Network Scanner

制作者 Doxygen 1.13.2

1 Network Scanner 项目文档 **1**
1.1 简介 . . . 1
1.2 主要功能 . . . 1
1.3 项目架构 . . . 1
1.4 构建说明 . . . 1

2 网络扫描器 **3**
2.1 主要功能 . . . 3
2.2 最新改进 . . . 4
2.3 构建要求 . . . 4
2.4 构建步骤 . . . 4
2.5 详细使用指南 . . . 4
2.5.1 基本扫描 . . . 4
2.5.2 网络拓扑 . . . 5
2.5.3 设备分析 . . . 5
2.5.4 扫描历史 . . . 5
2.5.5 计划任务 . . . 6
2.5.6 结果过滤 . . . 6
2.5.7 暗色模式 . . . 6
2.6 MAC地址厂商识别 . . . 6
2.7 技术细节 . . . 7
2.8 故障排除 . . . 7
2.8.1 常见问题 . . . 7
2.8.2 性能优化 . . . 7
2.9 开发注意事项 . . . 8
2.9.1 QtCharts命名空间 . . . 8
2.9.2 内存管理和界面初始化 . . . 8
2.10 版本历史 . . . 8
2.10.1 v2.2.0 . . . 8
2.10.2 v2.1.0 . . . 8
2.10.3 v2.0.0 . . . 9
2.10.4 v1.0.0 . . . 9

3 命名空间索引 **11**
3.1 命名空间列表 . . . 11

4 继承关系索引 **13**
4.1 类继承关系 . . . 13

5 类索引 **15**
5.1 类列表 . . . 15

6 文件索引 **17**
6.1 文件列表 . . . 17

7 命名空间文档 **19**

7.1 QT_WARNING_DISABLE_DEPRECATED 命名空间参考 19

8 类说明 **21**

8.1 ConnectionLine类 参考 21

8.1.1 构造及析构函数说明 23

8.1.1.1 ConnectionLine() 23

8.1.2 成员函数说明 23

8.1.2.1 boundingRect() 23

8.1.2.2 connectionType() 23

8.1.2.3 paint() 23

8.1.2.4 setConnectionType() 23

8.1.2.5 updatePosition() 24

8.1.3 类成员变量说明 24

8.1.3.1 m_connectionType 24

8.1.3.2 m_source 24

8.1.3.3 m_target 24

8.2 DeviceAnalyzer类 参考 24

8.2.1 构造及析构函数说明 27

8.2.1.1 DeviceAnalyzer() 27

8.2.2 成员函数说明 28

8.2.2.1 analysisCompleted 28

8.2.2.2 analyzeHosts() 28

8.2.2.3 clear() 28

8.2.2.4 createDeviceTypeChart() 29

8.2.2.5 createPortDistributionChart() 29

8.2.2.6 createVendorDistributionChart() 29

8.2.2.7 determineDeviceType() 30

8.2.2.8 generateSecurityReport() 30

8.2.2.9 getDeviceTypeChart() 30

8.2.2.10 getPortDistributionChart() 30

8.2.2.11 getReachableHostsCount() 30

8.2.2.12 getTotalHostsCount() 31

8.2.2.13 getUnreachableHostsCount() 31

8.2.2.14 getVendorDistributionChart() 31

8.2.3 类成员变量说明 31

8.2.3.1 m_deviceTypeChart 31

8.2.3.2 m_deviceTypeSeries 31

8.2.3.3 m_portDistributionChart 31

8.2.3.4 m_portSeries 31

8.2.3.5 m_reachableHosts 31

8.2.3.6 m_totalHosts 31

8.2.3.7 m_vendorDistributionChart . . . 31
8.2.3.8 m_vendorSeries . . . 32
8.3 DeviceNode类 参考 . . . 32
8.3.1 构造及析构函数说明 . . . 35
8.3.1.1 DeviceNode() . . . 35
8.3.2 成员函数说明 . . . 35
8.3.2.1 boundingRect() . . . 35
8.3.2.2 deviceType() . . . 35
8.3.2.3 hostInfo() . . . 36
8.3.2.4 hoverEnterEvent() . . . 36
8.3.2.5 hoverLeaveEvent() . . . 36
8.3.2.6 ipAddress() . . . 36
8.3.2.7 mouseMoveEvent() . . . 36
8.3.2.8 mousePressEvent() . . . 36
8.3.2.9 mouseReleaseEvent() . . . 36
8.3.2.10 networkLayer() . . . 36
8.3.2.11 paint() . . . 37
8.3.2.12 setDeviceType() . . . 37
8.3.2.13 setNetworkLayer() . . . 37
8.3.2.14 setPosition() . . . 37
8.3.2.15 setSubnetGroup() . . . 37
8.3.2.16 subnetGroup() . . . 38
8.3.3 类成员变量说明 . . . 38
8.3.3.1 m_dragStartPosition . . . 38
8.3.3.2 m_highlight . . . 38
8.3.3.3 m_host . . . 38
8.3.3.4 m_networkLayer . . . 38
8.3.3.5 m_subnetGroup . . . 38
8.3.3.6 m_type . . . 38
8.4 HostInfo结构体 参考 . . . 39
8.4.1 详细描述 . . . 39
8.4.2 类成员变量说明 . . . 40
8.4.2.1 hostName . . . 40
8.4.2.2 ipAddress . . . 40
8.4.2.3 isReachable . . . 40
8.4.2.4 macAddress . . . 40
8.4.2.5 macVendor . . . 40
8.4.2.6 openPorts . . . 40
8.4.2.7 scanTime . . . 40
8.5 MainWindow类 参考 . . . 41
8.5.1 构造及析构函数说明 . . . 45
8.5.1.1 MainWindow() . . . 45

8.5.1.2 ~MainWindow() . . . 46
8.5.2 成员函数说明 . . . 46
8.5.2.1 applySettings . . . 46
8.5.2.2 applyTheme() . . . 46
8.5.2.3 clearFilters . . . 47
8.5.2.4 clearResults . . . 47
8.5.2.5 compareScanResults . . . 47
8.5.2.6 createDetailsTab() . . . 48
8.5.2.7 createHistoryTab() . . . 48
8.5.2.8 createMenus() . . . 48
8.5.2.9 createSecurityTab() . . . 49
8.5.2.10 createSettingsDialog() . . . 49
8.5.2.11 createStatisticsTab() . . . 50
8.5.2.12 createTopologyTab() . . . 50
8.5.2.13 createUI() . . . 51
8.5.2.14 exportToCSV . . . 51
8.5.2.15 filterResults . . . 52
8.5.2.16 generateSecurityReport . . . 52
8.5.2.17 loadHistoryFromFile . . . 53
8.5.2.18 loadSettings() . . . 53
8.5.2.19 onHostFound . . . 53
8.5.2.20 onScanError . . . 54
8.5.2.21 onScanFinished . . . 54
8.5.2.22 onScanProgress . . . 54
8.5.2.23 onScanStarted . . . 55
8.5.2.24 onThemeChanged . . . 55
8.5.2.25 refreshTopology . . . 55
8.5.2.26 saveHistoryToFile . . . 55
8.5.2.27 saveResults . . . 56
8.5.2.28 saveSettings() . . . 56
8.5.2.29 saveTopologyImage . . . 56
8.5.2.30 scheduleScan . . . 57
8.5.2.31 setupConnections() . . . 57
8.5.2.32 showAbout . . . 59
8.5.2.33 showHistoryView . . . 59
8.5.2.34 showHostDetails . . . 59
8.5.2.35 showSettings . . . 59
8.5.2.36 showStatisticsView . . . 59
8.5.2.37 showTopologyView . . . 60
8.5.2.38 startScan . . . 60
8.5.2.39 stopScan . . . 60
8.5.2.40 toggleDarkMode . . . 61

8.5.2.41 togglePortScanOptions . . . 61
8.5.2.42 toggleRangeOptions . . . 61
8.5.2.43 updateNetworkTopology . . . 62
8.5.2.44 updatePortsList() . . . 62
8.5.2.45 updateStatistics() . . . 62
8.5.3 类成员变量说明 . . . 62
8.5.3.1 m_aboutAction . . . 62
8.5.3.2 m_centralWidget . . . 62
8.5.3.3 m_clearButton . . . 62
8.5.3.4 m_clearFilterButton . . . 63
8.5.3.5 m_controlLayout . . . 63
8.5.3.6 m_currentHostIndex . . . 63
8.5.3.7 m_customPortsCheckBox . . . 63
8.5.3.8 m_customRangeCheckBox . . . 63
8.5.3.9 m_darkModeAction . . . 63
8.5.3.10 m_darkModeEnabled . . . 63
8.5.3.11 m_detailsLayout . . . 63
8.5.3.12 m_detailsTab . . . 63
8.5.3.13 m_detailsTextEdit . . . 63
8.5.3.14 m_deviceAnalyzer . . . 64
8.5.3.15 m_deviceTypeChartView . . . 64
8.5.3.16 m_endIPLineEdit . . . 64
8.5.3.17 m_exitAction . . . 64
8.5.3.18 m_exportAction . . . 64
8.5.3.19 m_fileMenu . . . 64
8.5.3.20 m_filterButton . . . 64
8.5.3.21 m_filterIPLineEdit . . . 64
8.5.3.22 m_filterTypeComboBox . . . 64
8.5.3.23 m_filterVendorComboBox . . . 64
8.5.3.24 m_filterWidget . . . 65
8.5.3.25 m_helpMenu . . . 65
8.5.3.26 m_historyTab . . . 65
8.5.3.27 m_historyTable . . . 65
8.5.3.28 m_hostsFound . . . 65
8.5.3.29 m_loadHistoryAction . . . 65
8.5.3.30 m_mainLayout . . . 65
8.5.3.31 m_networkTopology . . . 65
8.5.3.32 m_portDistributionChartView . . . 65
8.5.3.33 m_portsGroupBox . . . 65
8.5.3.34 m_portsLineEdit . . . 66
8.5.3.35 m_progressBar . . . 66
8.5.3.36 m_rangeGroupBox . . . 66

8.5.3.37 m_resultsTable . . . 66
8.5.3.38 m_saveButton . . . 66
8.5.3.39 m_saveHistoryAction . . . 66
8.5.3.40 m_saveTopologyAction . . . 66
8.5.3.41 m_scanButton . . . 66
8.5.3.42 m_scanHistory . . . 66
8.5.3.43 m_scanner . . . 66
8.5.3.44 m_scanTab . . . 67
8.5.3.45 m_scheduleScanAction . . . 67
8.5.3.46 m_securityReportText . . . 67
8.5.3.47 m_sessionComboBox . . . 67
8.5.3.48 m_settingsAction . . . 67
8.5.3.49 m_settingsLayout . . . 67
8.5.3.50 m_settingsTab . . . 67
8.5.3.51 m_startIPLineEdit . . . 67
8.5.3.52 m_statisticsTab . . . 67
8.5.3.53 m_statusBar . . . 67
8.5.3.54 m_statusLabel . . . 68
8.5.3.55 m_stopButton . . . 68
8.5.3.56 m_tabWidget . . . 68
8.5.3.57 m_timeoutSpinBox . . . 68
8.5.3.58 m_toolsMenu . . . 68
8.5.3.59 m_topologyTab . . . 68
8.5.3.60 m_vendorChartView . . . 68
8.5.3.61 m_viewMenu . . . 68
8.6 NetworkScanner类 参考 . . . 69
8.6.1 详细描述 . . . 73
8.6.2 构造及析构函数说明 . . . 73
8.6.2.1 NetworkScanner() . . . 73
8.6.2.2 ~NetworkScanner() . . . 73
8.6.3 成员函数说明 . . . 74
8.6.3.1 executeProcess() . . . 74
8.6.3.2 generatePseudoMACFromIP() . . . 74
8.6.3.3 getAddressesToScan() . . . 75
8.6.3.4 getLocalNetworkAddresses() . . . 75
8.6.3.5 getMacAddressFromSystemCalls() . . . 75
8.6.3.6 getScannedHosts() . . . 76
8.6.3.7 hostFound . . . 76
8.6.3.8 isHostReachable() . . . 76
8.6.3.9 isReachableOnPorts() . . . 77
8.6.3.10 isScanning() . . . 77
8.6.3.11 lookupHostName() . . . 77

8.6.3.12 lookupMacAddress() . . . 78
8.6.3.13 lookupMacVendor() . . . 78
8.6.3.14 normalizeMacAddress() . . . 79
8.6.3.15 onHostNameLookedUp . . . 79
8.6.3.16 onScanTaskFinished . . . 79
8.6.3.17 performARPScan() . . . 80
8.6.3.18 pingTargetWithTimeout() . . . 80
8.6.3.19 processScanResults() . . . 81
8.6.3.20 quickPingScan() . . . 81
8.6.3.21 saveResultsToFile() . . . 82
8.6.3.22 scanError . . . 82
8.6.3.23 scanFinished . . . 83
8.6.3.24 scanHost() . . . 83
8.6.3.25 scanHostPorts() . . . 84
8.6.3.26 scanProgress . . . 84
8.6.3.27 scanStarted . . . 85
8.6.3.28 setCustomIPRange() . . . 85
8.6.3.29 setCustomPortsToScan() . . . 85
8.6.3.30 setScanStrategy() . . . 86
8.6.3.31 setScanTimeout() . . . 86
8.6.3.32 startScan() . . . 86
8.6.3.33 stopScan() . . . 87
8.6.3.34 updateScanProgress . . . 87
8.6.4 类成员变量说明 . . . 88
8.6.4.1 m_activeHosts . . . 88
8.6.4.2 m_endIPRange . . . 88
8.6.4.3 m_isScanning . . . 88
8.6.4.4 m_macAddressCache . . . 88
8.6.4.5 m_mutex . . . 88
8.6.4.6 m_portsToScan . . . 88
8.6.4.7 m_scanFutures . . . 88
8.6.4.8 m_scannedHosts . . . 89
8.6.4.9 m_scannedHostsList . . . 89
8.6.4.10 m_scanStrategy . . . 89
8.6.4.11 m_scanTimeout . . . 89
8.6.4.12 m_startIPRange . . . 89
8.6.4.13 m_threadPool . . . 89
8.6.4.14 m_totalHosts . . . 89
8.6.4.15 m_useCustomRange . . . 90
8.7 NetworkTopology类 参考 . . . 90
8.7.1 成员枚举类型说明 . . . 92
8.7.1.1 LayoutMode . . . 92

8.7.2 构造及析构函数说明 . . . 92
8.7.2.1 NetworkTopology() . . . 92
8.7.3 成员函数说明 . . . 93
8.7.3.1 clear() . . . 93
8.7.3.2 deviceSelected . . . 93
8.7.3.3 resetView() . . . 93
8.7.3.4 scale() . . . 93
8.7.3.5 setLayoutMode() . . . 93
8.7.3.6 updateTopology() . . . 94
8.7.4 类成员变量说明 . . . 94
8.7.4.1 m_controlPanel . . . 94
8.7.4.2 m_currentHosts . . . 94
8.7.4.3 m_layoutMode . . . 94
8.7.4.4 m_topologyView . . . 94
8.8 NetworkTopologyView类 参考 . . . 95
8.8.1 构造及析构函数说明 . . . 97
8.8.1.1 NetworkTopologyView() . . . 97
8.8.2 成员函数说明 . . . 97
8.8.2.1 autoLayout() . . . 97
8.8.2.2 clear() . . . 98
8.8.2.3 createConnection() . . . 98
8.8.2.4 determineDeviceType() . . . 98
8.8.2.5 groupedLayout() . . . 99
8.8.2.6 hierarchicalLayout() . . . 99
8.8.2.7 nodeSelected . . . 99
8.8.2.8 setHosts() . . . 99
8.8.3 类成员变量说明 . . . 100
8.8.3.1 m_analyzer . . . 100
8.8.3.2 m_connections . . . 100
8.8.3.3 m_nodes . . . 100
8.8.3.4 m_scene . . . 101
8.9 QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN10MainWindowE_t结构体 参考 . . . 101
8.10 QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN11ScanHistoryE_t结构体 参考 . . . 101
8.11 QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN14DeviceAnalyzerE_t结构体 参考 . . . 102
8.12 QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN14NetworkScannerE_t结构体 参考 . . . 102
8.13 QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN19NetworkTopologyViewE_t结构体 参考 . . . 103
8.14 ScanHistory类 参考 . . . 103
8.14.1 构造及析构函数说明 . . . 106
8.14.1.1 ScanHistory() . . . 106
8.14.2 成员函数说明 . . . 106
8.14.2.1 addSession() . . . 106
8.14.2.2 clearHistory() . . . 106

8.14.2.3 compareScans() . . . 107
8.14.2.4 getSession() . . . 107
8.14.2.5 getSessions() . . . 107
8.14.2.6 historyChanged . . . 107
8.14.2.7 loadFromFile() . . . 107
8.14.2.8 removeSession() . . . 108
8.14.2.9 saveToFile() . . . 108
8.14.2.10 sessionCount() . . . 108
8.14.3 类成员变量说明 . . . 108
8.14.3.1 m_sessions . . . 108
8.15 ScanSession结构体 参考 . . . 109
8.15.1 成员函数说明 . . . 109
8.15.1.1 portDistribution() . . . 109
8.15.1.2 reachableHosts() . . . 110
8.15.1.3 totalHosts() . . . 110
8.15.1.4 unreachableHosts() . . . 110
8.15.2 类成员变量说明 . . . 110
8.15.2.1 description . . . 110
8.15.2.2 hosts . . . 111
8.15.2.3 scanTime . . . 111
8.16 ScanStrategy类 参考 . . . 111
8.16.1 详细描述 . . . 112
8.16.2 成员枚举类型说明 . . . 112
8.16.2.1 ScanMode . . . 112
8.16.3 构造及析构函数说明 . . . 113
8.16.3.1 ScanStrategy() . . . 113
8.16.4 成员函数说明 . . . 113
8.16.4.1 getMaxParallelTasks() . . . 113
8.16.4.2 getMode() . . . 113
8.16.4.3 getPortsToScan() . . . 113
8.16.4.4 getScanTimeout() . . . 113
8.16.4.5 setMode() . . . 114
8.16.4.6 updateHostResponseTime() . . . 114
8.16.5 类成员变量说明 . . . 114
8.16.5.1 m_hostResponseTimes . . . 114
8.16.5.2 m_mode . . . 114
8.17 ScanTask类 参考 . . . 115
8.17.1 详细描述 . . . 116
8.17.2 构造及析构函数说明 . . . 117
8.17.2.1 ScanTask() . . . 117
8.17.3 成员函数说明 . . . 117
8.17.3.1 run() . . . 117

8.17.4 类成员变量说明 . . . 117
8.17.4.1 m_address . . . 117
8.17.4.2 m_parent . . . 118
8.17.4.3 m_ports . . . 118
8.17.4.4 m_timeout . . . 118
8.18 TopologyAnalyzer类 参考 . . . 118
8.18.1 构造及析构函数说明 . . . 119
8.18.1.1 TopologyAnalyzer() . . . 119
8.18.2 成员函数说明 . . . 120
8.18.2.1 analyzeSubnets() . . . 120
8.18.2.2 analyzeTTLLayers() . . . 120
8.18.2.3 calculateSubnet() . . . 121
8.18.2.4 clusterDevicesByResponseTime() . . . 121
8.18.2.5 getTTLValue() . . . 121
8.18.2.6 inferDeviceConnections() . . . 122
8.18.2.7 inSameSubnet() . . . 122
8.18.2.8 performTraceRoute() . . . 122

9 文件说明 **123**
9.1 build/CMakeFiles/3.31.5/CompilerIdCXX/CMakeCXXCompilerId.cpp 文件参考 . . . 123
9.1.1 宏定义说明 . . . 124
9.1.1.1 __has_include . . . 124
9.1.1.2 ARCHITECTURE_ID . . . 124
9.1.1.3 COMPILER_ID . . . 124
9.1.1.4 CXX_STD . . . 124
9.1.1.5 CXX_STD_11 . . . 124
9.1.1.6 CXX_STD_14 . . . 124
9.1.1.7 CXX_STD_17 . . . 124
9.1.1.8 CXX_STD_20 . . . 124
9.1.1.9 CXX_STD_23 . . . 124
9.1.1.10 CXX_STD_98 . . . 125
9.1.1.11 DEC . . . 125
9.1.1.12 HEX . . . 125
9.1.1.13 PLATFORM_ID . . . 125
9.1.1.14 STRINGIFY . . . 125
9.1.1.15 STRINGIFY_HELPER . . . 125
9.1.2 函数说明 . . . 126
9.1.2.1 main() . . . 126
9.1.3 变量说明 . . . 126
9.1.3.1 info_arch . . . 126
9.1.3.2 info_compiler . . . 126
9.1.3.3 info_language_extensions_default . . . 126

9.1.3.4 info_language_standard_default . . . 126
9.1.3.5 info_platform . . . 127
9.2 build/CMakeFiles/NetScanner.dir/deviceanalyzer.cpp.o.d 文件参考 . . . 127
9.3 build/CMakeFiles/NetScanner.dir/main.cpp.o.d 文件参考 . . . 127
9.4 build/CMakeFiles/NetScanner.dir/mainwindow.cpp.o.d 文件参考 . . . 127
9.5 build/CMakeFiles/NetScanner.dir/NetScanner_autogen/mocs_compilation.cpp.o.d 文件参考 . . . 127
9.6 build/CMakeFiles/NetScanner.dir/networkscanner.cpp.o.d 文件参考 . . . 127
9.7 build/CMakeFiles/NetScanner.dir/networktopology.cpp.o.d 文件参考 . . . 127
9.8 build/CMakeFiles/NetScanner.dir/scanhistory.cpp.o.d 文件参考 . . . 127
9.9 build/NetScanner_autogen/EWIEGA46WW/moc_deviceanalyzer.cpp 文件参考 . . . 127
9.9.1 宏定义说明 . . . 128
9.9.1.1 Q_CONSTINIT . . . 128
9.10 moc_deviceanalyzer.cpp . . . 128
9.11 NetScanner_autogen/EWIEGA46WW/moc_deviceanalyzer.cpp 文件参考 . . . 130
9.11.1 宏定义说明 . . . 131
9.11.1.1 Q_CONSTINIT . . . 131
9.12 moc_deviceanalyzer.cpp . . . 131
9.13 NetScanner_autogen/JRIAJ772TK/moc_deviceanalyzer.cpp 文件参考 . . . 132
9.13.1 宏定义说明 . . . 133
9.13.1.1 Q_CONSTINIT . . . 133
9.14 moc_deviceanalyzer.cpp . . . 134
9.15 build/NetScanner_autogen/EWIEGA46WW/moc_deviceanalyzer.cpp.d 文件参考 . . . 135
9.16 NetScanner_autogen/EWIEGA46WW/moc_deviceanalyzer.cpp.d 文件参考 . . . 135
9.17 NetScanner_autogen/JRIAJ772TK/moc_deviceanalyzer.cpp.d 文件参考 . . . 135
9.18 build/NetScanner_autogen/EWIEGA46WW/moc_mainwindow.cpp 文件参考 . . . 135
9.18.1 宏定义说明 . . . 136
9.18.1.1 Q_CONSTINIT . . . 136
9.19 moc_mainwindow.cpp . . . 136
9.20 NetScanner_autogen/EWIEGA46WW/moc_mainwindow.cpp 文件参考 . . . 139
9.20.1 宏定义说明 . . . 140
9.20.1.1 Q_CONSTINIT . . . 140
9.21 moc_mainwindow.cpp . . . 141
9.22 NetScanner_autogen/JRIAJ772TK/moc_mainwindow.cpp 文件参考 . . . 144
9.22.1 宏定义说明 . . . 144
9.22.1.1 Q_CONSTINIT . . . 144
9.23 moc_mainwindow.cpp . . . 145
9.24 build/NetScanner_autogen/EWIEGA46WW/moc_mainwindow.cpp.d 文件参考 . . . 148
9.25 NetScanner_autogen/EWIEGA46WW/moc_mainwindow.cpp.d 文件参考 . . . 148
9.26 NetScanner_autogen/JRIAJ772TK/moc_mainwindow.cpp.d 文件参考 . . . 148
9.27 build/NetScanner_autogen/EWIEGA46WW/moc_networkscanner.cpp 文件参考 . . . 148
9.27.1 宏定义说明 . . . 149
9.27.1.1 Q_CONSTINIT . . . 149

