6 #define ENDL "\n"

Definition at line 62 of file EPAXCommonInternal.hpp.

7.14.3.7 #define EPAX_PREFACE "-EPAX- "

Definition at line 61 of file EPAXCommonInternal.hpp.

7.14.3.8 #define EPAXAssert(__stmt__, __msg__)

Value:

```
if (!(__stmt__)){                                                    \
        EPAXErr << "Assert failure: " << __PRETTY_FUNCTION__ << " at " <<
      __FILE__ << ":" << __LINE__ << ENDL; \
        EPAXErr << __msg__ << ENDL;                                    \
        exit(1); }
```

Definition at line 87 of file EPAXCommonInternal.hpp.

7.14.3.9 #define EPAXDie(__msg__) EPAXAssert(false, __msg__)

Definition at line 95 of file EPAXCommonInternal.hpp.

7.14.3.10 #define EPAXErr std::cerr << EPAX_PREFACE

Definition at line 73 of file EPAXCommonInternal.hpp.

7.14.3.11 #define EPAXOut std::cout << EPAX_PREFACE

Definition at line 74 of file EPAXCommonInternal.hpp.

7.14.3.12 #define EPAXVerifyType(__type__, __obj__)

Value:

```
EPAXAssert(IS_VALID_PTR(__obj__), "invalid object (NULL) found instead of " #
      __type__);\
    EPAXAssert(__obj__->getClass() == EPAXExportClass_ ## __type__, "Non-" #
      __type__ << " object found");
```

Definition at line 122 of file EPAXCommonInternal.hpp.

7.14.3.13 #define EPAXWarn EPAXErr << " warning: "

Definition at line 75 of file EPAXCommonInternal.hpp.

7.14.3.14 #define HEX(__n__) std::hex << "0x" << (__n__)

Definition at line 65 of file EPAXCommonInternal.hpp.

7.14.3.15 #define INVALID_ADDRESS (0x0)

Definition at line 131 of file EPAXCommonInternal.hpp.

7.14.3.16 #define INVALID_PTR (NULL)

Definition at line 56 of file EPAXCommonInternal.hpp.

7.14.3.17 #define IS_VALID_PTR(__p__) ((__p__ != INVALID_PTR))

Definition at line 57 of file EPAXCommonInternal.hpp.

7.14.3.18 #define MAX_STRING_SIZE (1024)

Definition at line 58 of file EPAXCommonInternal.hpp.

7.14.3.19 #define NAME_UNKNOWN "__unknown__"

Definition at line 59 of file EPAXCommonInternal.hpp.

7.14.3.20 #define ShouldNotArrive EPAXDie("This function is not yet implemented")

Definition at line 96 of file EPAXCommonInternal.hpp.

7.14.3.21 #define TAB "\t"

Definition at line 63 of file EPAXCommonInternal.hpp.

7.14.4 Typedef Documentation

7.14.4.1 typedef char rawbyte_t

Definition at line 129 of file EPAXCommonInternal.hpp.

7.14.5 Enumeration Type Documentation

7.14.5.1 enum EPAXExportClass

Enumerator:

EPAXExportClass_undefined

EPAXExportClass_BIN

EPAXExportClass_SECT

EPAXExportClass_FUNC

EPAXExportClass_CFG

EPAXExportClass_LOOP

EPAXExportClass_BBL

EPAXExportClass_INSN

EPAXExportClass_SYM

EPAXExportClass_FLOW

EPAXExportClass_total

Definition at line 98 of file EPAXCommonInternal.hpp.

7.15 Function.cpp File Reference

```
#include "EPAXCommonInternal.hpp"  #include "BasicBlock.-
hpp"  #include "ControlFlow.hpp"  #include "Function.hpp"
#include "InputFile.hpp" #include "Instruction.hpp" #include
"Symbol.hpp"
```

Namespaces

- namespace EPAX

7.15.1 Detailed Description

7.15.2 LICENSE

This file is part of the EPAX toolkit.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file Function.cpp.

7.16 Function.hpp File Reference

```
#include "BaseClass.hpp" #include "darm.h"
```

Classes

- class EPAX::DetachedText
- class EPAX::Function

Namespaces

- namespace EPAX

7.16.1 Detailed Description

7.16.2 LICENSE

This file is part of the EPAX toolkit.

Copyright (c) 2013, EP Analytics, Inc. All rights reserved.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file Function.hpp.

7.17 InputFile.cpp File Reference

```
#include "EPAXCommonInternal.hpp"    #include "InputFile.-
hpp"
```

Namespaces

- namespace EPAX

7.17.1 Detailed Description

7.17.2 LICENSE

This file is part of the EPAX toolkit.

Copyright (c) 2013, EP Analytics, Inc. All rights reserved.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file InputFile.cpp.

7.18 InputFile.hpp File Reference

```
#include "BaseClass.hpp"
```

Classes

- class EPAX::InputFile

Namespaces

- namespace EPAX

7.18.1 Detailed Description

7.18.2 LICENSE

This file is part of the EPAX toolkit.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file InputFile.hpp.

7.19 Instruction.cpp File Reference

```
#include "EPAXCommonInternal.hpp" #include "Instruction.-
hpp" #include "Function.hpp" #include <assert.h>
```

Namespaces

- namespace EPAX

Defines

- #define DARM_REGLIST_HASREG(__list, __reg) ((__list >> __reg) != 0)
- #define DARM_PREDICATE_UNCOND(__pred) (C_AL == __pred)

7.19.1 Detailed Description

7.19.2 LICENSE

This file is part of the EPAX toolkit.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file Instruction.cpp.

7.19.3 Define Documentation

7.19.3.1 #define DARM_PREDICATE_UNCOND(__pred) (C_AL == __pred)

Definition at line 35 of file Instruction.cpp.

7.19.3.2 #define DARM_REGLIST_HASREG(__list, __reg) ((__list >> __reg) != 0)

Definition at line 34 of file Instruction.cpp.

7.20 Instruction.hpp File Reference

```
#include "BaseClass.hpp" #include "darm.h"
```

Classes

- class EPAX::Instruction

Namespaces

- namespace EPAX

7.20.1 Detailed Description

7.20.2 LICENSE

This file is part of the EPAX toolkit.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file Instruction.hpp.