9.28 moc_networkscanner.cpp . . . 149
9.29 NetScanner_autogen/JRIAJ772TK/moc_networkscanner.cpp 文件参考 . . . 152
9.29.1 宏定义说明 . . . 152
9.29.1.1 Q_CONSTINIT . . . 152
9.30 moc_networkscanner.cpp . . . 153
9.31 build/NetScanner_autogen/EWIEGA46WW/moc_networkscanner.cpp.d 文件参考 . . . 155
9.32 NetScanner_autogen/JRIAJ772TK/moc_networkscanner.cpp.d 文件参考 . . . 155
9.33 build/NetScanner_autogen/EWIEGA46WW/moc_networktopology.cpp 文件参考 . . . 155
9.33.1 宏定义说明 . . . 156
9.33.1.1 Q_CONSTINIT . . . 156
9.34 moc_networktopology.cpp . . . 156
9.35 NetScanner_autogen/JRIAJ772TK/moc_networktopology.cpp 文件参考 . . . 159
9.35.1 宏定义说明 . . . 160
9.35.1.1 Q_CONSTINIT . . . 160
9.36 moc_networktopology.cpp . . . 160
9.37 build/NetScanner_autogen/EWIEGA46WW/moc_networktopology.cpp.d 文件参考 . . . 163
9.38 NetScanner_autogen/JRIAJ772TK/moc_networktopology.cpp.d 文件参考 . . . 163
9.39 build/NetScanner_autogen/EWIEGA46WW/moc_scanhistory.cpp 文件参考 . . . 163
9.39.1 宏定义说明 . . . 164
9.39.1.1 Q_CONSTINIT . . . 164
9.40 moc_scanhistory.cpp . . . 164
9.41 NetScanner_autogen/JRIAJ772TK/moc_scanhistory.cpp 文件参考 . . . 165
9.41.1 宏定义说明 . . . 166
9.41.1.1 Q_CONSTINIT . . . 166
9.42 moc_scanhistory.cpp . . . 167
9.43 build/NetScanner_autogen/EWIEGA46WW/moc_scanhistory.cpp.d 文件参考 . . . 168
9.44 NetScanner_autogen/JRIAJ772TK/moc_scanhistory.cpp.d 文件参考 . . . 168
9.45 build/NetScanner_autogen/moc_predefs.h 文件参考 . . . 168
9.45.1 宏定义说明 . . . 178
9.45.1.1 __aarch64__ . . . 178
9.45.1.2 __AARCH64_CMODEL_SMALL__ . . . 178
9.45.1.3 __AARCH64_SIMD__ . . . 178
9.45.1.4 __AARCH64EL__ . . . 178
9.45.1.5 __APPLE__ . . . 179
9.45.1.6 __apple_build_version__ . . . 179
9.45.1.7 __APPLE_CC__ . . . 179
9.45.1.8 __arm64 . . . 179
9.45.1.9 __arm64__ . . . 179
9.45.1.10 __ARM64_ARCH_8__ . . . 179
9.45.1.11 __ARM_64BIT_STATE . . . 179
9.45.1.12 __ARM_ACLE . . . 179
9.45.1.13 __ARM_ALIGN_MAX_STACK_PWR . . . 179

9.45.1.14 __ARM_ARCH . . . 179
9.45.1.15 __ARM_ARCH_8_3__ . . . 180
9.45.1.16 __ARM_ARCH_8_4__ . . . 180
9.45.1.17 __ARM_ARCH_8_5__ . . . 180
9.45.1.18 __ARM_ARCH_ISA_A64 . . . 180
9.45.1.19 __ARM_ARCH_PROFILE . . . 180
9.45.1.20 __ARM_FEATURE_AES . . . 180
9.45.1.21 __ARM_FEATURE_ATOMICS . . . 180
9.45.1.22 __ARM_FEATURE_BTI . . . 180
9.45.1.23 __ARM_FEATURE_CLZ . . . 180
9.45.1.24 __ARM_FEATURE_COMPLEX . . . 180
9.45.1.25 __ARM_FEATURE_CRC32 . . . 181
9.45.1.26 __ARM_FEATURE_CRYPTO . . . 181
9.45.1.27 __ARM_FEATURE_DIRECTED_ROUNDING . . . 181
9.45.1.28 __ARM_FEATURE_DIV . . . 181
9.45.1.29 __ARM_FEATURE_DOTPROD . . . 181
9.45.1.30 __ARM_FEATURE_FMA . . . 181
9.45.1.31 __ARM_FEATURE_FP16_FML . . . 181
9.45.1.32 __ARM_FEATURE_FP16_SCALAR_ARITHMETIC . . . 181
9.45.1.33 __ARM_FEATURE_FP16_VECTOR_ARITHMETIC . . . 181
9.45.1.34 __ARM_FEATURE_FRINT . . . 181
9.45.1.35 __ARM_FEATURE_IDIV . . . 182
9.45.1.36 __ARM_FEATURE_JCVT . . . 182
9.45.1.37 __ARM_FEATURE_LDREX . . . 182
9.45.1.38 __ARM_FEATURE_NUMERIC_MAXMIN . . . 182
9.45.1.39 __ARM_FEATURE_PAUTH . . . 182
9.45.1.40 __ARM_FEATURE_QRDMX . . . 182
9.45.1.41 __ARM_FEATURE_RCPC . . . 182
9.45.1.42 __ARM_FEATURE_SHA2 . . . 182
9.45.1.43 __ARM_FEATURE_SHA3 . . . 182
9.45.1.44 __ARM_FEATURE_SHA512 . . . 182
9.45.1.45 __ARM_FEATURE_UNALIGNED . . . 183
9.45.1.46 __ARM_FP . . . 183
9.45.1.47 __ARM_FP16_ARGS . . . 183
9.45.1.48 __ARM_FP16_FORMAT_IEEE . . . 183
9.45.1.49 __ARM_NEON . . . 183
9.45.1.50 __ARM_NEON__ . . . 183
9.45.1.51 __ARM_NEON_FP . . . 183
9.45.1.52 __ARM_PCS_AAPCS64 . . . 183
9.45.1.53 __ARM_SIZEOF_MINIMAL_ENUM . . . 183
9.45.1.54 __ARM_SIZEOF_WCHAR_T . . . 183
9.45.1.55 __ARM_STATE_ZA . . . 184

9.45.1.56 __ARM_STATE_ZT0 . . . 184
9.45.1.57 __ATOMIC_ACQ_REL . . . 184
9.45.1.58 __ATOMIC_ACQUIRE . . . 184
9.45.1.59 __ATOMIC_CONSUME . . . 184
9.45.1.60 __ATOMIC_RELAXED . . . 184
9.45.1.61 __ATOMIC_RELEASE . . . 184
9.45.1.62 __ATOMIC_SEQ_CST . . . 184
9.45.1.63 __BIGGEST_ALIGNMENT__ . . . 184
9.45.1.64 __BITINT_MAXWIDTH__ . . . 184
9.45.1.65 __block . . . 185
9.45.1.66 __BLOCKS__ . . . 185
9.45.1.67 __BOOL_WIDTH__ . . . 185
9.45.1.68 __BYTE_ORDER__ . . . 185
9.45.1.69 __CHAR16_TYPE__ . . . 185
9.45.1.70 __CHAR32_TYPE__ . . . 185
9.45.1.71 __CHAR_BIT__ . . . 185
9.45.1.72 __clang__ . . . 185
9.45.1.73 __CLANG_ATOMIC_BOOL_LOCK_FREE . . . 185
9.45.1.74 __CLANG_ATOMIC_CHAR16_T_LOCK_FREE . . . 185
9.45.1.75 __CLANG_ATOMIC_CHAR32_T_LOCK_FREE . . . 186
9.45.1.76 __CLANG_ATOMIC_CHAR_LOCK_FREE . . . 186
9.45.1.77 __CLANG_ATOMIC_INT_LOCK_FREE . . . 186
9.45.1.78 __CLANG_ATOMIC_LLONG_LOCK_FREE . . . 186
9.45.1.79 __CLANG_ATOMIC_LONG_LOCK_FREE . . . 186
9.45.1.80 __CLANG_ATOMIC_POINTER_LOCK_FREE . . . 186
9.45.1.81 __CLANG_ATOMIC_SHORT_LOCK_FREE . . . 186
9.45.1.82 __CLANG_ATOMIC_WCHAR_T_LOCK_FREE . . . 186
9.45.1.83 __clang_literal_encoding__ . . . 186
9.45.1.84 __clang_major__ . . . 186
9.45.1.85 __clang_minor__ . . . 187
9.45.1.86 __clang_patchlevel__ . . . 187
9.45.1.87 __clang_version__ . . . 187
9.45.1.88 __clang_wide_literal_encoding__ . . . 187
9.45.1.89 __CONSTANT_CFSTRINGS__ . . . 187
9.45.1.90 __cplusplus . . . 187
9.45.1.91 __cpp_aggregate_bases . . . 187
9.45.1.92 __cpp_aggregate_nsdmi . . . 187
9.45.1.93 __cpp_alias_templates . . . 187
9.45.1.94 __cpp_aligned_new . . . 187
9.45.1.95 __cpp_attributes . . . 188
9.45.1.96 __cpp_binary_literals . . . 188
9.45.1.97 __cpp_capture_star_this . . . 188

9.45.1.98 __cpp_constexpr . . . 188
9.45.1.99 __cpp_constexpr_in_decltype . . . 188
9.45.1.100 __cpp_decltype . . . 188
9.45.1.101 __cpp_decltype_auto . . . 188
9.45.1.102 __cpp_deduction_guides . . . 188
9.45.1.103 __cpp_delegating_constructors . . . 188
9.45.1.104 __cpp_deleted_function . . . 188
9.45.1.105 __cpp_digit_separators . . . 189
9.45.1.106 __cpp_enumerator_attributes . . . 189
9.45.1.107 __cpp_exceptions . . . 189
9.45.1.108 __cpp_fold_expressions . . . 189
9.45.1.109 __cpp_generic_lambdas . . . 189
9.45.1.110 __cpp_guaranteed_copy_elision . . . 189
9.45.1.111 __cpp_hex_float . . . 189
9.45.1.112 __cpp_if_constexpr . . . 189
9.45.1.113 __cpp_impl_destroying_delete . . . 189
9.45.1.114 __cpp_inheriting_constructors . . . 189
9.45.1.115 __cpp_init_captures . . . 190
9.45.1.116 __cpp_initializer_lists . . . 190
9.45.1.117 __cpp_inline_variables . . . 190
9.45.1.118 __cpp_lambdas . . . 190
9.45.1.119 __cpp_named_character_escapes . . . 190
9.45.1.120 __cpp_namespace_attributes . . . 190
9.45.1.121 __cpp_nested_namespace_definitions . . . 190
9.45.1.122 __cpp_noexcept_function_type . . . 190
9.45.1.123 __cpp_nontype_template_args . . . 190
9.45.1.124 __cpp_nontype_template_parameter_auto . . . 190
9.45.1.125 __cpp_nsdmi . . . 191
9.45.1.126 __cpp_pack_indexing . . . 191
9.45.1.127 __cpp_placeholder_variables . . . 191
9.45.1.128 __cpp_range_based_for . . . 191
9.45.1.129 __cpp_raw_strings . . . 191
9.45.1.130 __cpp_ref_qualifiers . . . 191
9.45.1.131 __cpp_return_type_deduction . . . 191
9.45.1.132 __cpp_rtti . . . 191
9.45.1.133 __cpp_rvalue_references . . . 191
9.45.1.134 __cpp_static_assert . . . 191
9.45.1.135 __cpp_static_call_operator . . . 192
9.45.1.136 __cpp_structured_bindings . . . 192
9.45.1.137 __cpp_template_auto . . . 192
9.45.1.138 __cpp_template_template_args . . . 192
9.45.1.139 __cpp_threadsafe_static_init . . . 192

9.45.1.140 __cpp_unicode_characters . . . 192
9.45.1.141 __cpp_unicode_literals . . . 192
9.45.1.142 __cpp_user_defined_literals . . . 192
9.45.1.143 __cpp_variable_templates . . . 192
9.45.1.144 __cpp_variadic_templates . . . 192
9.45.1.145 __cpp_variadic_using . . . 193
9.45.1.146 __DBL_DECIMAL_DIG__ . . . 193
9.45.1.147 __DBL_DENORM_MIN__ . . . 193
9.45.1.148 __DBL_DIG__ . . . 193
9.45.1.149 __DBL_EPSILON__ . . . 193
9.45.1.150 __DBL_HAS_DENORM__ . . . 193
9.45.1.151 __DBL_HAS_INFINITY__ . . . 193
9.45.1.152 __DBL_HAS_QUIET_NAN__ . . . 193
9.45.1.153 __DBL_MANT_DIG__ . . . 193
9.45.1.154 __DBL_MAX_10_EXP__ . . . 193
9.45.1.155 __DBL_MAX__ . . . 194
9.45.1.156 __DBL_MAX_EXP__ . . . 194
9.45.1.157 __DBL_MIN_10_EXP__ . . . 194
9.45.1.158 __DBL_MIN__ . . . 194
9.45.1.159 __DBL_MIN_EXP__ . . . 194
9.45.1.160 __DBL_NORM_MAX__ . . . 194
9.45.1.161 __DECIMAL_DIG__ . . . 194
9.45.1.162 __DEPRECATED . . . 194
9.45.1.163 __DYNAMIC__ . . . 194
9.45.1.164 __ENVIRONMENT_MAC_OS_X_VERSION_MIN_REQUIRED__ . . . 194
9.45.1.165 __ENVIRONMENT_OS_VERSION_MIN_REQUIRED__ . . . 195
9.45.1.166 __EXCEPTIONS . . . 195
9.45.1.167 __FINITE_MATH_ONLY__ . . . 195
9.45.1.168 __FLT16_DECIMAL_DIG__ . . . 195
9.45.1.169 __FLT16_DENORM_MIN__ . . . 195
9.45.1.170 __FLT16_DIG__ . . . 195
9.45.1.171 __FLT16_EPSILON__ . . . 195
9.45.1.172 __FLT16_HAS_DENORM__ . . . 195
9.45.1.173 __FLT16_HAS_INFINITY__ . . . 195
9.45.1.174 __FLT16_HAS_QUIET_NAN__ . . . 195
9.45.1.175 __FLT16_MANT_DIG__ . . . 196
9.45.1.176 __FLT16_MAX_10_EXP__ . . . 196
9.45.1.177 __FLT16_MAX__ . . . 196
9.45.1.178 __FLT16_MAX_EXP__ . . . 196
9.45.1.179 __FLT16_MIN_10_EXP__ . . . 196
9.45.1.180 __FLT16_MIN__ . . . 196
9.45.1.181 __FLT16_MIN_EXP__ . . . 196

9.45.1.182 __FLT16_NORM_MAX__ . . . 196
9.45.1.183 __FLT_DECIMAL_DIG__ . . . 196
9.45.1.184 __FLT_DENORM_MIN__ . . . 196
9.45.1.185 __FLT_DIG__ . . . 197
9.45.1.186 __FLT_EPSILON__ . . . 197
9.45.1.187 __FLT_HAS_DENORM__ . . . 197
9.45.1.188 __FLT_HAS_INFINITY__ . . . 197
9.45.1.189 __FLT_HAS_QUIET_NAN__ . . . 197
9.45.1.190 __FLT_MANT_DIG__ . . . 197
9.45.1.191 __FLT_MAX_10_EXP__ . . . 197
9.45.1.192 __FLT_MAX__ . . . 197
9.45.1.193 __FLT_MAX_EXP__ . . . 197
9.45.1.194 __FLT_MIN_10_EXP__ . . . 197
9.45.1.195 __FLT_MIN__ . . . 198
9.45.1.196 __FLT_MIN_EXP__ . . . 198
9.45.1.197 __FLT_NORM_MAX__ . . . 198
9.45.1.198 __FLT_RADIX__ . . . 198
9.45.1.199 __FP_FAST_FMA . . . 198
9.45.1.200 __FP_FAST_FMAF . . . 198
9.45.1.201 __FPCLASS_NEGINF . . . 198
9.45.1.202 __FPCLASS_NEGNORMAL . . . 198
9.45.1.203 __FPCLASS_NEGSUBNORMAL . . . 198
9.45.1.204 __FPCLASS_NEGZERO . . . 198
9.45.1.205 __FPCLASS_POSINF . . . 199
9.45.1.206 __FPCLASS_POSNORMAL . . . 199
9.45.1.207 __FPCLASS_POSSUBNORMAL . . . 199
9.45.1.208 __FPCLASS_POSZERO . . . 199
9.45.1.209 __FPCLASS_QNAN . . . 199
9.45.1.210 __FPCLASS_SNAN . . . 199
9.45.1.211 __GCC_ASM_FLAG_OUTPUTS__ . . . 199
9.45.1.212 __GCC_ATOMIC_BOOL_LOCK_FREE . . . 199
9.45.1.213 __GCC_ATOMIC_CHAR16_T_LOCK_FREE . . . 199
9.45.1.214 __GCC_ATOMIC_CHAR32_T_LOCK_FREE . . . 199
9.45.1.215 __GCC_ATOMIC_CHAR_LOCK_FREE . . . 200
9.45.1.216 __GCC_ATOMIC_INT_LOCK_FREE . . . 200
9.45.1.217 __GCC_ATOMIC_LLONG_LOCK_FREE . . . 200
9.45.1.218 __GCC_ATOMIC_LONG_LOCK_FREE . . . 200
9.45.1.219 __GCC_ATOMIC_POINTER_LOCK_FREE . . . 200
9.45.1.220 __GCC_ATOMIC_SHORT_LOCK_FREE . . . 200
9.45.1.221 __GCC_ATOMIC_TEST_AND_SET_TRUEVAL . . . 200
9.45.1.222 __GCC_ATOMIC_WCHAR_T_LOCK_FREE . . . 200
9.45.1.223 __GCC_CONSTRUCTIVE_SIZE . . . 200

9.45.1.224 __GCC_DESTRUCTIVE_SIZE . . . 200
9.45.1.225 __GCC_HAVE_DWARF2_CFI_ASM . . . 201
9.45.1.226 __GCC_HAVE_SYNC_COMPARE_AND_SWAP_1 . . . 201
9.45.1.227 __GCC_HAVE_SYNC_COMPARE_AND_SWAP_16 . . . 201
9.45.1.228 __GCC_HAVE_SYNC_COMPARE_AND_SWAP_2 . . . 201
9.45.1.229 __GCC_HAVE_SYNC_COMPARE_AND_SWAP_4 . . . 201
9.45.1.230 __GCC_HAVE_SYNC_COMPARE_AND_SWAP_8 . . . 201
9.45.1.231 __GLIBCXX_BITSIZE_INT_N_0 . . . 201
9.45.1.232 __GLIBCXX_TYPE_INT_N_0 . . . 201
9.45.1.233 __GNUC__ . . . 201
9.45.1.234 __GNUC_GNU_INLINE__ . . . 201
9.45.1.235 __GNUC_MINOR__ . . . 202
9.45.1.236 __GNUC_PATCHLEVEL__ . . . 202
9.45.1.237 __GNUG__ . . . 202
9.45.1.238 __GXX_ABI_VERSION . . . 202
9.45.1.239 __GXX_EXPERIMENTAL_CXX0X__ . . . 202
9.45.1.240 __GXX_RTTI . . . 202
9.45.1.241 __GXX_WEAK__ . . . 202
9.45.1.242 __HAVE_FUNCTION_MULTI_VERSIONING . . . 202
9.45.1.243 __INT16_C_SUFFIX__ . . . 202
9.45.1.244 __INT16_FMTd__ . . . 202
9.45.1.245 __INT16_FMTi__ . . . 203
9.45.1.246 __INT16_MAX__ . . . 203
9.45.1.247 __INT16_TYPE__ . . . 203
9.45.1.248 __INT32_C_SUFFIX__ . . . 203
9.45.1.249 __INT32_FMTd__ . . . 203
9.45.1.250 __INT32_FMTi__ . . . 203
9.45.1.251 __INT32_MAX__ . . . 203
9.45.1.252 __INT32_TYPE__ . . . 203
9.45.1.253 __INT64_C_SUFFIX__ . . . 203
9.45.1.254 __INT64_FMTd__ . . . 203
9.45.1.255 __INT64_FMTi__ . . . 204
9.45.1.256 __INT64_MAX__ . . . 204
9.45.1.257 __INT64_TYPE__ . . . 204
9.45.1.258 __INT8_C_SUFFIX__ . . . 204
9.45.1.259 __INT8_FMTd__ . . . 204
9.45.1.260 __INT8_FMTi__ . . . 204
9.45.1.261 __INT8_MAX__ . . . 204
9.45.1.262 __INT8_TYPE__ . . . 204
9.45.1.263 __INT_FAST16_FMTd__ . . . 204
9.45.1.264 __INT_FAST16_FMTi__ . . . 204
9.45.1.265 __INT_FAST16_MAX__ . . . 205