7.21 Interface.cpp File Reference

```
#include "EPAXCommonInternal.hpp"  #include "Interface.-
hpp"  #include "Interface.h"  #include "BasicBlock.hpp" ×
#include "Binary.hpp" #include "ControlFlow.hpp" #include
"Function.hpp" #include "Instruction.hpp" #include "Loop.-
hpp" #include "Symbol.hpp" #include "Section.hpp" #include
<iostream> #include <fstream>
```

Namespaces

- namespace EPAX

Functions

- BIN EPAX::BIN_create (std::string fileName)
- std::string EPAX::BIN_getName (BIN bin)
- void EPAX::BIN_destroy (BIN bin)
- void EPAX::BIN_run (BIN bin, int argc, char ∗argv[])
- FUNC EPAX::BIN_firstFunc (BIN bin)
- FUNC EPAX::BIN_nextFunc (BIN bin, FUNC func)
- bool EPAX::BIN_isLastFunc (BIN bin, FUNC func)
- uint32_t EPAX::BIN_countFunc (BIN bin)
- bool EPAX::BIN_isExecutable (BIN bin)
- uint32_t EPAX::BIN_fileSize (BIN bin)
- void EPAX::BIN_printStaticFile (BIN bin, std::string fname)
- FUNC EPAX::BIN_findFunc (BIN bin, uint64_t addr)
- FUNC EPAX::FUNC_create (uint8_t ∗bytes, uint32_t size)
- void EPAX::FUNC_destroy (FUNC func)

- void EPAX::FUNC_print (FUNC func)
- std::string EPAX::FUNC_name (FUNC func)
- uint32_t EPAX::FUNC_size (FUNC func)
- uint64_t EPAX::FUNC_addr (FUNC func)
- std::string EPAX::FUNC_secName (FUNC func)
- BIN EPAX::FUNC_bin (FUNC func)
- uint32_t EPAX::FUNC_countBbl (FUNC func)
- BBL EPAX::FUNC_findBbl (FUNC func, uint64_t addr)
- BBL EPAX::FUNC_firstBbl (FUNC func)
- BBL EPAX::FUNC_nextBbl (FUNC func, BBL bbl)
- bool EPAX::FUNC_isLastBbl (FUNC func, BBL bbl)
- uint32_t EPAX::FUNC_countInsn (FUNC func)
- INSN EPAX::FUNC_findInsn (FUNC func, uint64_t addr)
- INSN EPAX::FUNC_firstInsn (FUNC func)
- INSN EPAX::FUNC_nextInsn (FUNC func, INSN insn)
- bool EPAX::FUNC_isLastInsn (FUNC func, INSN insn)
- CFG EPAX::FUNC_cfg (FUNC func)
- uint32_t EPAX::FUNC_countTargets (FUNC func)
- uint32_t EPAX::FUNC_targets (FUNC func, std::vector< FUNC > &funcList)
- uint32_t EPAX::CFG_countLoop (CFG cfg)
- LOOP EPAX::CFG_findLoop (CFG cfg, uint64_t addr)
- LOOP EPAX::CFG_firstLoop (CFG cfg)
- LOOP EPAX::CFG_nextLoop (CFG cfg, LOOP loop)
- bool EPAX::CFG_isLastLoop (CFG cfg, LOOP loop)
- CFG EPAX::LOOP_cfg (LOOP loop)
- FUNC EPAX::LOOP_func (LOOP loop)
- uint32_t EPAX::LOOP_size (LOOP loop)
- uint32_t EPAX::LOOP_countBbl (LOOP loop)
- BBL EPAX::LOOP_findBbl (LOOP loop, uint64_t addr)
- BBL EPAX::LOOP_firstBbl (LOOP loop)
- BBL EPAX::LOOP_nextBbl (LOOP loop, BBL bbl)
- bool EPAX::LOOP_isLastBbl (LOOP loop, BBL bbl)
- uint32_t EPAX::LOOP_countInsn (LOOP loop)
- INSN EPAX::LOOP_findInsn (LOOP loop, uint64_t addr)
- INSN EPAX::LOOP_firstInsn (LOOP loop)
- INSN EPAX::LOOP_nextInsn (LOOP loop, INSN insn)
- bool EPAX::LOOP_isLastInsn (LOOP loop, INSN insn)
- BBL EPAX::LOOP_head (LOOP loop)
- BBL EPAX::LOOP_tail (LOOP loop)
- uint32_t EPAX::LOOP_countExits (LOOP loop)
- uint32_t EPAX::LOOP_exits (LOOP loop, std::vector< EPAX::INSN > &insnList)
- bool EPAX::LOOP_isInnerLoop (LOOP loop1, LOOP loop2)
- LOOP EPAX::LOOP_parent (LOOP loop)
- uint32_t EPAX::LOOP_index (LOOP loop)
- uint32_t EPAX::LOOP_depth (LOOP loop)
- bool EPAX::BBL_isHead (BBL bbl, INSN insn)

- bool EPAX::BBL_isTail (BBL bbl, INSN insn)
- INSN EPAX::BBL_head (BBL bbl)
- INSN EPAX::BBL_tail (BBL bbl)
- FUNC EPAX::BBL_func (BBL bbl)
- LOOP EPAX::BBL_loop (BBL bbl)
- uint32_t EPAX::BBL_size (BBL bbl)
- uint64_t EPAX::BBL_addr (BBL bbl)
- uint32_t EPAX::BBL_countInsn (BBL bbl)
- INSN EPAX::BBL_findInsn (BBL bbl, uint64_t addr)
- INSN EPAX::BBL_firstInsn (BBL bbl)
- INSN EPAX::BBL_nextInsn (BBL bbl, INSN insn)
- bool EPAX::BBL_isLastInsn (BBL bbl, INSN insn)
- uint32_t EPAX::BBL_countTargets (BBL bbl)
- uint32_t EPAX::BBL_targets (BBL bbl, std::vector< EPAX::BBL > &bblList)
- bool EPAX::BBL_hasFallthroughTarget (BBL bbl)
- BBL EPAX::BBL_fallthroughTarget (BBL bbl)
- uint32_t EPAX::BBL_countJumpTargets (BBL bbl)
- uint32_t EPAX::BBL_jumpTargets (BBL bbl, std::vector< EPAX::BBL > &bblList)
- uint32_t EPAX::BBL_countSources (BBL bbl)
- uint32_t EPAX::BBL_sources (BBL bbl, std::vector< EPAX::BBL > &bblList)
- uint32_t EPAX::INSN_targets (INSN insn, std::vector< uint64_t > &tlist)
- BBL EPAX::INSN_bbl (INSN insn)
- FUNC EPAX::INSN_func (INSN insn)
- LOOP EPAX::INSN_loop (INSN insn)
- uint64_t EPAX::INSN_addr (INSN insn)
- std::string EPAX::INSN_string (INSN insn)
- uint64_t EPAX::INSN_callTarget (INSN insn)
- bool EPAX::INSN_isBranch (INSN insn)
- bool EPAX::INSN_isFpop (INSN insn)
- bool EPAX::INSN_isMemop (INSN insn)
- uint32_t EPAX::INSN_size (INSN insn)
- std::string EPAX::INSN_condName (INSN insn)
- bool EPAX::INSN_fallsThrough (INSN insn)
- uint32_t EPAX::INSN_sourceRegisterSizeInBits (INSN insn)
- uint32_t EPAX::INSN_sourceDatatypeSizeInBits (INSN insn)
- EPAX_bin EPAX_bin_create (const char *fileName)
- const char * EPAX_bin_getName (EPAX_bin bin)
- void EPAX_bin_destroy (EPAX_bin bin)
- void EPAX_bin_run (EPAX_bin bin, int argc, char **argv)
- EPAX_func EPAX_bin_firstFunc (EPAX_bin bin)
- EPAX_func EPAX_bin_nextFunc (EPAX_bin bin, EPAX_func func)
- uint32_t EPAX_bin_isLastFunc (EPAX_bin bin, EPAX_func func)
- uint32_t EPAX_bin_countFunc (EPAX_bin bin)
- uint32_t EPAX_bin_isExecutable (EPAX_bin bin)
- uint32_t EPAX_bin_fileSize (EPAX_bin bin)
- void EPAX_bin_printStaticFile (EPAX_bin bin, const char *fname)

- EPAX_func EPAX_bin_fundFunc (EPAX_bin bin, uint64_t addr)
- EPAX_func EPAX_func_create (uint8_t *bytes, uint32_t size)
- void EPAX_func_destroy (EPAX_func func)
- void EPAX_func_print (EPAX_func func)
- const char * EPAX_func_name (EPAX_func func)
- uint32_t EPAX_func_size (EPAX_func func)
- uint64_t EPAX_func_addr (EPAX_func func)
- const char * EPAX_func_secName (EPAX_func func)
- EPAX_bin EPAX_func_bin (EPAX_func func)
- uint32_t EPAX_func_countBbl (EPAX_func func)
- EPAX_bbl EPAX_func_findBbl (EPAX_func func, uint64_t addr)
- EPAX_bbl EPAX_func_firstBbl (EPAX_func func)
- EPAX_bbl EPAX_func_nextBbl (EPAX_func func, EPAX_bbl bbl)
- uint32_t EPAX_func_isLastBbl (EPAX_func func, EPAX_bbl bbl)
- uint32_t EPAX_func_countInsn (EPAX_func func)
- EPAX_insn EPAX_func_findInsn (EPAX_func func, uint64_t addr)
- EPAX_insn EPAX_func_firstInsn (EPAX_func func)
- EPAX_insn EPAX_func_nextInsn (EPAX_func func, EPAX_insn insn)
- uint32_t EPAX_func_isLastInsn (EPAX_func func, EPAX_insn insn)
- EPAX_cfg EPAX_func_cfg (EPAX_func func)
- uint32_t EPAX_func_countTargets (EPAX_func func)
- uint32_t EPAX_func_targets (EPAX_func func, EPAX_func *funcList)
- uint32_t EPAX_cfg_countLoop (EPAX_cfg cfg)
- EPAX_loop EPAX_cfg_findLoop (EPAX_cfg cfg, uint64_t addr)
- EPAX_loop EPAX_cfg_firstLoop (EPAX_cfg cfg)
- EPAX_loop EPAX_cfg_nextLoop (EPAX_cfg cfg, EPAX_loop loop)
- uint32_t EPAX_cfg_isLastLoop (EPAX_cfg cfg, EPAX_loop loop)
- EPAX_cfg EPAX_loop_cfg (EPAX_loop loop)
- EPAX_func EPAX_loop_func (EPAX_loop loop)
- uint32_t EPAX_loop_size (EPAX_loop loop)
- uint32_t EPAX_loop_countBbl (EPAX_loop loop)
- EPAX_bbl EPAX_loop_findBbl (EPAX_loop loop, uint64_t addr)
- EPAX_bbl EPAX_loop_firstBbl (EPAX_loop loop)
- EPAX_bbl EPAX_loop_nextBbl (EPAX_loop loop, EPAX_bbl bbl)
- uint32_t EPAX_loop_isLastBbl (EPAX_loop loop, EPAX_bbl bbl)
- uint32_t EPAX_loop_countInsn (EPAX_loop loop)
- EPAX_insn EPAX_loop_findInsn (EPAX_loop loop, uint64_t addr)
- EPAX_insn EPAX_loop_firstInsn (EPAX_loop loop)
- EPAX_insn EPAX_loop_nextInsn (EPAX_loop loop, EPAX_insn insn)
- uint32_t EPAX_loop_isLastInsn (EPAX_loop loop, EPAX_insn insn)
- EPAX_bbl EPAX_loop_head (EPAX_loop loop)
- EPAX_bbl EPAX_loop_tail (EPAX_loop loop)
- uint32_t EPAX_loop_countExits (EPAX_loop loop)
- uint32_t EPAX_loop_exits (EPAX_loop loop, EPAX_insn *insnList)
- uint32_t EPAX_loop_isInnerLoop (EPAX_loop loop1, EPAX_loop loop2)
- EPAX_loop EPAX_loop_parent (EPAX_loop loop)