9.45.1.266 __INT_FAST16_TYPE__ . . . 205
9.45.1.267 __INT_FAST16_WIDTH__ . . . 205
9.45.1.268 __INT_FAST32_FMTd__ . . . 205
9.45.1.269 __INT_FAST32_FMTi__ . . . 205
9.45.1.270 __INT_FAST32_MAX__ . . . 205
9.45.1.271 __INT_FAST32_TYPE__ . . . 205
9.45.1.272 __INT_FAST32_WIDTH__ . . . 205
9.45.1.273 __INT_FAST64_FMTd__ . . . 205
9.45.1.274 __INT_FAST64_FMTi__ . . . 205
9.45.1.275 __INT_FAST64_MAX__ . . . 206
9.45.1.276 __INT_FAST64_TYPE__ . . . 206
9.45.1.277 __INT_FAST64_WIDTH__ . . . 206
9.45.1.278 __INT_FAST8_FMTd__ . . . 206
9.45.1.279 __INT_FAST8_FMTi__ . . . 206
9.45.1.280 __INT_FAST8_MAX__ . . . 206
9.45.1.281 __INT_FAST8_TYPE__ . . . 206
9.45.1.282 __INT_FAST8_WIDTH__ . . . 206
9.45.1.283 __INT_LEAST16_FMTd__ . . . 206
9.45.1.284 __INT_LEAST16_FMTi__ . . . 206
9.45.1.285 __INT_LEAST16_MAX__ . . . 207
9.45.1.286 __INT_LEAST16_TYPE__ . . . 207
9.45.1.287 __INT_LEAST16_WIDTH__ . . . 207
9.45.1.288 __INT_LEAST32_FMTd__ . . . 207
9.45.1.289 __INT_LEAST32_FMTi__ . . . 207
9.45.1.290 __INT_LEAST32_MAX__ . . . 207
9.45.1.291 __INT_LEAST32_TYPE__ . . . 207
9.45.1.292 __INT_LEAST32_WIDTH__ . . . 207
9.45.1.293 __INT_LEAST64_FMTd__ . . . 207
9.45.1.294 __INT_LEAST64_FMTi__ . . . 207
9.45.1.295 __INT_LEAST64_MAX__ . . . 208
9.45.1.296 __INT_LEAST64_TYPE__ . . . 208
9.45.1.297 __INT_LEAST64_WIDTH__ . . . 208
9.45.1.298 __INT_LEAST8_FMTd__ . . . 208
9.45.1.299 __INT_LEAST8_FMTi__ . . . 208
9.45.1.300 __INT_LEAST8_MAX__ . . . 208
9.45.1.301 __INT_LEAST8_TYPE__ . . . 208
9.45.1.302 __INT_LEAST8_WIDTH__ . . . 208
9.45.1.303 __INT_MAX__ . . . 208
9.45.1.304 __INT_WIDTH__ . . . 208
9.45.1.305 __INTMAX_C_SUFFIX__ . . . 209
9.45.1.306 __INTMAX_FMTd__ . . . 209
9.45.1.307 __INTMAX_FMTi__ . . . 209

9.45.1.308 __INTMAX_MAX__ . . . 209
9.45.1.309 __INTMAX_TYPE__ . . . 209
9.45.1.310 __INTMAX_WIDTH__ . . . 209
9.45.1.311 __INTPTR_FMTd__ . . . 209
9.45.1.312 __INTPTR_FMTi__ . . . 209
9.45.1.313 __INTPTR_MAX__ . . . 209
9.45.1.314 __INTPTR_TYPE__ . . . 209
9.45.1.315 __INTPTR_WIDTH__ . . . 210
9.45.1.316 __LDBL_DECIMAL_DIG__ . . . 210
9.45.1.317 __LDBL_DENORM_MIN__ . . . 210
9.45.1.318 __LDBL_DIG__ . . . 210
9.45.1.319 __LDBL_EPSILON__ . . . 210
9.45.1.320 __LDBL_HAS_DENORM__ . . . 210
9.45.1.321 __LDBL_HAS_INFINITY__ . . . 210
9.45.1.322 __LDBL_HAS_QUIET_NAN__ . . . 210
9.45.1.323 __LDBL_MANT_DIG__ . . . 210
9.45.1.324 __LDBL_MAX_10_EXP__ . . . 210
9.45.1.325 __LDBL_MAX__ . . . 211
9.45.1.326 __LDBL_MAX_EXP__ . . . 211
9.45.1.327 __LDBL_MIN_10_EXP__ . . . 211
9.45.1.328 __LDBL_MIN__ . . . 211
9.45.1.329 __LDBL_MIN_EXP__ . . . 211
9.45.1.330 __LDBL_NORM_MAX__ . . . 211
9.45.1.331 __LITTLE_ENDIAN__ . . . 211
9.45.1.332 __LLONG_WIDTH__ . . . 211
9.45.1.333 __llvm__ . . . 211
9.45.1.334 __LONG_LONG_MAX__ . . . 211
9.45.1.335 __LONG_MAX__ . . . 212
9.45.1.336 __LONG_WIDTH__ . . . 212
9.45.1.337 __LP64__ . . . 212
9.45.1.338 __MACH__ . . . 212
9.45.1.339 __MEMORY_SCOPE_DEVICE . . . 212
9.45.1.340 __MEMORY_SCOPE_SINGLE . . . 212
9.45.1.341 __MEMORY_SCOPE_SYSTEM . . . 212
9.45.1.342 __MEMORY_SCOPE_WRKGRP . . . 212
9.45.1.343 __MEMORY_SCOPE_WVFRNT . . . 212
9.45.1.344 __NO_INLINE__ . . . 212
9.45.1.345 __NO_MATH_ERRNO__ . . . 213
9.45.1.346 __nonnull . . . 213
9.45.1.347 __null_unspecified . . . 213
9.45.1.348 __nullable . . . 213
9.45.1.349 __OBJC_BOOL_IS_BOOL . . . 213

9.45.1.350 __OPENCL_MEMORY_SCOPE_ALL_SVM_DEVICES . . . 213
9.45.1.351 __OPENCL_MEMORY_SCOPE_DEVICE . . . 213
9.45.1.352 __OPENCL_MEMORY_SCOPE_SUB_GROUP . . . 213
9.45.1.353 __OPENCL_MEMORY_SCOPE_WORK_GROUP . . . 213
9.45.1.354 __OPENCL_MEMORY_SCOPE_WORK_ITEM . . . 213
9.45.1.355 __ORDER_BIG_ENDIAN__ . . . 214
9.45.1.356 __ORDER_LITTLE_ENDIAN__ . . . 214
9.45.1.357 __ORDER_PDP_ENDIAN__ . . . 214
9.45.1.358 __PIC__ . . . 214
9.45.1.359 __pic__ . . . 214
9.45.1.360 __POINTER_WIDTH__ . . . 214
9.45.1.361 __PRAGMA_REDEFINE_EXTNAME . . . 214
9.45.1.362 __private_extern__ . . . 214
9.45.1.363 __PTRDIFF_FMTd__ . . . 214
9.45.1.364 __PTRDIFF_FMTi__ . . . 214
9.45.1.365 __PTRDIFF_MAX__ . . . 215
9.45.1.366 __PTRDIFF_TYPE__ . . . 215
9.45.1.367 __PTRDIFF_WIDTH__ . . . 215
9.45.1.368 __REGISTER_PREFIX__ . . . 215
9.45.1.369 __SCHAR_MAX__ . . . 215
9.45.1.370 __SHRT_MAX__ . . . 215
9.45.1.371 __SHRT_WIDTH__ . . . 215
9.45.1.372 __SIG_ATOMIC_MAX__ . . . 215
9.45.1.373 __SIG_ATOMIC_WIDTH__ . . . 215
9.45.1.374 __SIZE_FMTo__ . . . 215
9.45.1.375 __SIZE_FMTu__ . . . 216
9.45.1.376 __SIZE_FMTX__ . . . 216
9.45.1.377 __SIZE_FMTx__ . . . 216
9.45.1.378 __SIZE_MAX__ . . . 216
9.45.1.379 __SIZE_TYPE__ . . . 216
9.45.1.380 __SIZE_WIDTH__ . . . 216
9.45.1.381 __SIZEOF_DOUBLE__ . . . 216
9.45.1.382 __SIZEOF_FLOAT__ . . . 216
9.45.1.383 __SIZEOF_INT128__ . . . 216
9.45.1.384 __SIZEOF_INT__ . . . 216
9.45.1.385 __SIZEOF_LONG__ . . . 217
9.45.1.386 __SIZEOF_LONG_DOUBLE__ . . . 217
9.45.1.387 __SIZEOF_LONG_LONG__ . . . 217
9.45.1.388 __SIZEOF_POINTER__ . . . 217
9.45.1.389 __SIZEOF_PTRDIFF_T__ . . . 217
9.45.1.390 __SIZEOF_SHORT__ . . . 217
9.45.1.391 __SIZEOF_SIZE_T__ . . . 217

9.45.1.392 __SIZEOF_WCHAR_T__ . . . 217
9.45.1.393 __SIZEOF_WINT_T__ . . . 217
9.45.1.394 __SSP__ . . . 217
9.45.1.395 __STDC__ . . . 218
9.45.1.396 __STDC_EMBED_EMPTY__ . . . 218
9.45.1.397 __STDC_EMBED_FOUND__ . . . 218
9.45.1.398 __STDC_EMBED_NOT_FOUND__ . . . 218
9.45.1.399 __STDC_HOSTED__ . . . 218
9.45.1.400 __STDC_NO_THREADS__ . . . 218
9.45.1.401 __STDC_UTF_16__ . . . 218
9.45.1.402 __STDC_UTF_32__ . . . 218
9.45.1.403 __STDCPP_DEFAULT_NEW_ALIGNMENT__ . . . 218
9.45.1.404 __STDCPP_THREADS__ . . . 218
9.45.1.405 __strong . . . 219
9.45.1.406 __UINT16_C_SUFFIX__ . . . 219
9.45.1.407 __UINT16_FMTo__ . . . 219
9.45.1.408 __UINT16_FMTu__ . . . 219
9.45.1.409 __UINT16_FMTX__ . . . 219
9.45.1.410 __UINT16_FMTx__ . . . 219
9.45.1.411 __UINT16_MAX__ . . . 219
9.45.1.412 __UINT16_TYPE__ . . . 219
9.45.1.413 __UINT32_C_SUFFIX__ . . . 219
9.45.1.414 __UINT32_FMTo__ . . . 219
9.45.1.415 __UINT32_FMTu__ . . . 220
9.45.1.416 __UINT32_FMTX__ . . . 220
9.45.1.417 __UINT32_FMTx__ . . . 220
9.45.1.418 __UINT32_MAX__ . . . 220
9.45.1.419 __UINT32_TYPE__ . . . 220
9.45.1.420 __UINT64_C_SUFFIX__ . . . 220
9.45.1.421 __UINT64_FMTo__ . . . 220
9.45.1.422 __UINT64_FMTu__ . . . 220
9.45.1.423 __UINT64_FMTX__ . . . 220
9.45.1.424 __UINT64_FMTx__ . . . 220
9.45.1.425 __UINT64_MAX__ . . . 221
9.45.1.426 __UINT64_TYPE__ . . . 221
9.45.1.427 __UINT8_C_SUFFIX__ . . . 221
9.45.1.428 __UINT8_FMTo__ . . . 221
9.45.1.429 __UINT8_FMTu__ . . . 221
9.45.1.430 __UINT8_FMTX__ . . . 221
9.45.1.431 __UINT8_FMTx__ . . . 221
9.45.1.432 __UINT8_MAX__ . . . 221
9.45.1.433 __UINT8_TYPE__ . . . 221

9.45.1.434 __UINT_FAST16_FMTo__ . . . 221
9.45.1.435 __UINT_FAST16_FMTu__ . . . 222
9.45.1.436 __UINT_FAST16_FMTX__ . . . 222
9.45.1.437 __UINT_FAST16_FMTx__ . . . 222
9.45.1.438 __UINT_FAST16_MAX__ . . . 222
9.45.1.439 __UINT_FAST16_TYPE__ . . . 222
9.45.1.440 __UINT_FAST32_FMTo__ . . . 222
9.45.1.441 __UINT_FAST32_FMTu__ . . . 222
9.45.1.442 __UINT_FAST32_FMTX__ . . . 222
9.45.1.443 __UINT_FAST32_FMTx__ . . . 222
9.45.1.444 __UINT_FAST32_MAX__ . . . 222
9.45.1.445 __UINT_FAST32_TYPE__ . . . 223
9.45.1.446 __UINT_FAST64_FMTo__ . . . 223
9.45.1.447 __UINT_FAST64_FMTu__ . . . 223
9.45.1.448 __UINT_FAST64_FMTX__ . . . 223
9.45.1.449 __UINT_FAST64_FMTx__ . . . 223
9.45.1.450 __UINT_FAST64_MAX__ . . . 223
9.45.1.451 __UINT_FAST64_TYPE__ . . . 223
9.45.1.452 __UINT_FAST8_FMTo__ . . . 223
9.45.1.453 __UINT_FAST8_FMTu__ . . . 223
9.45.1.454 __UINT_FAST8_FMTX__ . . . 223
9.45.1.455 __UINT_FAST8_FMTx__ . . . 224
9.45.1.456 __UINT_FAST8_MAX__ . . . 224
9.45.1.457 __UINT_FAST8_TYPE__ . . . 224
9.45.1.458 __UINT_LEAST16_FMTo__ . . . 224
9.45.1.459 __UINT_LEAST16_FMTu__ . . . 224
9.45.1.460 __UINT_LEAST16_FMTX__ . . . 224
9.45.1.461 __UINT_LEAST16_FMTx__ . . . 224
9.45.1.462 __UINT_LEAST16_MAX__ . . . 224
9.45.1.463 __UINT_LEAST16_TYPE__ . . . 224
9.45.1.464 __UINT_LEAST32_FMTo__ . . . 224
9.45.1.465 __UINT_LEAST32_FMTu__ . . . 225
9.45.1.466 __UINT_LEAST32_FMTX__ . . . 225
9.45.1.467 __UINT_LEAST32_FMTx__ . . . 225
9.45.1.468 __UINT_LEAST32_MAX__ . . . 225
9.45.1.469 __UINT_LEAST32_TYPE__ . . . 225
9.45.1.470 __UINT_LEAST64_FMTo__ . . . 225
9.45.1.471 __UINT_LEAST64_FMTu__ . . . 225
9.45.1.472 __UINT_LEAST64_FMTX__ . . . 225
9.45.1.473 __UINT_LEAST64_FMTx__ . . . 225
9.45.1.474 __UINT_LEAST64_MAX__ . . . 225
9.45.1.475 __UINT_LEAST64_TYPE__ . . . 226

9.45.1.476 __UINT_LEAST8_FMTo__ . . . 226
9.45.1.477 __UINT_LEAST8_FMTu__ . . . 226
9.45.1.478 __UINT_LEAST8_FMTX__ . . . 226
9.45.1.479 __UINT_LEAST8_FMTx__ . . . 226
9.45.1.480 __UINT_LEAST8_MAX__ . . . 226
9.45.1.481 __UINT_LEAST8_TYPE__ . . . 226
9.45.1.482 __UINTMAX_C_SUFFIX__ . . . 226
9.45.1.483 __UINTMAX_FMTo__ . . . 226
9.45.1.484 __UINTMAX_FMTu__ . . . 226
9.45.1.485 __UINTMAX_FMTX__ . . . 227
9.45.1.486 __UINTMAX_FMTx__ . . . 227
9.45.1.487 __UINTMAX_MAX__ . . . 227
9.45.1.488 __UINTMAX_TYPE__ . . . 227
9.45.1.489 __UINTMAX_WIDTH__ . . . 227
9.45.1.490 __UINTPTR_FMTo__ . . . 227
9.45.1.491 __UINTPTR_FMTu__ . . . 227
9.45.1.492 __UINTPTR_FMTX__ . . . 227
9.45.1.493 __UINTPTR_FMTx__ . . . 227
9.45.1.494 __UINTPTR_MAX__ . . . 227
9.45.1.495 __UINTPTR_TYPE__ . . . 228
9.45.1.496 __UINTPTR_WIDTH__ . . . 228
9.45.1.497 __unsafe_unretained . . . 228
9.45.1.498 __USER_LABEL_PREFIX__ . . . 228
9.45.1.499 __VERSION__ . . . 228
9.45.1.500 __WCHAR_MAX__ . . . 228
9.45.1.501 __WCHAR_TYPE__ . . . 228
9.45.1.502 __WCHAR_WIDTH__ . . . 228
9.45.1.503 __weak . . . 228
9.45.1.504 __WINT_MAX__ . . . 228
9.45.1.505 __WINT_TYPE__ . . . 229
9.45.1.506 __WINT_WIDTH__ . . . 229
9.45.1.507 _LP64 . . . 229
9.45.1.508 QT_CHARTS_LIB . . . 229
9.45.1.509 QT_CHARTS_USE_NAMESPACE . . . 229
9.45.1.510 QT_CONCURRENT_LIB . . . 229
9.45.1.511 QT_CORE_LIB . . . 229
9.45.1.512 QT_GUI_LIB . . . 229
9.45.1.513 QT_NETWORK_LIB . . . 229
9.45.1.514 QT_NO_DEBUG . . . 229
9.45.1.515 QT_OPENGL_LIB . . . 230
9.45.1.516 QT_OPENGLWIDGETS_LIB . . . 230
9.45.1.517 QT_WIDGETS_LIB . . . 230

9.45.1.518 SIZEOF_DPTR . . . 230
9.45.1.519 TARGET_IPHONE_SIMULATOR . . . 230
9.45.1.520 TARGET_OS_ARROW . . . 230
9.45.1.521 TARGET_OS_BRIDGE . . . 230
9.45.1.522 TARGET_OS_DRIVERKIT . . . 230
9.45.1.523 TARGET_OS_EMBEDDED . . . 230
9.45.1.524 TARGET_OS_IOS . . . 230
9.45.1.525 TARGET_OS_IOSMAC . . . 231
9.45.1.526 TARGET_OS_IPHONE . . . 231
9.45.1.527 TARGET_OS_LINUX . . . 231
9.45.1.528 TARGET_OS_MAC . . . 231
9.45.1.529 TARGET_OS_MACCATALYST . . . 231
9.45.1.530 TARGET_OS_NANO . . . 231
9.45.1.531 TARGET_OS_OSX . . . 231
9.45.1.532 TARGET_OS_SIMULATOR . . . 231
9.45.1.533 TARGET_OS_TV . . . 231
9.45.1.534 TARGET_OS_UIKITFORMAC . . . 231
9.45.1.535 TARGET_OS_UNIX . . . 232
9.45.1.536 TARGET_OS_VISION . . . 232
9.45.1.537 TARGET_OS_WATCH . . . 232
9.45.1.538 TARGET_OS_WIN32 . . . 232
9.45.1.539 TARGET_OS_WINDOWS . . . 232
9.45.1.540 TARGET_OS_XR . . . 232
9.46 moc_predefs.h . . . 232
9.47 NetScanner_autogen/moc_predefs.h 文件参考 . . . 238
9.47.1 宏定义说明 . . . 248
9.47.1.1 __aarch64__ . . . 248
9.47.1.2 __AARCH64_CMODEL_SMALL__ . . . 248
9.47.1.3 __AARCH64_SIMD__ . . . 249
9.47.1.4 __AARCH64EL__ . . . 249
9.47.1.5 __APPLE__ . . . 249
9.47.1.6 __apple_build_version__ . . . 249
9.47.1.7 __APPLE_CC__ . . . 249
9.47.1.8 __arm64 . . . 249
9.47.1.9 __arm64__ . . . 249
9.47.1.10 __ARM64_ARCH_8__ . . . 249
9.47.1.11 __ARM_64BIT_STATE . . . 249
9.47.1.12 __ARM_ACLE . . . 249
9.47.1.13 __ARM_ALIGN_MAX_STACK_PWR . . . 250
9.47.1.14 __ARM_ARCH . . . 250
9.47.1.15 __ARM_ARCH_8_3__ . . . 250
9.47.1.16 __ARM_ARCH_8_4__ . . . 250

9.47.1.17 __ARM_ARCH_8_5__ . . . 250
9.47.1.18 __ARM_ARCH_ISA_A64 . . . 250
9.47.1.19 __ARM_ARCH_PROFILE . . . 250
9.47.1.20 __ARM_FEATURE_AES . . . 250
9.47.1.21 __ARM_FEATURE_ATOMICS . . . 250
9.47.1.22 __ARM_FEATURE_BTI . . . 250
9.47.1.23 __ARM_FEATURE_CLZ . . . 251
9.47.1.24 __ARM_FEATURE_COMPLEX . . . 251
9.47.1.25 __ARM_FEATURE_CRC32 . . . 251
9.47.1.26 __ARM_FEATURE_CRYPTO . . . 251
9.47.1.27 __ARM_FEATURE_DIRECTED_ROUNDING . . . 251
9.47.1.28 __ARM_FEATURE_DIV . . . 251
9.47.1.29 __ARM_FEATURE_DOTPROD . . . 251
9.47.1.30 __ARM_FEATURE_FMA . . . 251
9.47.1.31 __ARM_FEATURE_FP16_FML . . . 251
9.47.1.32 __ARM_FEATURE_FP16_SCALAR_ARITHMETIC . . . 251
9.47.1.33 __ARM_FEATURE_FP16_VECTOR_ARITHMETIC . . . 252
9.47.1.34 __ARM_FEATURE_FRINT . . . 252
9.47.1.35 __ARM_FEATURE_IDIV . . . 252
9.47.1.36 __ARM_FEATURE_JCVT . . . 252
9.47.1.37 __ARM_FEATURE_LDREX . . . 252
9.47.1.38 __ARM_FEATURE_NUMERIC_MAXMIN . . . 252
9.47.1.39 __ARM_FEATURE_PAUTH . . . 252
9.47.1.40 __ARM_FEATURE_QRDMX . . . 252
9.47.1.41 __ARM_FEATURE_RCPC . . . 252
9.47.1.42 __ARM_FEATURE_SHA2 . . . 252
9.47.1.43 __ARM_FEATURE_SHA3 . . . 253
9.47.1.44 __ARM_FEATURE_SHA512 . . . 253
9.47.1.45 __ARM_FEATURE_UNALIGNED . . . 253
9.47.1.46 __ARM_FP . . . 253
9.47.1.47 __ARM_FP16_ARGS . . . 253
9.47.1.48 __ARM_FP16_FORMAT_IEEE . . . 253
9.47.1.49 __ARM_NEON . . . 253
9.47.1.50 __ARM_NEON__ . . . 253
9.47.1.51 __ARM_NEON_FP . . . 253
9.47.1.52 __ARM_PCS_AAPCS64 . . . 253
9.47.1.53 __ARM_SIZEOF_MINIMAL_ENUM . . . 254
9.47.1.54 __ARM_SIZEOF_WCHAR_T . . . 254
9.47.1.55 __ARM_STATE_ZA . . . 254
9.47.1.56 __ARM_STATE_ZT0 . . . 254
9.47.1.57 __ATOMIC_ACQ_REL . . . 254
9.47.1.58 __ATOMIC_ACQUIRE . . . 254

9.47.1.59 __ATOMIC_CONSUME . . . 254
9.47.1.60 __ATOMIC_RELAXED . . . 254
9.47.1.61 __ATOMIC_RELEASE . . . 254
9.47.1.62 __ATOMIC_SEQ_CST . . . 254
9.47.1.63 __BIGGEST_ALIGNMENT__ . . . 255
9.47.1.64 __BITINT_MAXWIDTH__ . . . 255
9.47.1.65 __block . . . 255
9.47.1.66 __BLOCKS__ . . . 255
9.47.1.67 __BOOL_WIDTH__ . . . 255
9.47.1.68 __BYTE_ORDER__ . . . 255
9.47.1.69 __CHAR16_TYPE__ . . . 255
9.47.1.70 __CHAR32_TYPE__ . . . 255
9.47.1.71 __CHAR_BIT__ . . . 255
9.47.1.72 __clang__ . . . 255
9.47.1.73 __CLANG_ATOMIC_BOOL_LOCK_FREE . . . 256
9.47.1.74 __CLANG_ATOMIC_CHAR16_T_LOCK_FREE . . . 256
9.47.1.75 __CLANG_ATOMIC_CHAR32_T_LOCK_FREE . . . 256
9.47.1.76 __CLANG_ATOMIC_CHAR_LOCK_FREE . . . 256
9.47.1.77 __CLANG_ATOMIC_INT_LOCK_FREE . . . 256
9.47.1.78 __CLANG_ATOMIC_LLONG_LOCK_FREE . . . 256
9.47.1.79 __CLANG_ATOMIC_LONG_LOCK_FREE . . . 256
9.47.1.80 __CLANG_ATOMIC_POINTER_LOCK_FREE . . . 256
9.47.1.81 __CLANG_ATOMIC_SHORT_LOCK_FREE . . . 256
9.47.1.82 __CLANG_ATOMIC_WCHAR_T_LOCK_FREE . . . 256
9.47.1.83 __clang_literal_encoding__ . . . 257
9.47.1.84 __clang_major__ . . . 257
9.47.1.85 __clang_minor__ . . . 257
9.47.1.86 __clang_patchlevel__ . . . 257
9.47.1.87 __clang_version__ . . . 257
9.47.1.88 __clang_wide_literal_encoding__ . . . 257
9.47.1.89 __CONSTANT_CFSTRINGS__ . . . 257
9.47.1.90 __cplusplus . . . 257
9.47.1.91 __cpp_aggregate_bases . . . 257
9.47.1.92 __cpp_aggregate_nsdmi . . . 257
9.47.1.93 __cpp_alias_templates . . . 258
9.47.1.94 __cpp_aligned_new . . . 258
9.47.1.95 __cpp_attributes . . . 258
9.47.1.96 __cpp_binary_literals . . . 258
9.47.1.97 __cpp_capture_star_this . . . 258
9.47.1.98 __cpp_constexpr . . . 258
9.47.1.99 __cpp_constexpr_in_decltype . . . 258
9.47.1.100 __cpp_decltype . . . 258

9.47.1.101 __cpp_decltype_auto . . . 258
9.47.1.102 __cpp_deduction_guides . . . 258
9.47.1.103 __cpp_delegating_constructors . . . 259
9.47.1.104 __cpp_deleted_function . . . 259
9.47.1.105 __cpp_digit_separators . . . 259
9.47.1.106 __cpp_enumerator_attributes . . . 259
9.47.1.107 __cpp_exceptions . . . 259
9.47.1.108 __cpp_fold_expressions . . . 259
9.47.1.109 __cpp_generic_lambdas . . . 259
9.47.1.110 __cpp_guaranteed_copy_elision . . . 259
9.47.1.111 __cpp_hex_float . . . 259
9.47.1.112 __cpp_if_constexpr . . . 259
9.47.1.113 __cpp_impl_destroying_delete . . . 260
9.47.1.114 __cpp_inheriting_constructors . . . 260
9.47.1.115 __cpp_init_captures . . . 260
9.47.1.116 __cpp_initializer_lists . . . 260
9.47.1.117 __cpp_inline_variables . . . 260
9.47.1.118 __cpp_lambdas . . . 260
9.47.1.119 __cpp_named_character_escapes . . . 260
9.47.1.120 __cpp_namespace_attributes . . . 260
9.47.1.121 __cpp_nested_namespace_definitions . . . 260
9.47.1.122 __cpp_noexcept_function_type . . . 260
9.47.1.123 __cpp_nontype_template_args . . . 261
9.47.1.124 __cpp_nontype_template_parameter_auto . . . 261
9.47.1.125 __cpp_nsdmi . . . 261
9.47.1.126 __cpp_pack_indexing . . . 261
9.47.1.127 __cpp_placeholder_variables . . . 261
9.47.1.128 __cpp_range_based_for . . . 261
9.47.1.129 __cpp_raw_strings . . . 261
9.47.1.130 __cpp_ref_qualifiers . . . 261
9.47.1.131 __cpp_return_type_deduction . . . 261
9.47.1.132 __cpp_rtti . . . 261
9.47.1.133 __cpp_rvalue_references . . . 262
9.47.1.134 __cpp_static_assert . . . 262
9.47.1.135 __cpp_static_call_operator . . . 262
9.47.1.136 __cpp_structured_bindings . . . 262
9.47.1.137 __cpp_template_auto . . . 262
9.47.1.138 __cpp_template_template_args . . . 262
9.47.1.139 __cpp_threadsafe_static_init . . . 262
9.47.1.140 __cpp_unicode_characters . . . 262
9.47.1.141 __cpp_unicode_literals . . . 262
9.47.1.142 __cpp_user_defined_literals . . . 262

9.47.1.143 __cpp_variable_templates . . . 263
9.47.1.144 __cpp_variadic_templates . . . 263
9.47.1.145 __cpp_variadic_using . . . 263
9.47.1.146 __DBL_DECIMAL_DIG__ . . . 263
9.47.1.147 __DBL_DENORM_MIN__ . . . 263
9.47.1.148 __DBL_DIG__ . . . 263
9.47.1.149 __DBL_EPSILON__ . . . 263
9.47.1.150 __DBL_HAS_DENORM__ . . . 263
9.47.1.151 __DBL_HAS_INFINITY__ . . . 263
9.47.1.152 __DBL_HAS_QUIET_NAN__ . . . 263
9.47.1.153 __DBL_MANT_DIG__ . . . 264
9.47.1.154 __DBL_MAX_10_EXP__ . . . 264
9.47.1.155 __DBL_MAX__ . . . 264
9.47.1.156 __DBL_MAX_EXP__ . . . 264
9.47.1.157 __DBL_MIN_10_EXP__ . . . 264
9.47.1.158 __DBL_MIN__ . . . 264
9.47.1.159 __DBL_MIN_EXP__ . . . 264
9.47.1.160 __DBL_NORM_MAX__ . . . 264
9.47.1.161 __DECIMAL_DIG__ . . . 264
9.47.1.162 __DEPRECATED . . . 264
9.47.1.163 __DYNAMIC__ . . . 265
9.47.1.164 __ENVIRONMENT_MAC_OS_X_VERSION_MIN_REQUIRED__ . . . 265
9.47.1.165 __ENVIRONMENT_OS_VERSION_MIN_REQUIRED__ . . . 265
9.47.1.166 __EXCEPTIONS . . . 265
9.47.1.167 __FINITE_MATH_ONLY__ . . . 265
9.47.1.168 __FLT16_DECIMAL_DIG__ . . . 265
9.47.1.169 __FLT16_DENORM_MIN__ . . . 265
9.47.1.170 __FLT16_DIG__ . . . 265
9.47.1.171 __FLT16_EPSILON__ . . . 265
9.47.1.172 __FLT16_HAS_DENORM__ . . . 265
9.47.1.173 __FLT16_HAS_INFINITY__ . . . 266
9.47.1.174 __FLT16_HAS_QUIET_NAN__ . . . 266
9.47.1.175 __FLT16_MANT_DIG__ . . . 266
9.47.1.176 __FLT16_MAX_10_EXP__ . . . 266
9.47.1.177 __FLT16_MAX__ . . . 266
9.47.1.178 __FLT16_MAX_EXP__ . . . 266
9.47.1.179 __FLT16_MIN_10_EXP__ . . . 266
9.47.1.180 __FLT16_MIN__ . . . 266
9.47.1.181 __FLT16_MIN_EXP__ . . . 266
9.47.1.182 __FLT16_NORM_MAX__ . . . 266
9.47.1.183 __FLT_DECIMAL_DIG__ . . . 267
9.47.1.184 __FLT_DENORM_MIN__ . . . 267

9.47.1.185 __FLT_DIG__ . . . 267
9.47.1.186 __FLT_EPSILON__ . . . 267
9.47.1.187 __FLT_HAS_DENORM__ . . . 267
9.47.1.188 __FLT_HAS_INFINITY__ . . . 267
9.47.1.189 __FLT_HAS_QUIET_NAN__ . . . 267
9.47.1.190 __FLT_MANT_DIG__ . . . 267
9.47.1.191 __FLT_MAX_10_EXP__ . . . 267
9.47.1.192 __FLT_MAX__ . . . 267
9.47.1.193 __FLT_MAX_EXP__ . . . 268
9.47.1.194 __FLT_MIN_10_EXP__ . . . 268
9.47.1.195 __FLT_MIN__ . . . 268
9.47.1.196 __FLT_MIN_EXP__ . . . 268
9.47.1.197 __FLT_NORM_MAX__ . . . 268
9.47.1.198 __FLT_RADIX__ . . . 268
9.47.1.199 __FP_FAST_FMA . . . 268
9.47.1.200 __FP_FAST_FMAF . . . 268
9.47.1.201 __FPCLASS_NEGINF . . . 268
9.47.1.202 __FPCLASS_NEGNORMAL . . . 268
9.47.1.203 __FPCLASS_NEGSUBNORMAL . . . 269
9.47.1.204 __FPCLASS_NEGZERO . . . 269
9.47.1.205 __FPCLASS_POSINF . . . 269
9.47.1.206 __FPCLASS_POSNORMAL . . . 269
9.47.1.207 __FPCLASS_POSSUBNORMAL . . . 269
9.47.1.208 __FPCLASS_POSZERO . . . 269
9.47.1.209 __FPCLASS_QNAN . . . 269
9.47.1.210 __FPCLASS_SNAN . . . 269
9.47.1.211 __GCC_ASM_FLAG_OUTPUTS__ . . . 269
9.47.1.212 __GCC_ATOMIC_BOOL_LOCK_FREE . . . 269
9.47.1.213 __GCC_ATOMIC_CHAR16_T_LOCK_FREE . . . 270
9.47.1.214 __GCC_ATOMIC_CHAR32_T_LOCK_FREE . . . 270
9.47.1.215 __GCC_ATOMIC_CHAR_LOCK_FREE . . . 270
9.47.1.216 __GCC_ATOMIC_INT_LOCK_FREE . . . 270
9.47.1.217 __GCC_ATOMIC_LLONG_LOCK_FREE . . . 270
9.47.1.218 __GCC_ATOMIC_LONG_LOCK_FREE . . . 270
9.47.1.219 __GCC_ATOMIC_POINTER_LOCK_FREE . . . 270
9.47.1.220 __GCC_ATOMIC_SHORT_LOCK_FREE . . . 270
9.47.1.221 __GCC_ATOMIC_TEST_AND_SET_TRUEVAL . . . 270
9.47.1.222 __GCC_ATOMIC_WCHAR_T_LOCK_FREE . . . 270
9.47.1.223 __GCC_CONSTRUCTIVE_SIZE . . . 271
9.47.1.224 __GCC_DESTRUCTIVE_SIZE . . . 271
9.47.1.225 __GCC_HAVE_DWARF2_CFI_ASM . . . 271
9.47.1.226 __GCC_HAVE_SYNC_COMPARE_AND_SWAP_1 . . . 271

9.47.1.227 __GCC_HAVE_SYNC_COMPARE_AND_SWAP_16 . . . 271
9.47.1.228 __GCC_HAVE_SYNC_COMPARE_AND_SWAP_2 . . . 271
9.47.1.229 __GCC_HAVE_SYNC_COMPARE_AND_SWAP_4 . . . 271
9.47.1.230 __GCC_HAVE_SYNC_COMPARE_AND_SWAP_8 . . . 271
9.47.1.231 __GLIBCXX_BITSIZE_INT_N_0 . . . 271
9.47.1.232 __GLIBCXX_TYPE_INT_N_0 . . . 271
9.47.1.233 __GNUC__ . . . 272
9.47.1.234 __GNUC_GNU_INLINE__ . . . 272
9.47.1.235 __GNUC_MINOR__ . . . 272
9.47.1.236 __GNUC_PATCHLEVEL__ . . . 272
9.47.1.237 __GNUG__ . . . 272
9.47.1.238 __GXX_ABI_VERSION . . . 272
9.47.1.239 __GXX_EXPERIMENTAL_CXX0X__ . . . 272
9.47.1.240 __GXX_RTTI . . . 272
9.47.1.241 __GXX_WEAK__ . . . 272
9.47.1.242 __HAVE_FUNCTION_MULTI_VERSIONING . . . 272
9.47.1.243 __INT16_C_SUFFIX__ . . . 273
9.47.1.244 __INT16_FMTd__ . . . 273
9.47.1.245 __INT16_FMTi__ . . . 273
9.47.1.246 __INT16_MAX__ . . . 273
9.47.1.247 __INT16_TYPE__ . . . 273
9.47.1.248 __INT32_C_SUFFIX__ . . . 273
9.47.1.249 __INT32_FMTd__ . . . 273
9.47.1.250 __INT32_FMTi__ . . . 273
9.47.1.251 __INT32_MAX__ . . . 273
9.47.1.252 __INT32_TYPE__ . . . 273
9.47.1.253 __INT64_C_SUFFIX__ . . . 274
9.47.1.254 __INT64_FMTd__ . . . 274
9.47.1.255 __INT64_FMTi__ . . . 274
9.47.1.256 __INT64_MAX__ . . . 274
9.47.1.257 __INT64_TYPE__ . . . 274
9.47.1.258 __INT8_C_SUFFIX__ . . . 274
9.47.1.259 __INT8_FMTd__ . . . 274
9.47.1.260 __INT8_FMTi__ . . . 274
9.47.1.261 __INT8_MAX__ . . . 274
9.47.1.262 __INT8_TYPE__ . . . 274
9.47.1.263 __INT_FAST16_FMTd__ . . . 275
9.47.1.264 __INT_FAST16_FMTi__ . . . 275
9.47.1.265 __INT_FAST16_MAX__ . . . 275
9.47.1.266 __INT_FAST16_TYPE__ . . . 275
9.47.1.267 __INT_FAST16_WIDTH__ . . . 275
9.47.1.268 __INT_FAST32_FMTd__ . . . 275

9.47.1.269 __INT_FAST32_FMTi__ . . . 275
9.47.1.270 __INT_FAST32_MAX__ . . . 275
9.47.1.271 __INT_FAST32_TYPE__ . . . 275
9.47.1.272 __INT_FAST32_WIDTH__ . . . 275
9.47.1.273 __INT_FAST64_FMTd__ . . . 276
9.47.1.274 __INT_FAST64_FMTi__ . . . 276
9.47.1.275 __INT_FAST64_MAX__ . . . 276
9.47.1.276 __INT_FAST64_TYPE__ . . . 276
9.47.1.277 __INT_FAST64_WIDTH__ . . . 276
9.47.1.278 __INT_FAST8_FMTd__ . . . 276
9.47.1.279 __INT_FAST8_FMTi__ . . . 276
9.47.1.280 __INT_FAST8_MAX__ . . . 276
9.47.1.281 __INT_FAST8_TYPE__ . . . 276
9.47.1.282 __INT_FAST8_WIDTH__ . . . 276
9.47.1.283 __INT_LEAST16_FMTd__ . . . 277
9.47.1.284 __INT_LEAST16_FMTi__ . . . 277
9.47.1.285 __INT_LEAST16_MAX__ . . . 277
9.47.1.286 __INT_LEAST16_TYPE__ . . . 277
9.47.1.287 __INT_LEAST16_WIDTH__ . . . 277
9.47.1.288 __INT_LEAST32_FMTd__ . . . 277
9.47.1.289 __INT_LEAST32_FMTi__ . . . 277
9.47.1.290 __INT_LEAST32_MAX__ . . . 277
9.47.1.291 __INT_LEAST32_TYPE__ . . . 277
9.47.1.292 __INT_LEAST32_WIDTH__ . . . 277
9.47.1.293 __INT_LEAST64_FMTd__ . . . 278
9.47.1.294 __INT_LEAST64_FMTi__ . . . 278
9.47.1.295 __INT_LEAST64_MAX__ . . . 278
9.47.1.296 __INT_LEAST64_TYPE__ . . . 278
9.47.1.297 __INT_LEAST64_WIDTH__ . . . 278
9.47.1.298 __INT_LEAST8_FMTd__ . . . 278
9.47.1.299 __INT_LEAST8_FMTi__ . . . 278
9.47.1.300 __INT_LEAST8_MAX__ . . . 278
9.47.1.301 __INT_LEAST8_TYPE__ . . . 278
9.47.1.302 __INT_LEAST8_WIDTH__ . . . 278
9.47.1.303 __INT_MAX__ . . . 279
9.47.1.304 __INT_WIDTH__ . . . 279
9.47.1.305 __INTMAX_C_SUFFIX__ . . . 279
9.47.1.306 __INTMAX_FMTd__ . . . 279
9.47.1.307 __INTMAX_FMTi__ . . . 279
9.47.1.308 __INTMAX_MAX__ . . . 279
9.47.1.309 __INTMAX_TYPE__ . . . 279
9.47.1.310 __INTMAX_WIDTH__ . . . 279

9.47.1.311 __INTPTR_FMTd__ . . . 279
9.47.1.312 __INTPTR_FMTi__ . . . 279
9.47.1.313 __INTPTR_MAX__ . . . 280
9.47.1.314 __INTPTR_TYPE__ . . . 280
9.47.1.315 __INTPTR_WIDTH__ . . . 280
9.47.1.316 __LDBL_DECIMAL_DIG__ . . . 280
9.47.1.317 __LDBL_DENORM_MIN__ . . . 280
9.47.1.318 __LDBL_DIG__ . . . 280
9.47.1.319 __LDBL_EPSILON__ . . . 280
9.47.1.320 __LDBL_HAS_DENORM__ . . . 280
9.47.1.321 __LDBL_HAS_INFINITY__ . . . 280
9.47.1.322 __LDBL_HAS_QUIET_NAN__ . . . 280
9.47.1.323 __LDBL_MANT_DIG__ . . . 281
9.47.1.324 __LDBL_MAX_10_EXP__ . . . 281
9.47.1.325 __LDBL_MAX__ . . . 281
9.47.1.326 __LDBL_MAX_EXP__ . . . 281
9.47.1.327 __LDBL_MIN_10_EXP__ . . . 281
9.47.1.328 __LDBL_MIN__ . . . 281
9.47.1.329 __LDBL_MIN_EXP__ . . . 281
9.47.1.330 __LDBL_NORM_MAX__ . . . 281
9.47.1.331 __LITTLE_ENDIAN__ . . . 281
9.47.1.332 __LLONG_WIDTH__ . . . 281
9.47.1.333 __llvm__ . . . 282
9.47.1.334 __LONG_LONG_MAX__ . . . 282
9.47.1.335 __LONG_MAX__ . . . 282
9.47.1.336 __LONG_WIDTH__ . . . 282
9.47.1.337 __LP64__ . . . 282
9.47.1.338 __MACH__ . . . 282
9.47.1.339 __MEMORY_SCOPE_DEVICE . . . 282
9.47.1.340 __MEMORY_SCOPE_SINGLE . . . 282
9.47.1.341 __MEMORY_SCOPE_SYSTEM . . . 282
9.47.1.342 __MEMORY_SCOPE_WRKGRP . . . 282
9.47.1.343 __MEMORY_SCOPE_WVFRNT . . . 283
9.47.1.344 __NO_INLINE__ . . . 283
9.47.1.345 __NO_MATH_ERRNO__ . . . 283
9.47.1.346 __nonnull . . . 283
9.47.1.347 __null_unspecified . . . 283
9.47.1.348 __nullable . . . 283
9.47.1.349 __OBJC_BOOL_IS_BOOL . . . 283
9.47.1.350 __OPENCL_MEMORY_SCOPE_ALL_SVM_DEVICES . . . 283
9.47.1.351 __OPENCL_MEMORY_SCOPE_DEVICE . . . 283
9.47.1.352 __OPENCL_MEMORY_SCOPE_SUB_GROUP . . . 283

9.47.1.353 __OPENCL_MEMORY_SCOPE_WORK_GROUP . . . 284
9.47.1.354 __OPENCL_MEMORY_SCOPE_WORK_ITEM . . . 284
9.47.1.355 __ORDER_BIG_ENDIAN__ . . . 284
9.47.1.356 __ORDER_LITTLE_ENDIAN__ . . . 284
9.47.1.357 __ORDER_PDP_ENDIAN__ . . . 284
9.47.1.358 __PIC__ . . . 284
9.47.1.359 __pic__ . . . 284
9.47.1.360 __POINTER_WIDTH__ . . . 284
9.47.1.361 __PRAGMA_REDEFINE_EXTNAME . . . 284
9.47.1.362 __private_extern__ . . . 284
9.47.1.363 __PTRDIFF_FMTd__ . . . 285
9.47.1.364 __PTRDIFF_FMTi__ . . . 285
9.47.1.365 __PTRDIFF_MAX__ . . . 285
9.47.1.366 __PTRDIFF_TYPE__ . . . 285
9.47.1.367 __PTRDIFF_WIDTH__ . . . 285
9.47.1.368 __REGISTER_PREFIX__ . . . 285
9.47.1.369 __SCHAR_MAX__ . . . 285
9.47.1.370 __SHRT_MAX__ . . . 285
9.47.1.371 __SHRT_WIDTH__ . . . 285
9.47.1.372 __SIG_ATOMIC_MAX__ . . . 285
9.47.1.373 __SIG_ATOMIC_WIDTH__ . . . 286
9.47.1.374 __SIZE_FMTo__ . . . 286
9.47.1.375 __SIZE_FMTu__ . . . 286
9.47.1.376 __SIZE_FMTX__ . . . 286
9.47.1.377 __SIZE_FMTx__ . . . 286
9.47.1.378 __SIZE_MAX__ . . . 286
9.47.1.379 __SIZE_TYPE__ . . . 286
9.47.1.380 __SIZE_WIDTH__ . . . 286
9.47.1.381 __SIZEOF_DOUBLE__ . . . 286
9.47.1.382 __SIZEOF_FLOAT__ . . . 286
9.47.1.383 __SIZEOF_INT128__ . . . 287
9.47.1.384 __SIZEOF_INT__ . . . 287
9.47.1.385 __SIZEOF_LONG__ . . . 287
9.47.1.386 __SIZEOF_LONG_DOUBLE__ . . . 287
9.47.1.387 __SIZEOF_LONG_LONG__ . . . 287
9.47.1.388 __SIZEOF_POINTER__ . . . 287
9.47.1.389 __SIZEOF_PTRDIFF_T__ . . . 287
9.47.1.390 __SIZEOF_SHORT__ . . . 287
9.47.1.391 __SIZEOF_SIZE_T__ . . . 287
9.47.1.392 __SIZEOF_WCHAR_T__ . . . 287
9.47.1.393 __SIZEOF_WINT_T__ . . . 288
9.47.1.394 __SSP__ . . . 288