- uint32_t EPAX_loop_index (EPAX_loop loop)
- uint32_t EPAX_loop_depth (EPAX_loop loop)
- uint32_t EPAX_bbl_isHead (EPAX_bbl bbl, EPAX_insn insn)
- uint32_t EPAX_bbl_isTail (EPAX_bbl bbl, EPAX_insn insn)
- EPAX_insn EPAX_bbl_head (EPAX_bbl bbl)
- EPAX_insn EPAX_bbl_tail (EPAX_bbl bbl)
- EPAX_func EPAX_bbl_func (EPAX_bbl bbl)
- EPAX_loop EPAX_bbl_loop (EPAX_bbl bbl)
- uint32_t EPAX_bbl_size (EPAX_bbl bbl)
- uint64_t EPAX_bbl_addr (EPAX_bbl bbl)
- uint32_t EPAX_bbl_countInsn (EPAX_bbl bbl)
- EPAX_insn EPAX_bbl_findInsn (EPAX_bbl bbl, uint64_t addr)
- EPAX_insn EPAX_bbl_firstInsn (EPAX_bbl bbl)
- EPAX_insn EPAX_bbl_nextInsn (EPAX_bbl bbl, EPAX_insn insn)
- uint32_t EPAX_bbl_isLastInsn (EPAX_bbl bbl, EPAX_insn insn)
- uint32_t EPAX_bbl_countTargets (EPAX_bbl bbl)
- uint32_t EPAX_bbl_targets (EPAX_bbl bbl, EPAX_bbl *bblList)
- uint32_t EPAX_bbl_hasFallthroughTarget (EPAX_bbl bbl)
- EPAX_bbl EPAX_bbl_fallthroughTarget (EPAX_bbl bbl)
- uint32_t EPAX_bbl_countJumpTargets (EPAX_bbl bbl)
- uint32_t EPAX_bbl_jumpTargets (EPAX_bbl bbl, EPAX_bbl *bblList)
- uint32_t EPAX_bbl_countSources (EPAX_bbl bbl)
- uint32_t EPAX_bbl_sources (EPAX_bbl bbl, EPAX_bbl *bblList)
- uint32_t EPAX_insn_targets (EPAX_insn insn, uint64_t *tlist)
- EPAX_bbl EPAX_insn_bbl (EPAX_insn insn)
- EPAX_func EPAX_insn_func (EPAX_insn insn)
- EPAX_loop EPAX_insn_loop (EPAX_insn insn)
- uint64_t EPAX_insn_addr (EPAX_insn insn)
- const char * EPAX_insn_string (EPAX_insn insn)
- uint64_t EPAX_insn_callTarget (EPAX_insn insn)
- uint32_t EPAX_insn_isBranch (EPAX_insn insn)
- uint32_t EPAX_insn_isFpop (EPAX_insn insn)
- uint32_t EPAX_insn_isMemop (EPAX_insn insn)
- uint32_t EPAX_insn_size (EPAX_insn insn)
- const char * EPAX_insn_condName (EPAX_insn insn)
- uint32_t EPAX_insn_fallsThrough (EPAX_insn insn)
- uint32_t EPAX_insn_sourceRegisterSizeInBits (EPAX_insn insn)
- uint32_t EPAX_insn_sourceDatatypeSizeInBits (EPAX_insn insn)

7.21.1 Detailed Description

7.21.2 LICENSE

This file is part of the EPAX toolkit.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file Interface.cpp.

7.21.3 Function Documentation

7.21.3.1 uint64_t EPAX_bbl_addr (EPAX_bbl *bbl*)

Definition at line 1085 of file Interface.cpp.

7.21.3.2 uint32_t EPAX_bbl_countInsn (EPAX_bbl *bbl*)

Definition at line 1089 of file Interface.cpp.

7.21.3.3 uint32_t EPAX_bbl_countJumpTargets (EPAX_bbl *bbl*)

Definition at line 1133 of file Interface.cpp.

7.21.3.4 uint32_t EPAX_bbl_countSources (EPAX_bbl *bbl*)

Definition at line 1149 of file Interface.cpp.

7.21.3.5 uint32_t EPAX_bbl_countTargets (EPAX_bbl *bbl*)

Definition at line 1109 of file Interface.cpp.

7.21.3.6 EPAX_bbl EPAX_bbl_fallthroughTarget (EPAX_bbl *bbl*)

Definition at line 1129 of file Interface.cpp.

7.21.3.7 EPAX_insn EPAX_bbl_findInsn (EPAX_bbl *bbl,* uint64_t *addr*)

Definition at line 1093 of file Interface.cpp.

7.21.3.8 EPAX_insn EPAX_bbl_firstInsn (EPAX_bbl *bbl*)

Definition at line 1097 of file Interface.cpp.

7.21.3.9 EPAX_func EPAX_bbl_func (EPAX_bbl *bbl*)

Definition at line 1073 of file Interface.cpp.

7.21.3.10 uint32_t EPAX_bbl_hasFallthroughTarget (EPAX_bbl *bbl*)

Definition at line 1125 of file Interface.cpp.

7.21.3.11 EPAX_insn EPAX_bbl_head (EPAX_bbl *bbl*)

Definition at line 1065 of file Interface.cpp.

7.21.3.12 uint32_t EPAX_bbl_isHead (EPAX_bbl *bbl,* EPAX_insn *insn*)

Definition at line 1057 of file Interface.cpp.

7.21.3.13 uint32_t EPAX_bbl_isLastInsn (EPAX_bbl *bbl,* EPAX_insn *insn*)

Definition at line 1105 of file Interface.cpp.

7.21.3.14 uint32_t EPAX_bbl_isTail (EPAX_bbl *bbl,* EPAX_insn *insn*)

Definition at line 1061 of file Interface.cpp.

7.21.3.15 uint32_t EPAX_bbl_jumpTargets (EPAX_bbl *bbl,* EPAX_bbl * *bblList*)

Definition at line 1137 of file Interface.cpp.