9.47.1.395 __STDC__ . . . 288
9.47.1.396 __STDC_EMBED_EMPTY__ . . . 288
9.47.1.397 __STDC_EMBED_FOUND__ . . . 288
9.47.1.398 __STDC_EMBED_NOT_FOUND__ . . . 288
9.47.1.399 __STDC_HOSTED__ . . . 288
9.47.1.400 __STDC_NO_THREADS__ . . . 288
9.47.1.401 __STDC_UTF_16__ . . . 288
9.47.1.402 __STDC_UTF_32__ . . . 288
9.47.1.403 __STDCPP_DEFAULT_NEW_ALIGNMENT__ . . . 289
9.47.1.404 __STDCPP_THREADS__ . . . 289
9.47.1.405 __strong . . . 289
9.47.1.406 __UINT16_C_SUFFIX__ . . . 289
9.47.1.407 __UINT16_FMTo__ . . . 289
9.47.1.408 __UINT16_FMTu__ . . . 289
9.47.1.409 __UINT16_FMTX__ . . . 289
9.47.1.410 __UINT16_FMTx__ . . . 289
9.47.1.411 __UINT16_MAX__ . . . 289
9.47.1.412 __UINT16_TYPE__ . . . 289
9.47.1.413 __UINT32_C_SUFFIX__ . . . 290
9.47.1.414 __UINT32_FMTo__ . . . 290
9.47.1.415 __UINT32_FMTu__ . . . 290
9.47.1.416 __UINT32_FMTX__ . . . 290
9.47.1.417 __UINT32_FMTx__ . . . 290
9.47.1.418 __UINT32_MAX__ . . . 290
9.47.1.419 __UINT32_TYPE__ . . . 290
9.47.1.420 __UINT64_C_SUFFIX__ . . . 290
9.47.1.421 __UINT64_FMTo__ . . . 290
9.47.1.422 __UINT64_FMTu__ . . . 290
9.47.1.423 __UINT64_FMTX__ . . . 291
9.47.1.424 __UINT64_FMTx__ . . . 291
9.47.1.425 __UINT64_MAX__ . . . 291
9.47.1.426 __UINT64_TYPE__ . . . 291
9.47.1.427 __UINT8_C_SUFFIX__ . . . 291
9.47.1.428 __UINT8_FMTo__ . . . 291
9.47.1.429 __UINT8_FMTu__ . . . 291
9.47.1.430 __UINT8_FMTX__ . . . 291
9.47.1.431 __UINT8_FMTx__ . . . 291
9.47.1.432 __UINT8_MAX__ . . . 291
9.47.1.433 __UINT8_TYPE__ . . . 292
9.47.1.434 __UINT_FAST16_FMTo__ . . . 292
9.47.1.435 __UINT_FAST16_FMTu__ . . . 292
9.47.1.436 __UINT_FAST16_FMTX__ . . . 292

9.47.1.437 __UINT_FAST16_FMTx__ . . . 292
9.47.1.438 __UINT_FAST16_MAX__ . . . 292
9.47.1.439 __UINT_FAST16_TYPE__ . . . 292
9.47.1.440 __UINT_FAST32_FMTo__ . . . 292
9.47.1.441 __UINT_FAST32_FMTu__ . . . 292
9.47.1.442 __UINT_FAST32_FMTX__ . . . 292
9.47.1.443 __UINT_FAST32_FMTx__ . . . 293
9.47.1.444 __UINT_FAST32_MAX__ . . . 293
9.47.1.445 __UINT_FAST32_TYPE__ . . . 293
9.47.1.446 __UINT_FAST64_FMTo__ . . . 293
9.47.1.447 __UINT_FAST64_FMTu__ . . . 293
9.47.1.448 __UINT_FAST64_FMTX__ . . . 293
9.47.1.449 __UINT_FAST64_FMTx__ . . . 293
9.47.1.450 __UINT_FAST64_MAX__ . . . 293
9.47.1.451 __UINT_FAST64_TYPE__ . . . 293
9.47.1.452 __UINT_FAST8_FMTo__ . . . 293
9.47.1.453 __UINT_FAST8_FMTu__ . . . 294
9.47.1.454 __UINT_FAST8_FMTX__ . . . 294
9.47.1.455 __UINT_FAST8_FMTx__ . . . 294
9.47.1.456 __UINT_FAST8_MAX__ . . . 294
9.47.1.457 __UINT_FAST8_TYPE__ . . . 294
9.47.1.458 __UINT_LEAST16_FMTo__ . . . 294
9.47.1.459 __UINT_LEAST16_FMTu__ . . . 294
9.47.1.460 __UINT_LEAST16_FMTX__ . . . 294
9.47.1.461 __UINT_LEAST16_FMTx__ . . . 294
9.47.1.462 __UINT_LEAST16_MAX__ . . . 294
9.47.1.463 __UINT_LEAST16_TYPE__ . . . 295
9.47.1.464 __UINT_LEAST32_FMTo__ . . . 295
9.47.1.465 __UINT_LEAST32_FMTu__ . . . 295
9.47.1.466 __UINT_LEAST32_FMTX__ . . . 295
9.47.1.467 __UINT_LEAST32_FMTx__ . . . 295
9.47.1.468 __UINT_LEAST32_MAX__ . . . 295
9.47.1.469 __UINT_LEAST32_TYPE__ . . . 295
9.47.1.470 __UINT_LEAST64_FMTo__ . . . 295
9.47.1.471 __UINT_LEAST64_FMTu__ . . . 295
9.47.1.472 __UINT_LEAST64_FMTX__ . . . 295
9.47.1.473 __UINT_LEAST64_FMTx__ . . . 296
9.47.1.474 __UINT_LEAST64_MAX__ . . . 296
9.47.1.475 __UINT_LEAST64_TYPE__ . . . 296
9.47.1.476 __UINT_LEAST8_FMTo__ . . . 296
9.47.1.477 __UINT_LEAST8_FMTu__ . . . 296
9.47.1.478 __UINT_LEAST8_FMTX__ . . . 296

9.47.1.479 __UINT_LEAST8_FMTx__ . . . 296
9.47.1.480 __UINT_LEAST8_MAX__ . . . 296
9.47.1.481 __UINT_LEAST8_TYPE__ . . . 296
9.47.1.482 __UINTMAX_C_SUFFIX__ . . . 296
9.47.1.483 __UINTMAX_FMTo__ . . . 297
9.47.1.484 __UINTMAX_FMTu__ . . . 297
9.47.1.485 __UINTMAX_FMTX__ . . . 297
9.47.1.486 __UINTMAX_FMTx__ . . . 297
9.47.1.487 __UINTMAX_MAX__ . . . 297
9.47.1.488 __UINTMAX_TYPE__ . . . 297
9.47.1.489 __UINTMAX_WIDTH__ . . . 297
9.47.1.490 __UINTPTR_FMTo__ . . . 297
9.47.1.491 __UINTPTR_FMTu__ . . . 297
9.47.1.492 __UINTPTR_FMTX__ . . . 297
9.47.1.493 __UINTPTR_FMTx__ . . . 298
9.47.1.494 __UINTPTR_MAX__ . . . 298
9.47.1.495 __UINTPTR_TYPE__ . . . 298
9.47.1.496 __UINTPTR_WIDTH__ . . . 298
9.47.1.497 __unsafe_unretained . . . 298
9.47.1.498 __USER_LABEL_PREFIX__ . . . 298
9.47.1.499 __VERSION__ . . . 298
9.47.1.500 __WCHAR_MAX__ . . . 298
9.47.1.501 __WCHAR_TYPE__ . . . 298
9.47.1.502 __WCHAR_WIDTH__ . . . 298
9.47.1.503 __weak . . . 299
9.47.1.504 __WINT_MAX__ . . . 299
9.47.1.505 __WINT_TYPE__ . . . 299
9.47.1.506 __WINT_WIDTH__ . . . 299
9.47.1.507 _LP64 . . . 299
9.47.1.508 QT_CHARTS_LIB . . . 299
9.47.1.509 QT_CORE_LIB . . . 299
9.47.1.510 QT_GUI_LIB . . . 299
9.47.1.511 QT_NETWORK_LIB . . . 299
9.47.1.512 QT_NO_DEBUG . . . 299
9.47.1.513 QT_OPENGL_LIB . . . 300
9.47.1.514 QT_OPENGLWIDGETS_LIB . . . 300
9.47.1.515 QT_WIDGETS_LIB . . . 300
9.47.1.516 SIZEOF_DPTR . . . 300
9.47.1.517 TARGET_IPHONE_SIMULATOR . . . 300
9.47.1.518 TARGET_OS_ARROW . . . 300
9.47.1.519 TARGET_OS_BRIDGE . . . 300
9.47.1.520 TARGET_OS_DRIVERKIT . . . 300

9.47.1.521 TARGET_OS_EMBEDDED . . . 300
9.47.1.522 TARGET_OS_IOS . . . 300
9.47.1.523 TARGET_OS_IOSMAC . . . 301
9.47.1.524 TARGET_OS_IPHONE . . . 301
9.47.1.525 TARGET_OS_LINUX . . . 301
9.47.1.526 TARGET_OS_MAC . . . 301
9.47.1.527 TARGET_OS_MACCATALYST . . . 301
9.47.1.528 TARGET_OS_NANO . . . 301
9.47.1.529 TARGET_OS_OSX . . . 301
9.47.1.530 TARGET_OS_SIMULATOR . . . 301
9.47.1.531 TARGET_OS_TV . . . 301
9.47.1.532 TARGET_OS_UIKITFORMAC . . . 301
9.47.1.533 TARGET_OS_UNIX . . . 302
9.47.1.534 TARGET_OS_VISION . . . 302
9.47.1.535 TARGET_OS_WATCH . . . 302
9.47.1.536 TARGET_OS_WIN32 . . . 302
9.47.1.537 TARGET_OS_WINDOWS . . . 302
9.47.1.538 TARGET_OS_XR . . . 302
9.48 moc_predefs.h . . . 302
9.49 build/NetScanner_autogen/mocs_compilation.cpp 文件参考 . . . 308
9.50 NetScanner_autogen/mocs_compilation.cpp 文件参考 . . . 309
9.51 deviceanalyzer.cpp 文件参考 . . . 309
9.52 deviceanalyzer.h 文件参考 . . . 310
9.53 deviceanalyzer.h . . . 310
9.54 main.cpp 文件参考 . . . 311
9.54.1 函数说明 . . . 312
9.54.1.1 main() . . . 312
9.55 mainwindow.cpp 文件参考 . . . 312
9.56 mainwindow.h 文件参考 . . . 312
9.57 mainwindow.h . . . 313
9.58 networkscanner.cpp 文件参考 . . . 316
9.58.1 详细描述 . . . 316
9.59 networkscanner.h 文件参考 . . . 317
9.59.1 详细描述 . . . 317
9.60 networkscanner.h . . . 318
9.61 networktopology.cpp 文件参考 . . . 320
9.62 networktopology.h 文件参考 . . . 320
9.62.1 枚举类型说明 . . . 321
9.62.1.1 ConnectionType . . . 321
9.62.1.2 DeviceType . . . 322
9.63 networktopology.h . . . 322
9.64 README.dox 文件参考 . . . 325

9.65 README.md 文件参考 325
9.66 scanhistory.cpp 文件参考 325
9.67 scanhistory.h 文件参考 325
9.68 scanhistory.h 326

Index **329**

Chapter 1

Network Scanner 项目文档

1.1 简介

Network Scanner是一个用于网络扫描和拓扑分析的工具。 它可以帮助您发现网络中的设备，分析设备特性，并生成网络拓扑图。

1.2 主要功能

- 网络设备扫描
- 设备分析
- 网络拓扑可视化
- 扫描历史记录

1.3 项目架构

该项目主要包含以下模块：

- 主窗口(MainWindow)：用户界面
- 网络扫描器(NetworkScanner)：负责扫描网络设备
- 网络拓扑(NetworkTopology)：生成和展示网络拓扑图
- 设备分析器(DeviceAnalyzer)：分析设备特性
- 扫描历史(ScanHistory)：管理扫描历史记录

1.4 构建说明

该项目使用CMake构建系统，依赖于Qt6库。 构建步骤：

1. 确保已安装CMake和Qt6
2. 创建构建目录：`mkdir build && cd build`
3. 配置项目：`cmake ..`
4. 构建项目：`make`

Chapter 2

网络扫描器

一个基于Qt的综合性网络扫描工具，用于扫描局域网中的计算机，以可视化方式展示网络拓扑结构，并提供详细的设备分析。

2.1 主要功能

- 自动检测本地网络接口
- 扫描局域网内所有可达的计算机
- 显示主机IP地址、主机名、MAC地址和可连接状态
- 识别并显示设备的MAC地址厂商信息
- 直观的网络拓扑图可视化，显示网络设备之间的关系
- 智能设备类型识别（路由器、服务器、**PC**、移动设备等）
- 设备统计分析和图表展示（设备类型分布、厂商分布）
- 开放端口分布统计和图表展示
- 安全风险分析和安全报告生成
- 扫描历史记录和会话比较功能
- 支持明暗两种主题模式切换
- 实时显示扫描进度和结果
- 支持多线程并行扫描，提高扫描速度
- 可配置自定义端口扫描列表
- 可设置扫描超时时间
- 可指定自定义IP地址范围
- 可保存和导出扫描结果
- 可查看主机的详细信息和开放端口
- 支持设置保存和加载
- 支持计划任务扫描
- 过滤和搜索扫描结果

2.2 最新改进

- **扫描性能优化**：减少每个子网的扫描IP数量，更快完成扫描
- **超时控制**：添加全局扫描超时机制，防止扫描过程卡住
- **稳定性提升**：改进了错误处理，提高了应用在不同网络环境下的稳定性
- **资源管理**：优化线程和内存使用，减少资源占用
- **代码文档**：添加了完整的Doxygen风格注释，便于维护和开发
- **UI响应性**：改进了UI反馈机制，提供更好的用户体验

2.3 构建要求

- Qt 6.0或更高版本（也兼容Qt 5）
- C++17标准支持
- CMake 3.16或更高版本
- 需要包含QtCharts模块（Qt5需要单独安装，Qt6已包含）

2.4 构建步骤

1. 确保已安装Qt和CMake
2. 克隆或下载此项目
3. 创建构建目录：`mkdir build && cd build`
4. 运行CMake生成构建文件：`cmake ..`
5. 编译项目：`make` 或 `cmake --build .`
6. 运行程序：`./NetScanner`

```
cd /Users/lifulin/Desktop/NetworkScanner
rm -rf build
mkdir build
cd build
cmake ..
make -j8
./NetScanner
```

2.5 详细使用指南

2.5.1 基本扫描

1. 启动应用程序
2. 可选：在"扫描设置"标签页配置扫描参数
   - 可以设置自定义端口列表，格式为逗号分隔的端口号，例如：`21,22,23,80,443`
   - 可以设置连接超时时间（100-10000毫秒之间），默认为500毫秒
   - 可以设置自定义IP地址范围，例如从`192.168.1.1`到`192.168.1.254`
3. 返回"扫描结果"标签页，点击"开始扫描"按钮
4. 等待扫描完成（进度条显示当前扫描进度）
5. 查看结果表格中显示的主机信息，包括IP地址、主机名、MAC地址和厂商信息
6. 双击任一主机查看其详细信息，包括所有开放端口

2.5.2 网络拓扑

1. 完成扫描后，切换到"网络拓扑"标签页
2. 查看自动生成的网络拓扑图：
   - 中心节点通常是网关或路由器（红色图标）
   - 服务器显示为蓝色图标
   - 个人电脑显示为绿色图标
   - 移动设备显示为橙色图标
   - 打印机显示为紫色图标
   - 智能设备显示为青色图标
3. 使用"自动布局"按钮重新排列设备位置
4. 使用"+"和"-"按钮放大或缩小视图
5. 使用"重置视图"按钮恢复默认视图
6. 鼠标悬停在设备节点上可查看设备详细信息
7. 点击设备节点可选中该设备，并自动在"扫描结果"中突出显示
8. 通过"保存网络拓扑图"功能可导出当前拓扑图为PNG或JPG图像文件

2.5.3 设备分析

1. 完成扫描后，切换到"统计分析"标签页
2. 查看三种图表：
   - **设备类型分布**：饼图显示不同类型设备（路由器、服务器、PC、移动设备等）的数量和百分比
   - **厂商分布**：饼图显示不同厂商设备的数量和百分比，展示前5大厂商，其余归为"其他"
   - **端口分布**：柱状图显示常见端口（如21、22、80、443等）的开放情况
3. 点击"生成安全报告"按钮获取网络安全分析和建议，报告将包含：
   - 设备数量统计
   - 高风险开放端口清单（如21/FTP、22/SSH、23/Telnet等）
   - 每台设备的安全风险评估
   - 具体的安全加固建议

2.5.4 扫描历史

1. 每次完成扫描后，结果会自动保存到扫描历史
2. 在"扫描历史"标签页查看历史记录：
   - 从下拉列表中选择历史会话查看具体结果
   - 点击"加载会话"按钮可将历史结果加载到当前视图
   - 点击"删除会话"按钮可删除不需要的历史记录
3. 使用"比较会话"功能对比两次扫描结果：
   - 选择两个不同的扫描会话进行比较
   - 查看"新增主机"和"消失主机"两个标签页了解网络变化
   - 识别网络中的新设备或已移除设备
4. 通过"文件→保存扫描历史"菜单将历史记录保存到JSON文件中
5. 通过"文件→加载扫描历史"菜单从JSON文件加载历史记录

2.5.5 计划任务

1. 通过"工具→计划扫描"菜单设置定时扫描任务
2. 在弹出的对话框中：
    - 选择扫描的日期和时间
    - 设置是否使用当前扫描设置
    - 配置是否自动保存结果
    - 设置是否在扫描完成后显示通知
3. 点击确定后，系统将在指定时间自动执行扫描
4. 扫描结果将按照设置保存，并可选择性地显示通知

2.5.6 结果过滤

1. 使用扫描结果页面顶部的过滤工具栏筛选结果：
    - 按IP地址过滤（支持部分匹配）
    - 按厂商过滤（从下拉列表选择）
    - 按设备类型过滤（路由器、服务器、PC等）
2. 点击"应用过滤"按钮应用筛选条件
3. 点击"清除过滤"恢复显示所有结果

2.5.7 暗色模式

- 通过"视图→暗色模式"菜单项切换明暗主题
- 暗色模式下所有UI元素（包括图表）将采用深色背景，更适合夜间使用
- 设置将自动保存，下次启动时恢复上次的主题设置

2.6 MAC地址厂商识别

应用程序内置了常见设备制造商的MAC地址前缀数据库，可以识别出许多设备的厂商，例如：

- 苹果 (Apple)
- 微软 (Microsoft)
- 华为 (Huawei)
- 思科 (Cisco)
- 小米 (Xiaomi)
- TP-Link
- 联想 (Lenovo)
- 华硕 (ASUS)
- VMware虚拟机 等多种设备。

程序支持多种格式的MAC地址解析，包括：

- 标准冒号分隔格式：`00:1A:2B:3C:4D:5E`
- 连字符分隔格式：`00-1A-2B-3C-4D-5E`
- 不规则格式：例如某些系统输出的单个字符格式，如`0:1A:2B:3:4D:5E`

2.7 技术细节

- **网络发现机制**：使用优化的算法探测局域网设备，包括自适应超时设置和智能重试机制
- **多线程扫描**：根据系统CPU核心数自动调整并行扫描线程数
- **伪MAC地址生成**：当无法通过系统ARP表获取真实MAC地址时，使用基于IP的一致性伪MAC地址生成
- **超时控制**：全局扫描超时保护机制，防止在网络环境较差时应用程序卡住
- **代码文档**：完整的Doxygen格式代码注释，便于开发和维护

2.8 故障排除

2.8.1 常见问题

1. 无法检测到网络接口
   - 确保用户有足够的权限访问网络接口
   - 在某些系统上，可能需要管理员/root权限运行程序
2. 扫描卡在**99**不动
   - 已修复此问题，现在添加了120秒的全局超时限制
   - 如果仍然出现此问题，尝试减少自定义IP范围或使用"快速扫描"模式
3. 扫描速度慢
   - 尝试减少要扫描的端口数量
   - 减小超时设置值（默认为500毫秒）
   - 请注意，设置过小的超时值可能导致漏检主机
4. 某些设备没有被检测到
   - 某些设备可能配置了防火墙阻止ICMP和常见端口
   - 尝试增加超时设置值
   - 考虑在扫描设置中添加设备可能开放的特定端口
5. 拓扑图不显示某些设备之间的连接
   - 当前版本使用简化的拓扑推断算法
   - 所有设备默认连接到检测到的路由器/网关

2.8.2 性能优化

- 扫描大型网络（超过254个地址）时，建议使用自定义IP范围，分段扫描
- 在配置较低的计算机上，可以减少扫描的端口数量以提高速度
- 禁用"自动保存结果"可减少内存占用

2.9 开发注意事项

2.9.1 QtCharts命名空间

此项目使用QtCharts库进行数据可视化，需要注意：

1. 在CMakeLists.txt中，我们使用 `-DQT_CHARTS_USE_NAMESPACE` 宏定义来简化命名空间的使用
2. 如果您直接在代码中使用 `using namespace QtCharts;`，在某些Qt版本中可能导致编译错误
3. 当使用QtCharts类时，推荐以下两种方式之一：
   - 使用完整命名空间：`QtCharts::QChart`
   - 在CMakeLists.txt中添加全局宏定义：`add_definitions(-DQT_CHARTS_USE_NAMESPACE)`

2.9.2 内存管理和界面初始化

为确保程序稳定性：

1. 确保所有QWidget对象都有正确的父子关系，避免浮动窗口
2. 在MainWindow构造函数中，先创建UI组件，再创建功能对象（如扫描器、分析器等）
3. 信号槽连接前，确保对应的对象已正确初始化
4. 图表组件的创建和使用需要特别注意其生命周期管理

2.10 版本历史

2.10.1 v2.2.0

- 增加了扫描超时控制，防止在部分网络环境下卡在99%
- 优化扫描性能，减少扫描IP数量，更快完成扫描
- 提高了应用程序稳定性，改进了错误处理机制
- 添加了完整的Doxygen格式代码注释
- 优化了MAC地址处理算法，增强了多种格式MAC地址的兼容性

2.10.2 v2.1.0

- 修复了MAC地址解析问题，增强了对不规则MAC地址格式的支持
- 实现了网络拓扑图的缩放和重置功能
- 优化了设备类型识别算法
- 改进了主窗口组件初始化顺序，提高程序稳定性

2.10.3 v2.0.0

- 添加了网络拓扑可视化功能
- 添加了设备类型识别功能
- 添加了扫描历史和会话比较功能
- 添加了统计分析和图表展示功能
- 添加了安全报告生成功能
- 添加了计划任务扫描功能
- 添加了结果过滤功能
- 添加了暗色模式

2.10.4 v1.0.0

- 初始版本
- 基本网络扫描功能
- 显示主机IP、主机名和MAC地址
- 支持端口扫描
- 支持MAC厂商识别

项目GitHub：https://github.com/example/networkscanner

Chapter 3

命名空间索引

3.1 命名空间列表

这里列出了所有命名空间定义，附带简要说明:

QT_WARNING_DISABLE_DEPRECATED 19

Chapter 4

继承关系索引

4.1 类继承关系

此继承关系列表按字典顺序粗略的排序:

HostInfo 39
QGraphicsItem
    ConnectionLine 21
    DeviceNode 32
QGraphicsView
    NetworkTopologyView 95
QMainWindow
    MainWindow 41
QObject
    NetworkScanner 69
    ScanHistory 103
QRunnable
    ScanTask 115
QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN10MainWindowE_t 101
QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN11ScanHistoryE_t 101
QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN14DeviceAnalyzerE_t 102
QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN14NetworkScannerE_t 102
QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN19NetworkTopologyViewE_t 103
QWidget
    DeviceAnalyzer 24
    NetworkTopology 90
ScanSession 109
ScanStrategy 111
TopologyAnalyzer 118

Chapter 5

类索引

5.1 类列表

这里列出了所有类、结构、联合以及接口定义等，并附带简要说明:

ConnectionLine 21
DeviceAnalyzer 24
DeviceNode 32
HostInfo
存储主机信息的结构体 39
MainWindow 41
NetworkScanner
网络扫描器类 69
NetworkTopology 90
NetworkTopologyView 95
QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN10MainWindowE_t 101
QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN11ScanHistoryE_t 101
QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN14DeviceAnalyzerE_t 102
QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN14NetworkScannerE_t 102
QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN19NetworkTopologyViewE_t 103
ScanHistory 103
ScanSession 109
ScanStrategy
扫描策略类 111
ScanTask
扫描任务类 115
TopologyAnalyzer 118

Chapter 6

文件索引

6.1 文件列表

这里列出了所有文件，并附带简要说明:

deviceanalyzer.cpp . . . 309
deviceanalyzer.h . . . 310
main.cpp . . . 311
mainwindow.cpp . . . 312
mainwindow.h . . . 312
networkscanner.cpp
网络扫描器类的实现 . . . 316
networkscanner.h
网络扫描器类定义 . . . 317
networktopology.cpp . . . 320
networktopology.h . . . 320
scanhistory.cpp . . . 325
scanhistory.h . . . 325
build/CMakeFiles/3.31.5/CompilerIdCXX/CMakeCXXCompilerId.cpp . . . 123
build/CMakeFiles/NetScanner.dir/deviceanalyzer.cpp.o.d . . . 127
build/CMakeFiles/NetScanner.dir/main.cpp.o.d . . . 127
build/CMakeFiles/NetScanner.dir/mainwindow.cpp.o.d . . . 127
build/CMakeFiles/NetScanner.dir/networkscanner.cpp.o.d . . . 127
build/CMakeFiles/NetScanner.dir/networktopology.cpp.o.d . . . 127
build/CMakeFiles/NetScanner.dir/scanhistory.cpp.o.d . . . 127
build/CMakeFiles/NetScanner.dir/NetScanner_autogen/mocs_compilation.cpp.o.d . . . 127
build/NetScanner_autogen/moc_predefs.h . . . 168
build/NetScanner_autogen/mocs_compilation.cpp . . . 308
build/NetScanner_autogen/EWIEGA46WW/moc_deviceanalyzer.cpp . . . 127
build/NetScanner_autogen/EWIEGA46WW/moc_deviceanalyzer.cpp.d . . . 135
build/NetScanner_autogen/EWIEGA46WW/moc_mainwindow.cpp . . . 135
build/NetScanner_autogen/EWIEGA46WW/moc_mainwindow.cpp.d . . . 148
build/NetScanner_autogen/EWIEGA46WW/moc_networkscanner.cpp . . . 148
build/NetScanner_autogen/EWIEGA46WW/moc_networkscanner.cpp.d . . . 155
build/NetScanner_autogen/EWIEGA46WW/moc_networktopology.cpp . . . 155
build/NetScanner_autogen/EWIEGA46WW/moc_networktopology.cpp.d . . . 163
build/NetScanner_autogen/EWIEGA46WW/moc_scanhistory.cpp . . . 163
build/NetScanner_autogen/EWIEGA46WW/moc_scanhistory.cpp.d . . . 168
NetScanner_autogen/moc_predefs.h . . . 238
NetScanner_autogen/mocs_compilation.cpp . . . 309

NetScanner_autogen/EWIEGA46WW/moc_deviceanalyzer.cpp 130
NetScanner_autogen/EWIEGA46WW/moc_deviceanalyzer.cpp.d 135
NetScanner_autogen/EWIEGA46WW/moc_mainwindow.cpp 139
NetScanner_autogen/EWIEGA46WW/moc_mainwindow.cpp.d 148
NetScanner_autogen/JRIAJ772TK/moc_deviceanalyzer.cpp 132
NetScanner_autogen/JRIAJ772TK/moc_deviceanalyzer.cpp.d 135
NetScanner_autogen/JRIAJ772TK/moc_mainwindow.cpp 144
NetScanner_autogen/JRIAJ772TK/moc_mainwindow.cpp.d 148
NetScanner_autogen/JRIAJ772TK/moc_networkscanner.cpp 152
NetScanner_autogen/JRIAJ772TK/moc_networkscanner.cpp.d 155
NetScanner_autogen/JRIAJ772TK/moc_networktopology.cpp 159
NetScanner_autogen/JRIAJ772TK/moc_networktopology.cpp.d 163
NetScanner_autogen/JRIAJ772TK/moc_scanhistory.cpp 165
NetScanner_autogen/JRIAJ772TK/moc_scanhistory.cpp.d 168

Chapter 7

命名空间文档

7.1 QT_WARNING_DISABLE_DEPRECATED 命名空间参考

类

- struct qt_meta_tag_ZN10MainWindowE_t
- struct qt_meta_tag_ZN11ScanHistoryE_t
- struct qt_meta_tag_ZN14DeviceAnalyzerE_t
- struct qt_meta_tag_ZN14NetworkScannerE_t
- struct qt_meta_tag_ZN19NetworkTopologyViewE_t

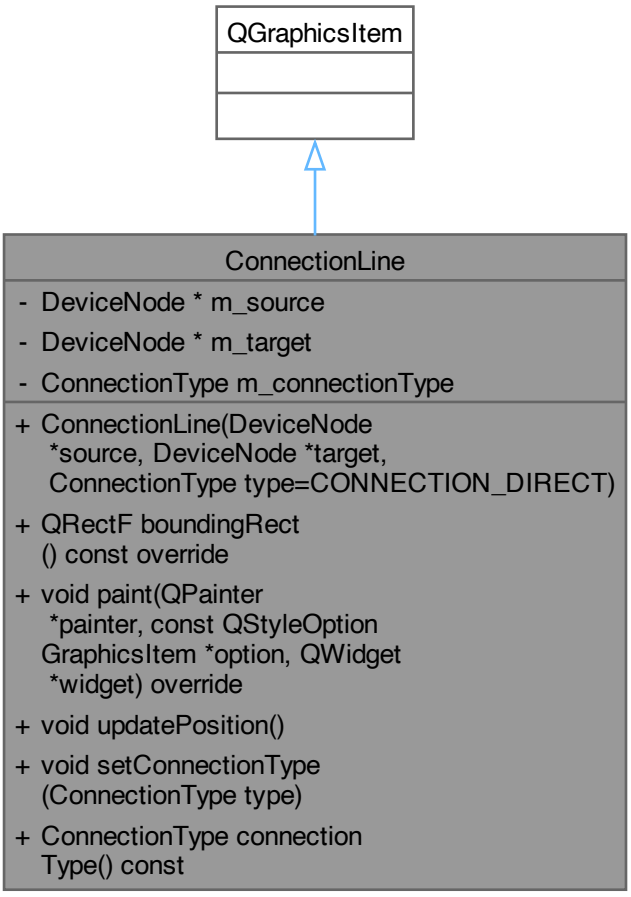
Chapter 8

类说明

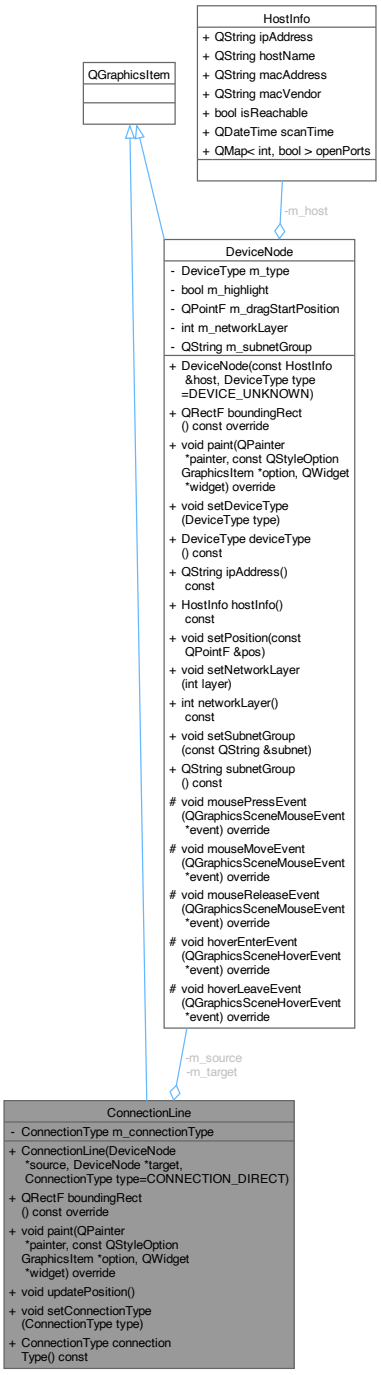
8.1 ConnectionLine类 参考

`#include <networktopology.h>`

类 ConnectionLine 继承关系图:



ConnectionLine 的协作图:



Public 成员函数

- ConnectionLine (DeviceNode ∗source, DeviceNode ∗target, ConnectionType type=CONNECTION_DIRECT)
- QRectF boundingRect () const override
- void paint (QPainter ∗painter, const QStyleOptionGraphicsItem ∗option, QWidget ∗widget) override
- void updatePosition ()
- void setConnectionType (ConnectionType type)
- ConnectionType connectionType () const

Private 属性

- DeviceNode * m_source
- DeviceNode * m_target
- ConnectionType m_connectionType

8.1.1 构造及析构函数说明

8.1.1.1 ConnectionLine()

```
ConnectionLine::ConnectionLine (
            DeviceNode * source,
            DeviceNode * target,
            ConnectionType type = CONNECTION_DIRECT)
```

函数调用图:

8.1.2 成员函数说明

8.1.2.1 boundingRect()

```
QRectF ConnectionLine::boundingRect () const  [override]
```

8.1.2.2 connectionType()

```
ConnectionType ConnectionLine::connectionType () const  [inline]
```

8.1.2.3 paint()

```
void ConnectionLine::paint (
            QPainter * painter,
            const QStyleOptionGraphicsItem * option,
            QWidget * widget)  [override]
```

8.1.2.4 setConnectionType()

```
void ConnectionLine::setConnectionType (
            ConnectionType type)
```

8.1.2.5 updatePosition()

```
void ConnectionLine::updatePosition ()
```

这是这个函数的调用关系图:

8.1.3 类成员变量说明

8.1.3.1 m_connectionType

```
ConnectionType ConnectionLine::m_connectionType  [private]
```

8.1.3.2 m_source

```
DeviceNode* ConnectionLine::m_source  [private]
```

8.1.3.3 m_target

```
DeviceNode* ConnectionLine::m_target  [private]
```

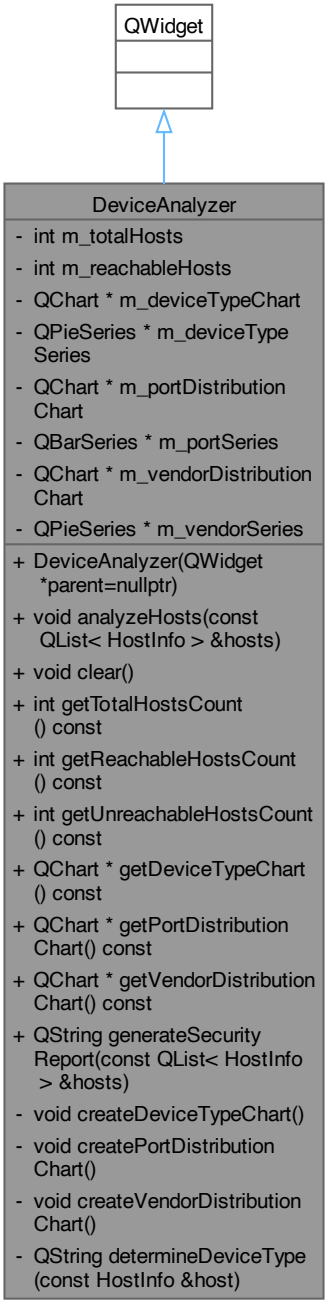
该类的文档由以下文件生成:

- networktopology.h
- networktopology.cpp

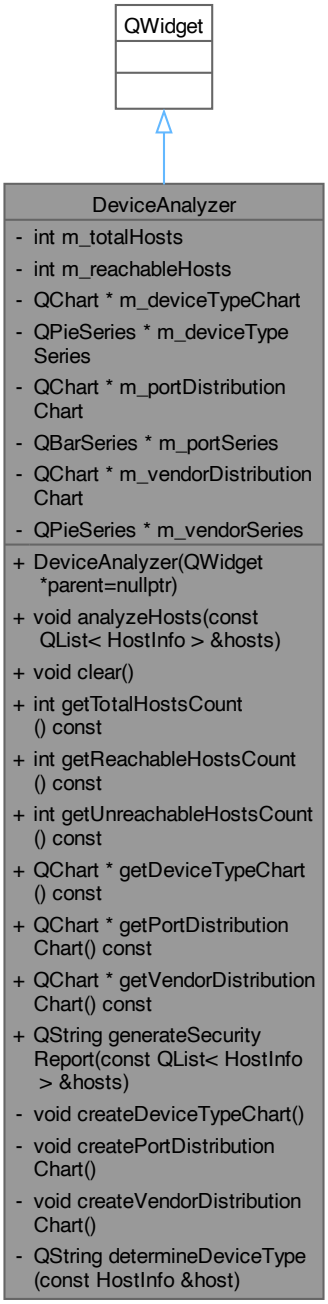
8.2 DeviceAnalyzer类 参考

```
#include <deviceanalyzer.h>
```

类 DeviceAnalyzer 继承关系图:



DeviceAnalyzer 的协作图:



信号

- void analysisCompleted ()

Public 成员函数

- DeviceAnalyzer (QWidget *parent=nullptr)

- void analyzeHosts (const QList< HostInfo > &hosts)
- void clear ()
- int getTotalHostsCount () const
- int getReachableHostsCount () const
- int getUnreachableHostsCount () const
- QChart * getDeviceTypeChart () const
- QChart * getPortDistributionChart () const
- QChart * getVendorDistributionChart () const
- QString generateSecurityReport (const QList< HostInfo > &hosts)

Private 成员函数

- void createDeviceTypeChart ()
- void createPortDistributionChart ()
- void createVendorDistributionChart ()
- QString determineDeviceType (const HostInfo &host)

Private 属性

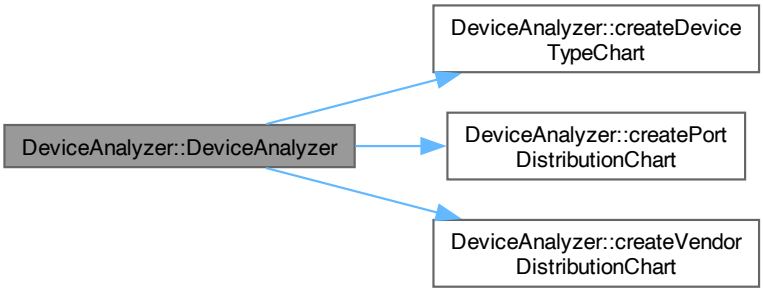
- int m_totalHosts
- int m_reachableHosts
- QChart * m_deviceTypeChart
- QPieSeries * m_deviceTypeSeries
- QChart * m_portDistributionChart
- QBarSeries * m_portSeries
- QChart * m_vendorDistributionChart
- QPieSeries * m_vendorSeries

8.2.1 构造及析构函数说明

8.2.1.1 DeviceAnalyzer()

```
DeviceAnalyzer::DeviceAnalyzer (
            QWidget * parent = nullptr)
```

函数调用图:

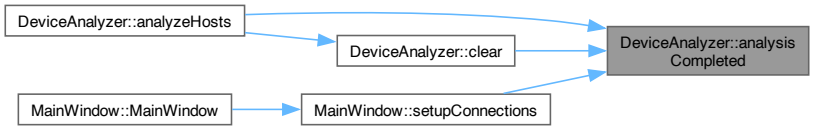


8.2.2 成员函数说明

8.2.2.1 analysisCompleted

```
void DeviceAnalyzer::analysisCompleted ()  [signal]
```

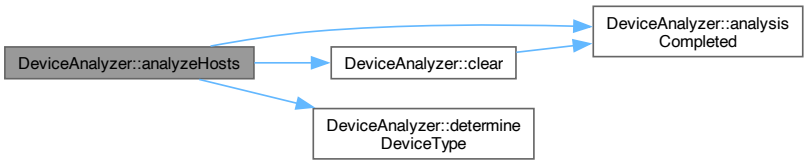
这是这个函数的调用关系图:



8.2.2.2 analyzeHosts()

```
void DeviceAnalyzer::analyzeHosts (
            const QList< HostInfo > & hosts)
```

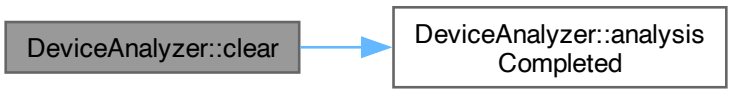
函数调用图:



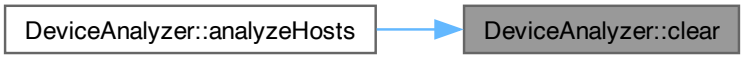
8.2.2.3 clear()

```
void DeviceAnalyzer::clear ()
```

函数调用图:



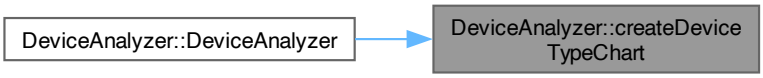
这是这个函数的调用关系图:



8.2.2.4 createDeviceTypeChart()

```
void DeviceAnalyzer::createDeviceTypeChart ()  [private]
```

这是这个函数的调用关系图:



8.2.2.5 createPortDistributionChart()

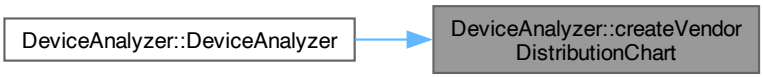
```
void DeviceAnalyzer::createPortDistributionChart ()  [private]
```

这是这个函数的调用关系图:

8.2.2.6 createVendorDistributionChart()

```
void DeviceAnalyzer::createVendorDistributionChart ()  [private]
```

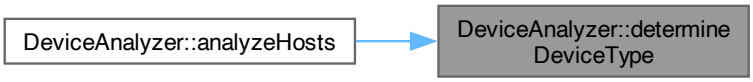
这是这个函数的调用关系图:



8.2.2.7 determineDeviceType()

```
QString DeviceAnalyzer::determineDeviceType (
            const HostInfo & host)  [private]
```

这是这个函数的调用关系图:



8.2.2.8 generateSecurityReport()

```
QString DeviceAnalyzer::generateSecurityReport (
            const QList< HostInfo > & hosts)
```

8.2.2.9 getDeviceTypeChart()

```
QChart * DeviceAnalyzer::getDeviceTypeChart () const  [inline]
```

8.2.2.10 getPortDistributionChart()

```
QChart * DeviceAnalyzer::getPortDistributionChart () const  [inline]
```

8.2.2.11 getReachableHostsCount()

```
int DeviceAnalyzer::getReachableHostsCount () const  [inline]
```

8.2.2.12 getTotalHostsCount()

```
int DeviceAnalyzer::getTotalHostsCount () const  [inline]
```

8.2.2.13 getUnreachableHostsCount()

```
int DeviceAnalyzer::getUnreachableHostsCount () const  [inline]
```

8.2.2.14 getVendorDistributionChart()

```
QChart * DeviceAnalyzer::getVendorDistributionChart () const  [inline]
```

8.2.3 类成员变量说明

8.2.3.1 m_deviceTypeChart

```
QChart* DeviceAnalyzer::m_deviceTypeChart  [private]
```

8.2.3.2 m_deviceTypeSeries

```
QPieSeries* DeviceAnalyzer::m_deviceTypeSeries  [private]
```

8.2.3.3 m_portDistributionChart

```
QChart* DeviceAnalyzer::m_portDistributionChart  [private]
```

8.2.3.4 m_portSeries

```
QBarSeries* DeviceAnalyzer::m_portSeries  [private]
```

8.2.3.5 m_reachableHosts

```
int DeviceAnalyzer::m_reachableHosts  [private]
```

8.2.3.6 m_totalHosts

```
int DeviceAnalyzer::m_totalHosts  [private]
```

8.2.3.7 m_vendorDistributionChart

```
QChart* DeviceAnalyzer::m_vendorDistributionChart  [private]
```

8.2.3.8 m_vendorSeries

```
QPieSeries* DeviceAnalyzer::m_vendorSeries  [private]
```