7.21.3.16 EPAX_loop EPAX_bbl_loop (EPAX_bbl *bbl*)

Definition at line 1077 of file Interface.cpp.

7.21.3.17 EPAX_insn EPAX_bbl_nextInsn (EPAX_bbl *bbl,* EPAX_insn *insn*)

Definition at line 1101 of file Interface.cpp.

7.21.3.18 uint32_t EPAX_bbl_size (EPAX_bbl *bbl*)

Definition at line 1081 of file Interface.cpp.

7.21.3.19 uint32_t EPAX_bbl_sources (EPAX_bbl *bbl,* EPAX_bbl * *bblList*)

Definition at line 1153 of file Interface.cpp.

7.21.3.20 EPAX_insn EPAX_bbl_tail (EPAX_bbl *bbl*)

Definition at line 1069 of file Interface.cpp.

7.21.3.21 uint32_t EPAX_bbl_targets (EPAX_bbl *bbl,* EPAX_bbl * *bblList*)

Definition at line 1113 of file Interface.cpp.

7.21.3.22 uint32_t EPAX_bin_countFunc (EPAX_bin *bin*)

Definition at line 830 of file Interface.cpp.

7.21.3.23 EPAX_bin EPAX_bin_create (const char * *fileName*)

Definition at line 800 of file Interface.cpp.

7.21.3.24 void EPAX_bin_destroy (EPAX_bin *bin*)

Definition at line 810 of file Interface.cpp.

7.21.3.25 uint32_t EPAX_bin_fileSize (EPAX_bin *bin*)

Definition at line 838 of file Interface.cpp.

7.21.3.26 EPAX_func EPAX_bin_firstFunc (EPAX_bin *bin*)

Definition at line 818 of file Interface.cpp.

7.21.3.27 EPAX_func EPAX_bin_fundFunc (EPAX_bin *bin,* uint64_t *addr*)

Definition at line 847 of file Interface.cpp.

7.21.3.28 const char* EPAX_bin_getName (EPAX_bin *bin*)

Definition at line 805 of file Interface.cpp.

7.21.3.29 uint32_t EPAX_bin_isExecutable (EPAX_bin *bin*)

Definition at line 834 of file Interface.cpp.

7.21.3.30 uint32_t EPAX_bin_isLastFunc (EPAX_bin *bin,* EPAX_func *func*)

Definition at line 826 of file Interface.cpp.

7.21.3.31 EPAX_func EPAX_bin_nextFunc (EPAX_bin *bin,* EPAX_func *func*)

Definition at line 822 of file Interface.cpp.

7.21.3.32 void EPAX_bin_printStaticFile (EPAX_bin *bin,* const char * *fname*)

Definition at line 842 of file Interface.cpp.

7.21.3.33 void EPAX_bin_run (EPAX_bin *bin,* int *argc,* char ** *argv*)

Definition at line 814 of file Interface.cpp.

7.21.3.34 uint32_t EPAX_cfg_countLoop (EPAX_cfg *cfg*)

Definition at line 945 of file Interface.cpp.

7.21.3.35 EPAX_loop EPAX_cfg_findLoop (EPAX_cfg *cfg,* uint64_t *addr*)

Definition at line 949 of file Interface.cpp.

7.21.3.36 EPAX_loop EPAX_cfg_firstLoop (EPAX_cfg *cfg*)

Definition at line 953 of file Interface.cpp.

7.21.3.37 uint32_t EPAX_cfg_isLastLoop (EPAX_cfg *cfg,* EPAX_loop *loop*)

Definition at line 961 of file Interface.cpp.

7.21.3.38 EPAX_loop EPAX_cfg_nextLoop (EPAX_cfg *cfg,* EPAX_loop *loop*)

Definition at line 957 of file Interface.cpp.

7.21.3.39 uint64_t EPAX_func_addr (EPAX_func *func*)

Definition at line 872 of file Interface.cpp.

7.21.3.40 EPAX_bin EPAX_func_bin (EPAX_func *func*)

Definition at line 881 of file Interface.cpp.

7.21.3.41 EPAX_cfg EPAX_func_cfg (EPAX_func *func*)

Definition at line 925 of file Interface.cpp.

7.21.3.42 uint32_t EPAX_func_countBbl (EPAX_func *func*)

Definition at line 885 of file Interface.cpp.

7.21.3.43 uint32_t EPAX_func_countInsn (EPAX_func *func*)

Definition at line 905 of file Interface.cpp.

7.21.3.44 uint32_t EPAX_func_countTargets (EPAX_func *func*)

Definition at line 929 of file Interface.cpp.

7.21.3.45 EPAX_func EPAX_func_create (uint8_t * *bytes,* uint32_t *size*)

Definition at line 851 of file Interface.cpp.

7.21.3.46 void EPAX_func_destroy (EPAX_func *func*)

Definition at line 855 of file Interface.cpp.

7.21.3.47 EPAX_bbl EPAX_func_findBbl (EPAX_func *func,* uint64_t *addr*)

Definition at line 889 of file Interface.cpp.

7.21.3.48 EPAX_insn EPAX_func_findInsn (EPAX_func *func,* uint64_t *addr*)

Definition at line 909 of file Interface.cpp.

7.21.3.49 EPAX_bbl EPAX_func_firstBbl (EPAX_func *func*)

Definition at line 893 of file Interface.cpp.

7.21.3.50 EPAX_insn EPAX_func_firstInsn (EPAX_func *func*)

Definition at line 913 of file Interface.cpp.

7.21.3.51 uint32_t EPAX_func_isLastBbl (EPAX_func *func,* EPAX_bbl *bbl*)

Definition at line 901 of file Interface.cpp.

7.21.3.52 uint32_t EPAX_func_isLastInsn (EPAX_func *func,* EPAX_insn *insn*)

Definition at line 921 of file Interface.cpp.

7.21.3.53 const char* EPAX_func_name (EPAX_func *func*)

Definition at line 863 of file Interface.cpp.

7.21.3.54 EPAX_bbl EPAX_func_nextBbl (EPAX_func *func,* EPAX_bbl *bbl*)

Definition at line 897 of file Interface.cpp.