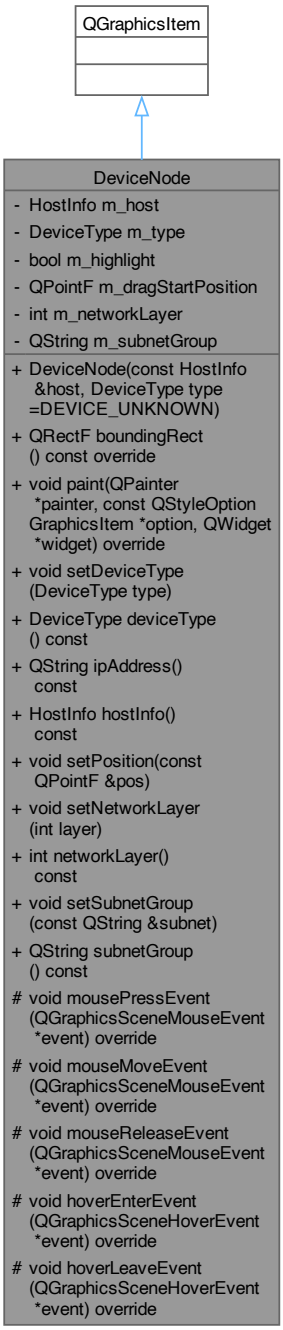
该类的文档由以下文件生成:

- deviceanalyzer.h
- build/NetScanner_autogen/EWIEGA46WW/moc_deviceanalyzer.cpp
- deviceanalyzer.cpp
- NetScanner_autogen/EWIEGA46WW/moc_deviceanalyzer.cpp
- NetScanner_autogen/JRIAJ772TK/moc_deviceanalyzer.cpp

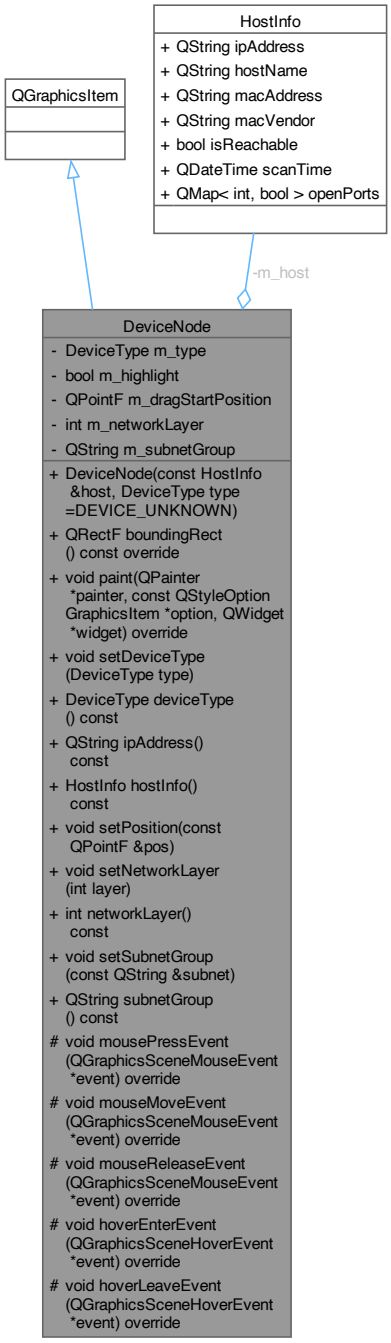
8.3 DeviceNode类 参考

```
#include <networktopology.h>
```

类 DeviceNode 继承关系图:



DeviceNode 的协作图:



Public 成员函数

- DeviceNode (const HostInfo &host, DeviceType type=DEVICE_UNKNOWN)
- QRectF boundingRect () const override
- void paint (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget) override
- void setDeviceType (DeviceType type)
- DeviceType deviceType () const

- QString ipAddress () const
- HostInfo hostInfo () const
- void setPosition (const QPointF &pos)
- void setNetworkLayer (int layer)
- int networkLayer () const
- void setSubnetGroup (const QString &subnet)
- QString subnetGroup () const

Protected 成员函数

- void mousePressEvent (QGraphicsSceneMouseEvent *event) override
- void mouseMoveEvent (QGraphicsSceneMouseEvent *event) override
- void mouseReleaseEvent (QGraphicsSceneMouseEvent *event) override
- void hoverEnterEvent (QGraphicsSceneHoverEvent *event) override
- void hoverLeaveEvent (QGraphicsSceneHoverEvent *event) override

Private 属性

- HostInfo m_host
- DeviceType m_type
- bool m_highlight
- QPointF m_dragStartPosition
- int m_networkLayer
- QString m_subnetGroup

8.3.1 构造及析构函数说明

8.3.1.1 DeviceNode()

```
DeviceNode::DeviceNode (
            const HostInfo & host,
            DeviceType type = DEVICE_UNKNOWN)
```

8.3.2 成员函数说明

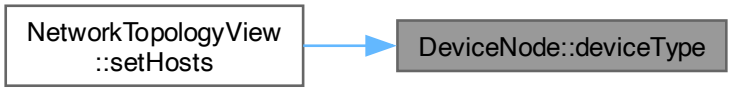
8.3.2.1 boundingRect()

```
QRectF DeviceNode::boundingRect () const  [override]
```

8.3.2.2 deviceType()

```
DeviceType DeviceNode::deviceType () const  [inline]
```

这是这个函数的调用关系图:



8.3.2.3 hostInfo()

```
HostInfo DeviceNode::hostInfo () const  [inline]
```

8.3.2.4 hoverEnterEvent()

```
void DeviceNode::hoverEnterEvent (
            QGraphicsSceneHoverEvent * event)  [override], [protected]
```

8.3.2.5 hoverLeaveEvent()

```
void DeviceNode::hoverLeaveEvent (
            QGraphicsSceneHoverEvent * event)  [override], [protected]
```

8.3.2.6 ipAddress()

```
QString DeviceNode::ipAddress () const  [inline]
```

8.3.2.7 mouseMoveEvent()

```
void DeviceNode::mouseMoveEvent (
            QGraphicsSceneMouseEvent * event)  [override], [protected]
```

8.3.2.8 mousePressEvent()

```
void DeviceNode::mousePressEvent (
            QGraphicsSceneMouseEvent * event)  [override], [protected]
```

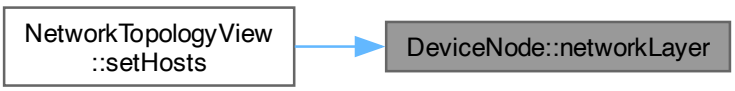
8.3.2.9 mouseReleaseEvent()

```
void DeviceNode::mouseReleaseEvent (
            QGraphicsSceneMouseEvent * event)  [override], [protected]
```

8.3.2.10 networkLayer()

```
int DeviceNode::networkLayer () const  [inline]
```

这是这个函数的调用关系图:



8.3.2.11 paint()

```
void DeviceNode::paint (
            QPainter * painter,
            const QStyleOptionGraphicsItem * option,
            QWidget * widget)  [override]
```

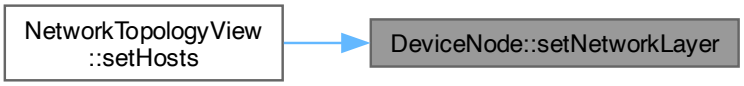
8.3.2.12 setDeviceType()

```
void DeviceNode::setDeviceType (
            DeviceType type)
```

8.3.2.13 setNetworkLayer()

```
void DeviceNode::setNetworkLayer (
            int layer)  [inline]
```

这是这个函数的调用关系图:



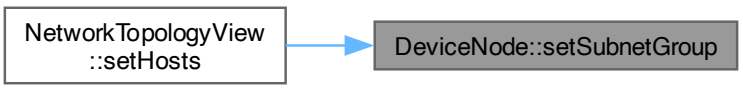
8.3.2.14 setPosition()

```
void DeviceNode::setPosition (
            const QPointF & pos)
```

8.3.2.15 setSubnetGroup()

```
void DeviceNode::setSubnetGroup (
            const QString & subnet)  [inline]
```

这是这个函数的调用关系图:



8.3.2.16 subnetGroup()

```
QString DeviceNode::subnetGroup () const  [inline]
```

8.3.3 类成员变量说明

8.3.3.1 m_dragStartPosition

```
QPointF DeviceNode::m_dragStartPosition  [private]
```

8.3.3.2 m_highlight

```
bool DeviceNode::m_highlight  [private]
```

8.3.3.3 m_host

```
HostInfo DeviceNode::m_host  [private]
```

8.3.3.4 m_networkLayer

```
int DeviceNode::m_networkLayer  [private]
```

8.3.3.5 m_subnetGroup

```
QString DeviceNode::m_subnetGroup  [private]
```

8.3.3.6 m_type

```
DeviceType DeviceNode::m_type  [private]
```

该类的文档由以下文件生成:

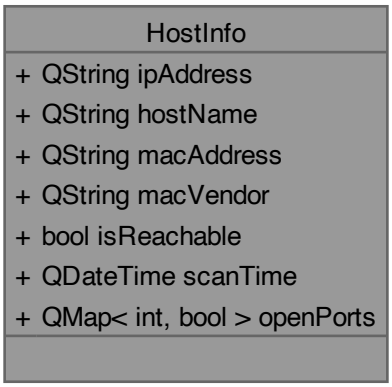
- networktopology.h
- networktopology.cpp

8.4 HostInfo结构体 参考

存储主机信息的结构体

`#include <networkscanner.h>`

HostInfo 的协作图:



Public 属性

- QString ipAddress
  主机*IP*地址
- QString hostName
  主机名称
- QString macAddress
  *MAC*物理地址
- QString macVendor
  *MAC*地址对应的厂商
- bool isReachable
  主机是否可达
- QDateTime scanTime
  扫描时间
- QMap< int, bool > openPorts
  开放的端口及状态 *(*端口号 -> 是否开放*)*

8.4.1 详细描述

存储主机信息的结构体

包含IP地址、主机名、MAC地址、厂商信息等扫描结果

8.4.2 类成员变量说明

8.4.2.1 hostName

```
QString HostInfo::hostName
```

主机名称

8.4.2.2 ipAddress

```
QString HostInfo::ipAddress
```

主机IP地址

8.4.2.3 isReachable

```
bool HostInfo::isReachable
```

主机是否可达

8.4.2.4 macAddress

```
QString HostInfo::macAddress
```

MAC物理地址

8.4.2.5 macVendor

```
QString HostInfo::macVendor
```

MAC地址对应的厂商

8.4.2.6 openPorts

```
QMap<int, bool> HostInfo::openPorts
```

开放的端口及状态 (端口号 -> 是否开放)

8.4.2.7 scanTime

```
QDateTime HostInfo::scanTime
```

扫描时间

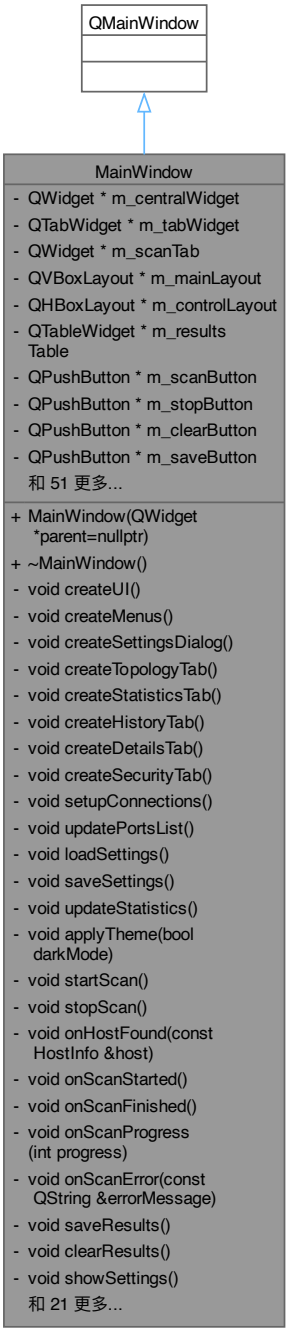
该结构体的文档由以下文件生成:

- networkscanner.h

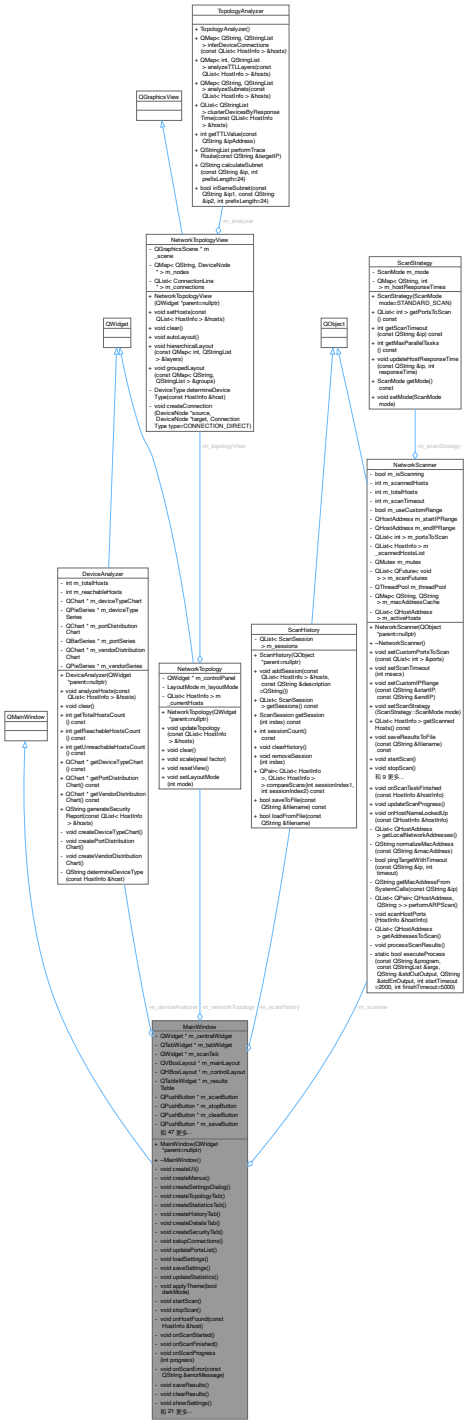
8.5 MainWindow类 参考

`#include <mainwindow.h>`

类 MainWindow 继承关系图:



MainWindow 的协作图:



Public 成员函数

- MainWindow (QWidget *parent=nullptr)
- ~MainWindow ()

Private 槽

- void startScan ()

- void stopScan ()
- void onHostFound (const HostInfo &host)
- void onScanStarted ()
- void onScanFinished ()
- void onScanProgress (int progress)
- void onScanError (const QString &errorMessage)
- void saveResults ()
- void clearResults ()
- void showSettings ()
- void applySettings ()
- void showAbout ()
- void showHostDetails (int row, int column)
- void exportToCSV ()
- void togglePortScanOptions (bool checked)
- void toggleRangeOptions (bool checked)
- void showTopologyView ()
- void showStatisticsView ()
- void showHistoryView ()
- void generateSecurityReport ()
- void saveTopologyImage ()
- void toggleDarkMode (bool enable)
- void compareScanResults ()
- void scheduleScan ()
- void saveHistoryToFile ()
- void loadHistoryFromFile ()
- void updateNetworkTopology ()
- void refreshTopology ()
- void filterResults ()
- void clearFilters ()
- void onThemeChanged ()

Private 成员函数

- void createUI ()
- void createMenus ()
- void createSettingsDialog ()
- void createTopologyTab ()
- void createStatisticsTab ()
- void createHistoryTab ()
- void createDetailsTab ()
- void createSecurityTab ()
- void setupConnections ()
- void updatePortsList ()
- void loadSettings ()
- void saveSettings ()
- void updateStatistics ()
- void applyTheme (bool darkMode)

Private 属性

- QWidget * m_centralWidget
- QTabWidget * m_tabWidget
- QWidget * m_scanTab
- QVBoxLayout * m_mainLayout
- QHBoxLayout * m_controlLayout
- QTableWidget * m_resultsTable
- QPushButton * m_scanButton
- QPushButton * m_stopButton
- QPushButton * m_clearButton
- QPushButton * m_saveButton
- QProgressBar * m_progressBar
- QLabel * m_statusLabel
- QStatusBar * m_statusBar
- QWidget * m_settingsTab
- QVBoxLayout * m_settingsLayout
- QGroupBox * m_portsGroupBox
- QCheckBox * m_customPortsCheckBox
- QLineEdit * m_portsLineEdit
- QSpinBox * m_timeoutSpinBox
- QGroupBox * m_rangeGroupBox
- QCheckBox * m_customRangeCheckBox
- QLineEdit * m_startIPLineEdit
- QLineEdit * m_endIPLineEdit
- QWidget * m_detailsTab
- QVBoxLayout * m_detailsLayout
- QTextEdit * m_detailsTextEdit
- QWidget * m_topologyTab
- NetworkTopology * m_networkTopology
- QWidget * m_statisticsTab
- DeviceAnalyzer * m_deviceAnalyzer
- QChartView * m_deviceTypeChartView
- QChartView * m_vendorChartView
- QChartView * m_portDistributionChartView
- QTextEdit * m_securityReportText
- QWidget * m_historyTab
- ScanHistory * m_scanHistory
- QComboBox * m_sessionComboBox
- QTableWidget * m_historyTable
- QMenu * m_fileMenu
- QMenu * m_viewMenu
- QMenu * m_toolsMenu
- QMenu * m_helpMenu
- QAction * m_exportAction
- QAction * m_saveHistoryAction
- QAction * m_loadHistoryAction
- QAction * m_saveTopologyAction
- QAction * m_exitAction
- QAction * m_settingsAction
- QAction * m_darkModeAction
- QAction * m_scheduleScanAction
- QAction * m_aboutAction
- QWidget * m_filterWidget
- QLineEdit * m_filterIPLineEdit

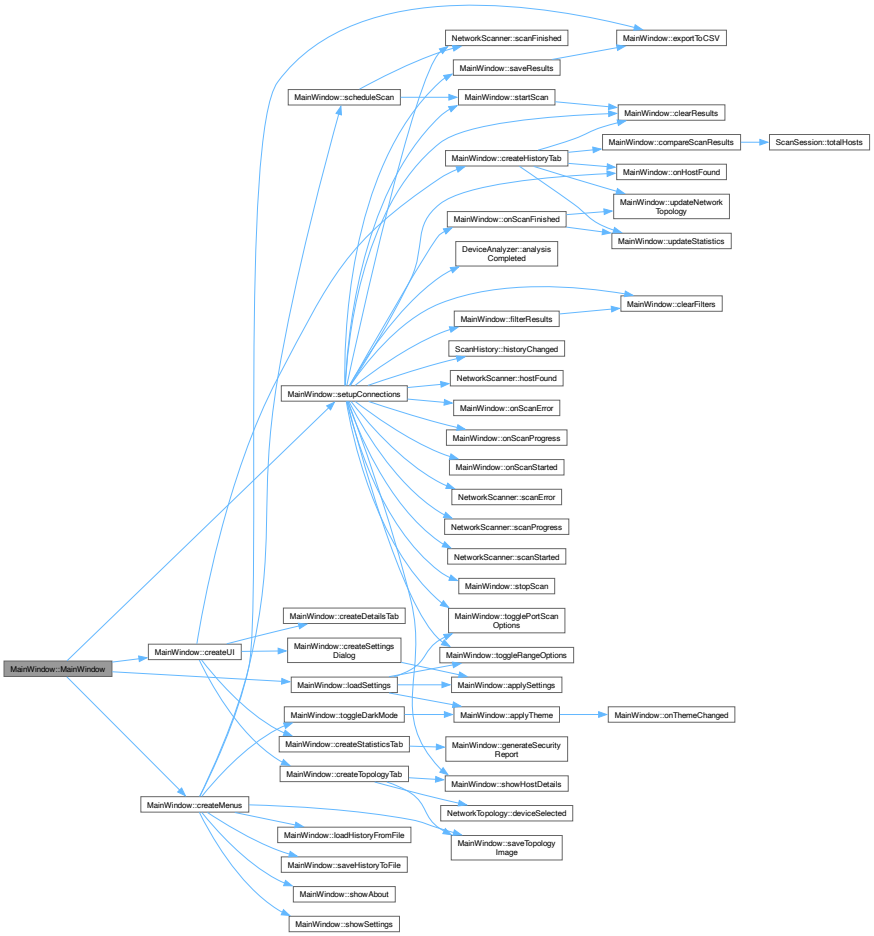
- QComboBox * m_filterVendorComboBox
- QComboBox * m_filterTypeComboBox
- QPushButton * m_filterButton
- QPushButton * m_clearFilterButton
- NetworkScanner * m_scanner
- int m_hostsFound
- int m_currentHostIndex
- bool m_darkModeEnabled

8.5.1 构造及析构函数说明

8.5.1.1 MainWindow()

```
MainWindow::MainWindow (
            QWidget * parent = nullptr)
```

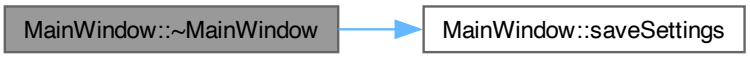
函数调用图:



8.5.1.2 ~MainWindow()

```
MainWindow::~MainWindow ()
```

函数调用图:

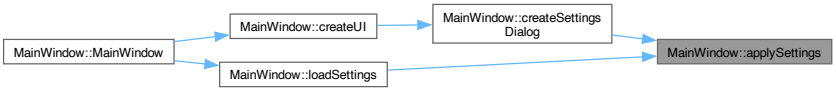


8.5.2 成员函数说明

8.5.2.1 applySettings

```
void MainWindow::applySettings ()  [private], [slot]
```

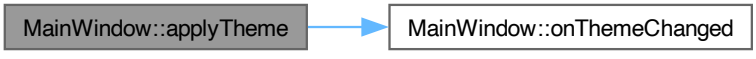
这是这个函数的调用关系图:



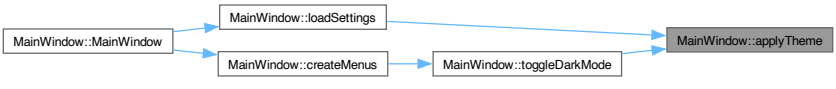
8.5.2.2 applyTheme()

```
void MainWindow::applyTheme (
            bool darkMode)  [private]
```

函数调用图:



这是这个函数的调用关系图:



8.5.2.3 clearFilters

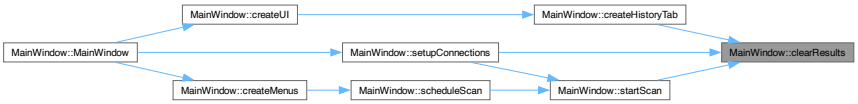
```
void MainWindow::clearFilters ()  [private], [slot]
```

这是这个函数的调用关系图:

8.5.2.4 clearResults

```
void MainWindow::clearResults ()  [private], [slot]
```

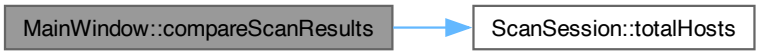
这是这个函数的调用关系图:



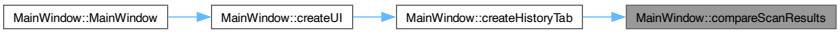
8.5.2.5 compareScanResults

```
void MainWindow::compareScanResults ()  [private], [slot]
```