7.21.3.55 EPAX_insn EPAX_func_nextInsn (EPAX_func *func,* EPAX_insn *insn*)

Definition at line 917 of file Interface.cpp.

7.21.3.56 void EPAX_func_print (EPAX_func *func*)

Definition at line 859 of file Interface.cpp.

7.21.3.57 const char* EPAX_func_secName (EPAX_func *func*)

Definition at line 876 of file Interface.cpp.

7.21.3.58 uint32_t EPAX_func_size (EPAX_func *func*)

Definition at line 868 of file Interface.cpp.

7.21.3.59 uint32_t EPAX_func_targets (EPAX_func *func,* EPAX_func * *funcList*)

Definition at line 933 of file Interface.cpp.

7.21.3.60 uint64_t EPAX_insn_addr (EPAX_insn *insn*)

Definition at line 1189 of file Interface.cpp.

7.21.3.61 EPAX_bbl EPAX_insn_bbl (EPAX_insn *insn*)

Definition at line 1177 of file Interface.cpp.

7.21.3.62 uint64_t EPAX_insn_callTarget (EPAX_insn *insn*)

Definition at line 1198 of file Interface.cpp.

7.21.3.63 const char* EPAX_insn_condName (EPAX_insn *insn*)

Definition at line 1218 of file Interface.cpp.

7.21.3.64 uint32_t EPAX_insn_fallsThrough (EPAX_insn *insn*)

Definition at line 1223 of file Interface.cpp.

7.21.3.65 EPAX_func EPAX_insn_func (EPAX_insn *insn*)

Definition at line 1181 of file Interface.cpp.

7.21.3.66 uint32_t EPAX_insn_isBranch (EPAX_insn *insn*)

Definition at line 1202 of file Interface.cpp.

7.21.3.67 uint32_t EPAX_insn_isFpop (EPAX_insn *insn*)

Definition at line 1206 of file Interface.cpp.

7.21.3.68 uint32_t EPAX_insn_isMemop (EPAX_insn *insn*)

Definition at line 1210 of file Interface.cpp.

7.21.3.69 EPAX_loop EPAX_insn_loop (EPAX_insn *insn*)

Definition at line 1185 of file Interface.cpp.

7.21.3.70 uint32_t EPAX_insn_size (EPAX_insn *insn*)

Definition at line 1214 of file Interface.cpp.

7.21.3.71 uint32_t EPAX_insn_sourceDatatypeSizeInBits (EPAX_insn *insn*)

Definition at line 1231 of file Interface.cpp.

7.21.3.72 uint32_t EPAX_insn_sourceRegisterSizeInBits (EPAX_insn *insn*)

Definition at line 1227 of file Interface.cpp.

7.21.3.73 const char* EPAX_insn_string (EPAX_insn *insn*)

Definition at line 1193 of file Interface.cpp.

7.21.3.74 uint32_t EPAX_insn_targets (EPAX_insn *insn*, uint64_t * *tlist*)

Definition at line 1165 of file Interface.cpp.

7.21.3.75 EPAX_cfg EPAX_loop_cfg (EPAX_loop *loop*)

Definition at line 965 of file Interface.cpp.

7.21.3.76 uint32_t EPAX_loop_countBbl (EPAX_loop *loop*)

Definition at line 977 of file Interface.cpp.

7.21.3.77 uint32_t EPAX_loop_countExits (EPAX_loop *loop*)

Definition at line 1025 of file Interface.cpp.

7.21.3.78 uint32_t EPAX_loop_countInsn (EPAX_loop *loop*)

Definition at line 997 of file Interface.cpp.

7.21.3.79 uint32_t EPAX_loop_depth (EPAX_loop *loop*)

Definition at line 1053 of file Interface.cpp.

7.21.3.80 uint32_t EPAX_loop_exits (EPAX_loop *loop,* EPAX_insn * *insnList*)

Definition at line 1029 of file Interface.cpp.

7.21.3.81 EPAX_bbl EPAX_loop_findBbl (EPAX_loop *loop,* uint64_t *addr*)

Definition at line 981 of file Interface.cpp.

7.21.3.82 EPAX_insn EPAX_loop_findInsn (EPAX_loop *loop,* uint64_t *addr*)

Definition at line 1001 of file Interface.cpp.

7.21.3.83 EPAX_bbl EPAX_loop_firstBbl (EPAX_loop *loop*)

Definition at line 985 of file Interface.cpp.

7.21.3.84 EPAX_insn EPAX_loop_firstInsn (EPAX_loop *loop*)

Definition at line 1005 of file Interface.cpp.

7.21.3.85 EPAX_func EPAX_loop_func (EPAX_loop *loop*)

Definition at line 969 of file Interface.cpp.

7.21.3.86 EPAX_bbl EPAX_loop_head (EPAX_loop *loop*)

Definition at line 1017 of file Interface.cpp.

7.21.3.87 uint32_t EPAX_loop_index (EPAX_loop *loop*)

Definition at line 1049 of file Interface.cpp.

7.21.3.88 uint32_t EPAX_loop_isInnerLoop (EPAX_loop *loop1,* EPAX_loop *loop2*)

Definition at line 1041 of file Interface.cpp.

7.21.3.89 uint32_t EPAX_loop_isLastBbl (EPAX_loop *loop,* EPAX_bbl *bbl*)

Definition at line 993 of file Interface.cpp.

7.21.3.90 uint32_t EPAX_loop_isLastInsn (EPAX_loop *loop,* EPAX_insn *insn*)

Definition at line 1013 of file Interface.cpp.

7.21.3.91 EPAX_bbl EPAX_loop_nextBbl (EPAX_loop *loop,* EPAX_bbl *bbl*)

Definition at line 989 of file Interface.cpp.

7.21.3.92 EPAX_insn EPAX_loop_nextInsn (EPAX_loop *loop,* EPAX_insn *insn*)

Definition at line 1009 of file Interface.cpp.

7.21.3.93 EPAX_loop EPAX_loop_parent (EPAX_loop *loop*)

Definition at line 1045 of file Interface.cpp.

7.21.3.94 uint32_t EPAX_loop_size (EPAX_loop *loop*)

Definition at line 973 of file Interface.cpp.

7.21.3.95 EPAX_bbl EPAX_loop_tail (EPAX_loop *loop*)

Definition at line 1021 of file Interface.cpp.

7.22 Interface.hpp File Reference

```
#include <stdint.h> #include <string> #include <vector>
```

Namespaces

- namespace EPAX

Typedefs

- typedef Binary * EPAX::BIN
- typedef Section * EPAX::SECT
- typedef Function * EPAX::FUNC
- typedef ControlFlow * EPAX::CFG
- typedef Loop * EPAX::LOOP
- typedef BasicBlock * EPAX::BBL
- typedef Instruction * EPAX::INSN
- typedef Symbol * EPAX::SYM
- typedef FlowEquation * EPAX::FLOW

Functions

- BIN EPAX::BIN_create (std::string fileName)
- std::string EPAX::BIN_getName (BIN bin)
- void EPAX::BIN_destroy (BIN bin)
- void EPAX::BIN_run (BIN bin, int argc, char **argv)
- FUNC EPAX::BIN_firstFunc (BIN bin)
- FUNC EPAX::BIN_nextFunc (BIN bin, FUNC func)
- bool EPAX::BIN_isLastFunc (BIN bin, FUNC func)
- uint32_t EPAX::BIN_countFunc (BIN bin)
- bool EPAX::BIN_isExecutable (BIN bin)
- uint32_t EPAX::BIN_fileSize (BIN bin)
- void EPAX::BIN_printStaticFile (BIN bin, std::string fname)
- FUNC EPAX::BIN_findFunc (BIN bin, uint64_t addr)
- FUNC EPAX::FUNC_create (uint8_t *bytes, uint32_t size)
- void EPAX::FUNC_Destroy (FUNC func)
- void EPAX::FUNC_print (FUNC func)
- std::string EPAX::FUNC_name (FUNC func)
- uint32_t EPAX::FUNC_size (FUNC func)
- uint64_t EPAX::FUNC_addr (FUNC func)
- std::string EPAX::FUNC_secName (FUNC func)
- BIN EPAX::FUNC_bin (FUNC func)
- uint32_t EPAX::FUNC_countBbl (FUNC func)
- BBL EPAX::FUNC_findBbl (FUNC func, uint64_t addr)
- BBL EPAX::FUNC_firstBbl (FUNC func)
- BBL EPAX::FUNC_nextBbl (FUNC func, BBL bbl)
- bool EPAX::FUNC_isLastBbl (FUNC func, BBL bbl)
- uint32_t EPAX::FUNC_countInsn (FUNC func)
- INSN EPAX::FUNC_findInsn (FUNC func, uint64_t addr)
- INSN EPAX::FUNC_firstInsn (FUNC func)

- INSN EPAX::FUNC_nextInsn (FUNC func, INSN insn)
- bool EPAX::FUNC_isLastInsn (FUNC func, INSN insn)
- CFG EPAX::FUNC_cfg (FUNC func)
- uint32_t EPAX::FUNC_countTargets (FUNC func)
- uint32_t EPAX::FUNC_targets (FUNC func, std::vector< FUNC > &funcList)
- uint32_t EPAX::CFG_countLoop (CFG cfg)
- LOOP EPAX::CFG_findLoop (CFG cfg, uint64_t addr)
- LOOP EPAX::CFG_firstLoop (CFG cfg)
- LOOP EPAX::CFG_nextLoop (CFG cfg, LOOP loop)
- bool EPAX::CFG_isLastLoop (CFG cfg, LOOP loop)
- CFG EPAX::LOOP_cfg (LOOP loop)
- FUNC EPAX::LOOP_func (LOOP loop)
- uint32_t EPAX::LOOP_size (LOOP loop)
- uint32_t EPAX::LOOP_countBbl (LOOP loop)
- BBL EPAX::LOOP_findBbl (LOOP loop, uint64_t addr)
- BBL EPAX::LOOP_firstBbl (LOOP loop)
- BBL EPAX::LOOP_nextBbl (LOOP loop, BBL bbl)
- bool EPAX::LOOP_isLastBbl (LOOP loop, BBL bbl)
- uint32_t EPAX::LOOP_countInsn (LOOP loop)
- INSN EPAX::LOOP_findInsn (LOOP loop, uint64_t addr)
- INSN EPAX::LOOP_firstInsn (LOOP loop)
- INSN EPAX::LOOP_nextInsn (LOOP loop, INSN insn)
- bool EPAX::LOOP_isLastInsn (LOOP loop, INSN insn)
- BBL EPAX::LOOP_head (LOOP loop)
- BBL EPAX::LOOP_tail (LOOP loop)
- uint32_t EPAX::LOOP_countExits (LOOP loop)
- uint32_t EPAX::LOOP_exits (LOOP loop, std::vector< INSN > &insnList)
- bool EPAX::LOOP_isInnerLoop (LOOP loop1, LOOP loop2)
- LOOP EPAX::LOOP_parent (LOOP loop)
- uint32_t EPAX::LOOP_index (LOOP loop)
- uint32_t EPAX::LOOP_depth (LOOP loop)
- bool EPAX::BBL_isHead (BBL bbl, INSN insn)
- bool EPAX::BBL_isTail (BBL bbl, INSN insn)
- INSN EPAX::BBL_head (BBL bbl)
- INSN EPAX::BBL_tail (BBL bbl)
- FUNC EPAX::BBL_func (BBL bbl)
- LOOP EPAX::BBL_loop (BBL bbl)
- uint32_t EPAX::BBL_size (BBL bbl)
- uint64_t EPAX::BBL_addr (BBL bbl)
- uint32_t EPAX::BBL_countInsn (BBL bbl)
- INSN EPAX::BBL_findInsn (BBL bbl, uint64_t addr)
- INSN EPAX::BBL_firstInsn (BBL bbl)
- INSN EPAX::BBL_nextInsn (BBL bbl, INSN insn)
- bool EPAX::BBL_isLastInsn (BBL bbl, INSN insn)
- uint32_t EPAX::BBL_countTargets (BBL bbl)
- uint32_t EPAX::BBL_targets (BBL bbl, std::vector< BBL > &bblList)

- bool EPAX::BBL_hasFallthroughTarget (BBL bbl)
- BBL EPAX::BBL_fallthroughTarget (BBL bbl)
- uint32_t EPAX::BBL_countJumpTargets (BBL bbl)
- uint32_t EPAX::BBL_jumpTargets (BBL bbl, std::vector< BBL > &bblList)
- uint32_t EPAX::BBL_countSources (BBL bbl)
- uint32_t EPAX::BBL_sources (BBL bbl, std::vector< BBL > &bblList)
- uint32_t EPAX::INSN_targets (INSN insn, std::vector< uint64_t > &tlist)
- BBL EPAX::INSN_bbl (INSN insn)
- FUNC EPAX::INSN_func (INSN insn)
- LOOP EPAX::INSN_loop (INSN insn)
- uint64_t EPAX::INSN_addr (INSN insn)
- std::string EPAX::INSN_string (INSN insn)
- uint64_t EPAX::INSN_callTarget (INSN insn)
- bool EPAX::INSN_isBranch (INSN insn)
- bool EPAX::INSN_isFpop (INSN insn)
- bool EPAX::INSN_isMemop (INSN insn)
- uint32_t EPAX::INSN_size (INSN insn)
- std::string EPAX::INSN_condName (INSN insn)
- bool EPAX::INSN_fallsThrough (INSN insn)
- uint32_t EPAX::INSN_sourceRegisterSizeInBits (INSN insn)
- uint32_t EPAX::INSN_sourceDatatypeSizeInBits (INSN insn)

7.22.1 Detailed Description

7.22.2 LICENSE

This file is part of the EPAX toolkit.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file Interface.hpp.

7.23 Loop.cpp File Reference

```
#include "EPAXCommonInternal.hpp"  #include "BasicBlock.-
hpp" #include "ControlFlow.hpp" #include "Function.hpp" ×
#include "Loop.hpp"
```

Namespaces

- namespace EPAX

7.23.1 Detailed Description

7.23.2 LICENSE

This file is part of the EPAX toolkit.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file Loop.cpp.

7.24 Loop.hpp File Reference

```
#include "DataStruct.hpp" #include "BaseClass.hpp"
```

Classes

- class EPAX::Loop

Namespaces

- namespace EPAX

7.24.1 Detailed Description

7.24.2 LICENSE

This file is part of the EPAX toolkit.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file Loop.hpp.

7.25 MachOBinary.cpp File Reference

```
#include "EPAXCommonInternal.hpp" #include "MachO/loader.-
h"  #include "InputFile.hpp"  #include "MachOBinary.hpp" ×
#include "BaseClass.hpp"
```

Namespaces

- namespace EPAX
- namespace EPAX::MachO

Defines

- #define MHDR32_ENTRY ((mach_header∗)entry)
- #define MHDR64_ENTRY ((mach_header_64∗)entry)
- #define SCASE(__typ__) case CPU_SUBTYPE_ARM_ ## __typ__: std::cout << "isa=ARM_" << #__typ__; break
- #define CCASE(__typ__) case CPU_TYPE_ ## __typ__: std::cout << "isa=" << #__typ__; break

7.25.1 Detailed Description

7.25.2 LICENSE

This file is part of the EPAX toolkit.

Copyright (c) 2013, EP Analytics, Inc. All rights reserved.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file MachOBinary.cpp.

7.25.3 Define Documentation

7.25.3.1 #define CCASE(__*typ*__) case CPU_TYPE_ ## __typ__: std::cout << "isa=" << #__typ__; break

7.25.3.2 #define MHDR32_ENTRY ((mach_header*)entry)

Definition at line 37 of file MachOBinary.cpp.

7.25.3.3 #define MHDR64_ENTRY ((mach_header_64*)entry)

Definition at line 38 of file MachOBinary.cpp.

7.25.3.4 #define SCASE(__*typ*__) case CPU_SUBTYPE_ARM_ ## __typ__: std::cout << "isa=ARM_" << #__typ__; break

7.26 MachOBinary.hpp File Reference

```
#include "BaseClass.hpp"
```

Classes

- class EPAX::MachO::MachOBinary
- class EPAX::MachO::MachOBinary32
- class EPAX::MachO::MachOBinary64
- class EPAX::MachO::MachHeader
- class EPAX::MachO::MachHeader32
- class EPAX::MachO::MachHeader64

Namespaces

- namespace EPAX
- namespace EPAX::MachO

7.26.1 Detailed Description

7.26.2 LICENSE

This file is part of the EPAX toolkit.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file MachOBinary.hpp.

7.27 Section.cpp File Reference

```
#include "EPAXCommonInternal.hpp" #include "Section.hpp"
```

Namespaces

- namespace EPAX

7.27.1 Detailed Description

7.27.2 LICENSE

This file is part of the EPAX toolkit.

Copyright (c) 2013, EP Analytics, Inc. All rights reserved.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file Section.cpp.

7.28 Section.hpp File Reference

```
#include "BaseClass.hpp"
```

Classes

- class EPAX::Section

Namespaces

- namespace EPAX

7.28.1 Detailed Description

7.28.2 LICENSE

This file is part of the EPAX toolkit.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file Section.hpp.

7.29 Symbol.cpp File Reference

```
#include "EPAXCommonInternal.hpp" #include "Symbol.hpp"
```

Namespaces

- namespace EPAX

7.29.1 Detailed Description

7.29.2 LICENSE

This file is part of the EPAX toolkit.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file Symbol.cpp.

7.30 Symbol.hpp File Reference

```
#include "Binary.hpp" #include "BaseClass.hpp" #include "-
Section.hpp"
```

Classes

- class EPAX::Symbol
- class EPAX::SymbolTable
- class EPAX::StringTable

Namespaces

- namespace EPAX

7.30.1 Detailed Description

7.30.2 LICENSE

This file is part of the EPAX toolkit.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Definition in file Symbol.hpp.