函数调用图:



这是这个函数的调用关系图:



8.5.2.6 createDetailsTab()

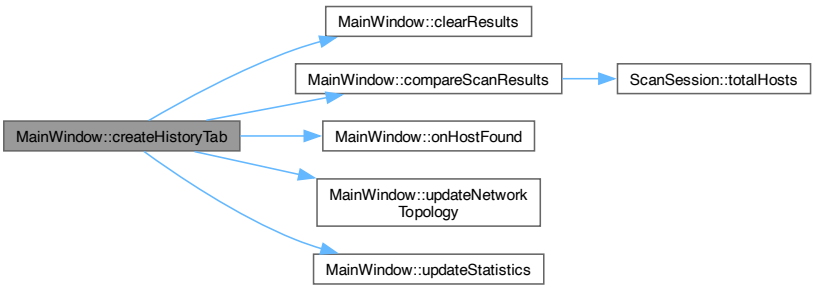
```
void MainWindow::createDetailsTab ()  [private]
```

这是这个函数的调用关系图:

8.5.2.7 createHistoryTab()

```
void MainWindow::createHistoryTab ()  [private]
```

函数调用图:

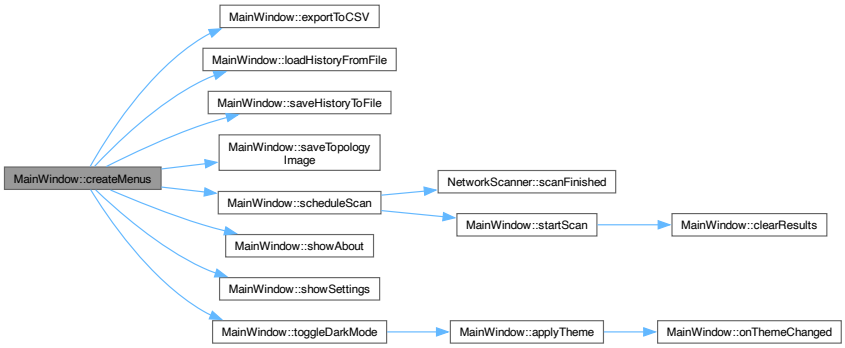


这是这个函数的调用关系图:

8.5.2.8 createMenus()

```
void MainWindow::createMenus ()  [private]
```

函数调用图:



这是这个函数的调用关系图:

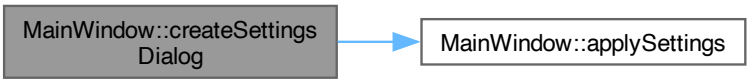
8.5.2.9 createSecurityTab()

```
void MainWindow::createSecurityTab ()  [private]
```

8.5.2.10 createSettingsDialog()

```
void MainWindow::createSettingsDialog ()  [private]
```

函数调用图:

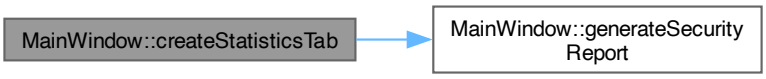


这是这个函数的调用关系图:

8.5.2.11 createStatisticsTab()

```
void MainWindow::createStatisticsTab ()  [private]
```

函数调用图:

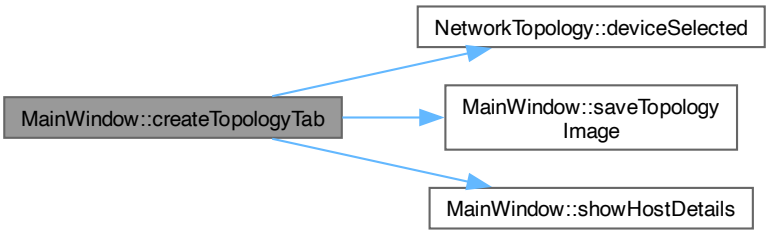


这是这个函数的调用关系图:

8.5.2.12 createTopologyTab()

```
void MainWindow::createTopologyTab ()  [private]
```

函数调用图:

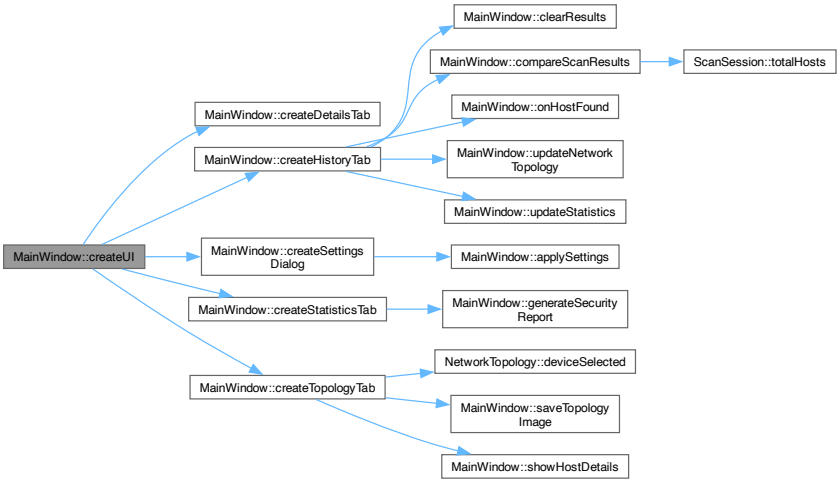


这是这个函数的调用关系图:

8.5.2.13 createUI()

```
void MainWindow::createUI () [private]
```

函数调用图:

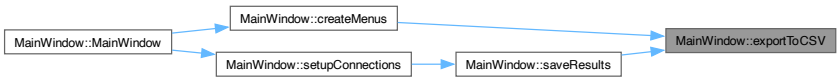


这是这个函数的调用关系图:

8.5.2.14 exportToCSV

```
void MainWindow::exportToCSV () [private], [slot]
```

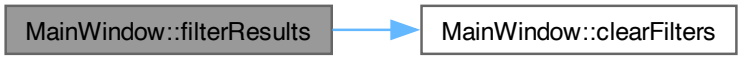
这是这个函数的调用关系图:



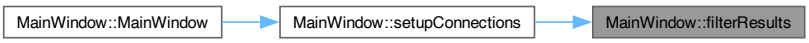
8.5.2.15 filterResults

```
void MainWindow::filterResults ()  [private], [slot]
```

函数调用图:



这是这个函数的调用关系图:



8.5.2.16 generateSecurityReport

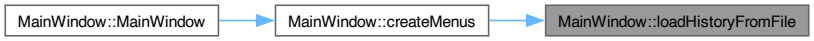
```
void MainWindow::generateSecurityReport ()  [private], [slot]
```

这是这个函数的调用关系图:

8.5.2.17 loadHistoryFromFile

```
void MainWindow::loadHistoryFromFile () [private], [slot]
```

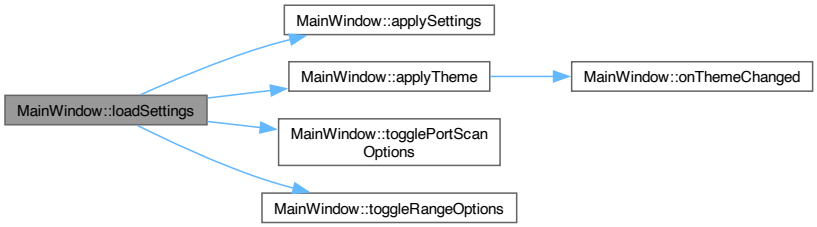
这是这个函数的调用关系图:



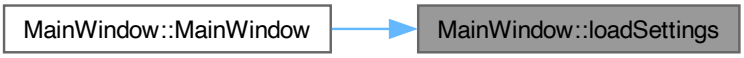
8.5.2.18 loadSettings()

```
void MainWindow::loadSettings () [private]
```

函数调用图:



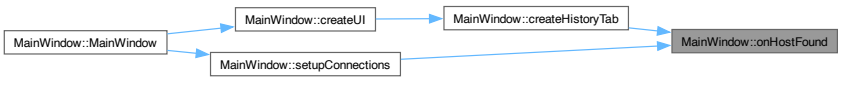
这是这个函数的调用关系图:



8.5.2.19 onHostFound

```
void MainWindow::onHostFound (
            const HostInfo & host) [private], [slot]
```

这是这个函数的调用关系图:



8.5.2.20 onScanError

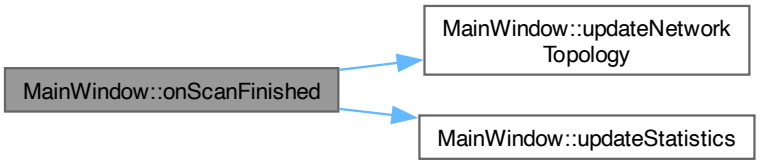
```
void MainWindow::onScanError (
            const QString & errorMessage)  [private], [slot]
```

这是这个函数的调用关系图:

8.5.2.21 onScanFinished

```
void MainWindow::onScanFinished ()  [private], [slot]
```

函数调用图:

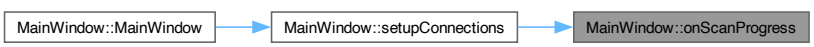


这是这个函数的调用关系图:

8.5.2.22 onScanProgress

```
void MainWindow::onScanProgress (
            int progress)  [private], [slot]
```

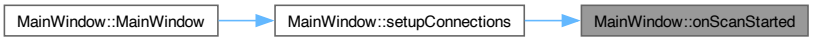
这是这个函数的调用关系图:



8.5.2.23 onScanStarted

```
void MainWindow::onScanStarted ()  [private], [slot]
```

这是这个函数的调用关系图:



8.5.2.24 onThemeChanged

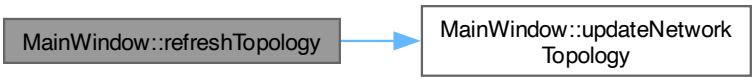
```
void MainWindow::onThemeChanged ()  [private], [slot]
```

这是这个函数的调用关系图:

8.5.2.25 refreshTopology

```
void MainWindow::refreshTopology ()  [private], [slot]
```

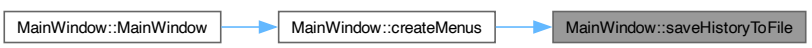
函数调用图:



8.5.2.26 saveHistoryToFile

```
void MainWindow::saveHistoryToFile ()  [private], [slot]
```

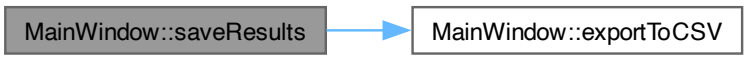
这是这个函数的调用关系图:



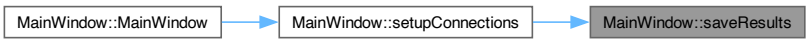
8.5.2.27 saveResults

```
void MainWindow::saveResults ()  [private], [slot]
```

函数调用图:



这是这个函数的调用关系图:



8.5.2.28 saveSettings()

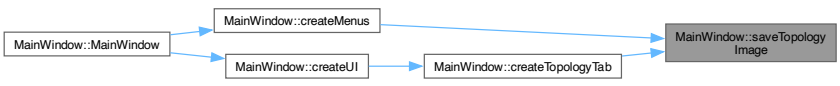
```
void MainWindow::saveSettings ()  [private]
```

这是这个函数的调用关系图:

8.5.2.29 saveTopologyImage

```
void MainWindow::saveTopologyImage ()  [private], [slot]
```

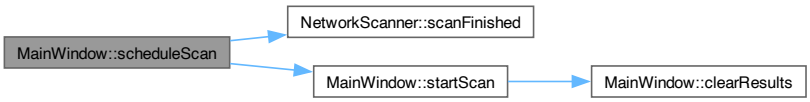
这是这个函数的调用关系图:



8.5.2.30 scheduleScan

```
void MainWindow::scheduleScan ()  [private], [slot]
```

函数调用图:

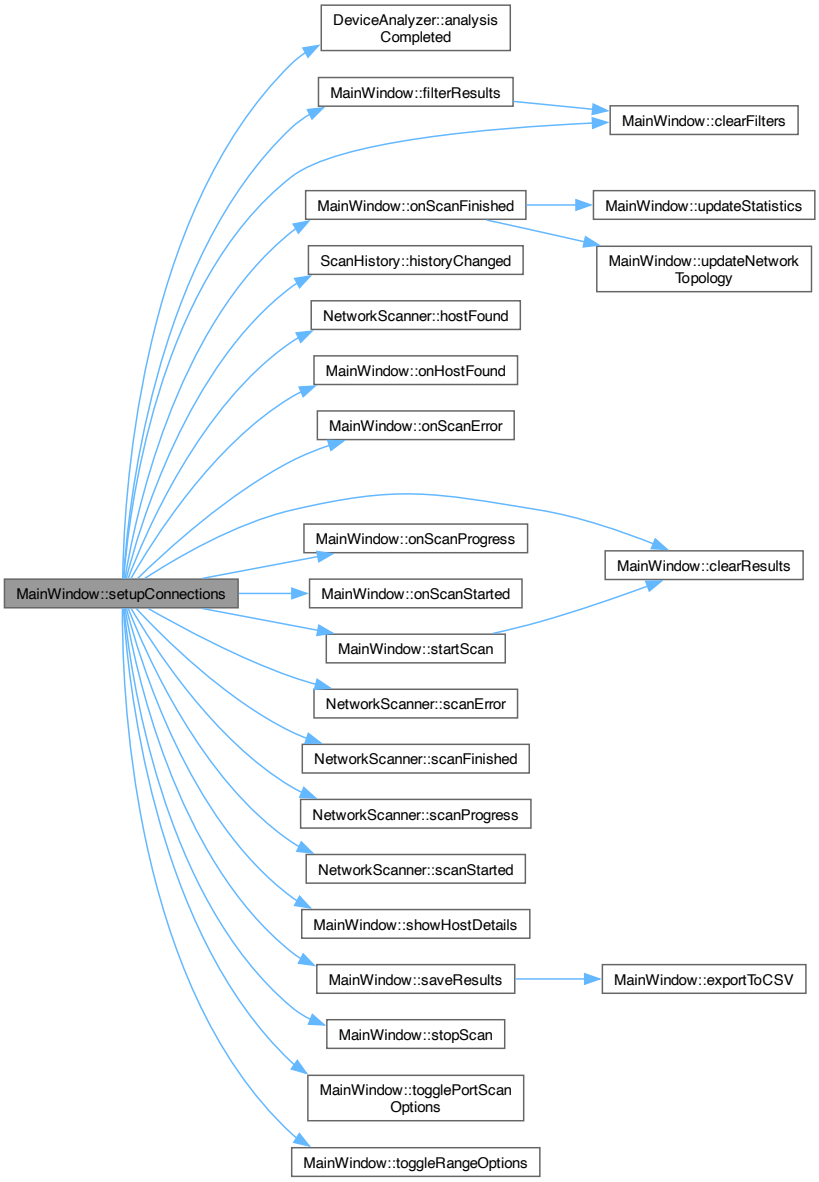


这是这个函数的调用关系图:

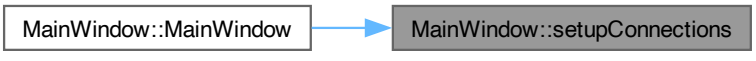
8.5.2.31 setupConnections()

```
void MainWindow::setupConnections ()  [private]
```

函数调用图:



这是这个函数的调用关系图:



8.5.2.32 showAbout

```
void MainWindow::showAbout ()  [private], [slot]
```

这是这个函数的调用关系图:

8.5.2.33 showHistoryView

```
void MainWindow::showHistoryView ()  [private], [slot]
```

8.5.2.34 showHostDetails

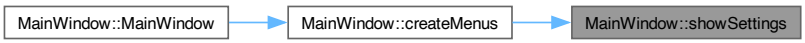
```
void MainWindow::showHostDetails (
            int row,
            int column)  [private], [slot]
```

这是这个函数的调用关系图:

8.5.2.35 showSettings

```
void MainWindow::showSettings ()  [private], [slot]
```

这是这个函数的调用关系图:



8.5.2.36 showStatisticsView

```
void MainWindow::showStatisticsView ()  [private], [slot]
```

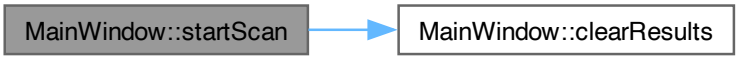
8.5.2.37 showTopologyView

```
void MainWindow::showTopologyView ()  [private], [slot]
```

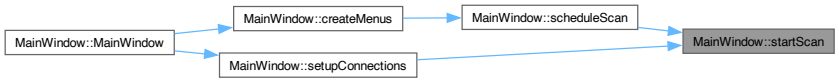
8.5.2.38 startScan

```
void MainWindow::startScan ()  [private], [slot]
```

函数调用图:



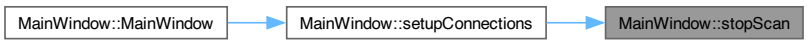
这是这个函数的调用关系图:



8.5.2.39 stopScan

```
void MainWindow::stopScan ()  [private], [slot]
```

这是这个函数的调用关系图:

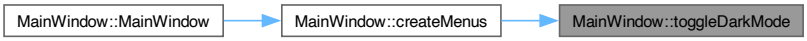


8.5.2.40 toggleDarkMode

```
void MainWindow::toggleDarkMode (
            bool enable)  [private], [slot]
```

函数调用图:

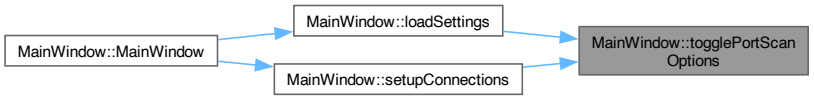
这是这个函数的调用关系图:



8.5.2.41 togglePortScanOptions

```
void MainWindow::togglePortScanOptions (
            bool checked)  [private], [slot]
```

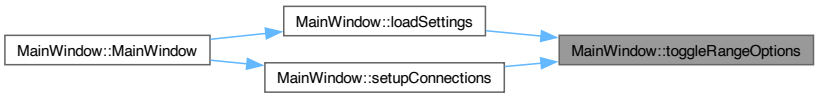
这是这个函数的调用关系图:



8.5.2.42 toggleRangeOptions

```
void MainWindow::toggleRangeOptions (
            bool checked)  [private], [slot]
```

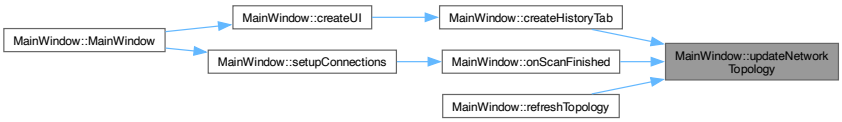
这是这个函数的调用关系图:



8.5.2.43 updateNetworkTopology

```
void MainWindow::updateNetworkTopology () [private], [slot]
```

这是这个函数的调用关系图:



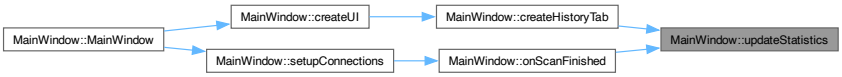
8.5.2.44 updatePortsList()

```
void MainWindow::updatePortsList () [private]
```

8.5.2.45 updateStatistics()

```
void MainWindow::updateStatistics () [private]
```

这是这个函数的调用关系图:



8.5.3 类成员变量说明

8.5.3.1 m_aboutAction

```
QAction* MainWindow::m_aboutAction [private]
```

8.5.3.2 m_centralWidget

```
QWidget* MainWindow::m_centralWidget [private]
```

8.5.3.3 m_clearButton

```
QPushButton* MainWindow::m_clearButton [private]
```

8.5.3.4 m_clearFilterButton

```
QPushButton* MainWindow::m_clearFilterButton  [private]
```

8.5.3.5 m_controlLayout

```
QHBoxLayout* MainWindow::m_controlLayout  [private]
```

8.5.3.6 m_currentHostIndex

```
int MainWindow::m_currentHostIndex  [private]
```

8.5.3.7 m_customPortsCheckBox

```
QCheckBox* MainWindow::m_customPortsCheckBox  [private]
```

8.5.3.8 m_customRangeCheckBox

```
QCheckBox* MainWindow::m_customRangeCheckBox  [private]
```

8.5.3.9 m_darkModeAction

```
QAction* MainWindow::m_darkModeAction  [private]
```

8.5.3.10 m_darkModeEnabled

```
bool MainWindow::m_darkModeEnabled  [private]
```

8.5.3.11 m_detailsLayout

```
QVBoxLayout* MainWindow::m_detailsLayout  [private]
```

8.5.3.12 m_detailsTab

```
QWidget* MainWindow::m_detailsTab  [private]
```

8.5.3.13 m_detailsTextEdit

```
QTextEdit* MainWindow::m_detailsTextEdit  [private]
```

8.5.3.14 m_deviceAnalyzer

```
DeviceAnalyzer* MainWindow::m_deviceAnalyzer  [private]
```

8.5.3.15 m_deviceTypeChartView

```
QChartView* MainWindow::m_deviceTypeChartView  [private]
```

8.5.3.16 m_endIPLineEdit

```
QLineEdit* MainWindow::m_endIPLineEdit  [private]
```

8.5.3.17 m_exitAction

```
QAction* MainWindow::m_exitAction  [private]
```

8.5.3.18 m_exportAction

```
QAction* MainWindow::m_exportAction  [private]
```

8.5.3.19 m_fileMenu

```
QMenu* MainWindow::m_fileMenu  [private]
```

8.5.3.20 m_filterButton

```
QPushButton* MainWindow::m_filterButton  [private]
```

8.5.3.21 m_filterIPLineEdit

```
QLineEdit* MainWindow::m_filterIPLineEdit  [private]
```

8.5.3.22 m_filterTypeComboBox

```
QComboBox* MainWindow::m_filterTypeComboBox  [private]
```

8.5.3.23 m_filterVendorComboBox

```
QComboBox* MainWindow::m_filterVendorComboBox  [private]
```

8.5.3.24 m_filterWidget

```
QWidget* MainWindow::m_filterWidget  [private]
```

8.5.3.25 m_helpMenu

```
QMenu* MainWindow::m_helpMenu  [private]
```

8.5.3.26 m_historyTab

```
QWidget* MainWindow::m_historyTab  [private]
```

8.5.3.27 m_historyTable

```
QTableWidget* MainWindow::m_historyTable  [private]
```

8.5.3.28 m_hostsFound

```
int MainWindow::m_hostsFound  [private]
```

8.5.3.29 m_loadHistoryAction

```
QAction* MainWindow::m_loadHistoryAction  [private]
```

8.5.3.30 m_mainLayout

```
QVBoxLayout* MainWindow::m_mainLayout  [private]
```

8.5.3.31 m_networkTopology

```
NetworkTopology* MainWindow::m_networkTopology  [private]
```

8.5.3.32 m_portDistributionChartView

```
QChartView* MainWindow::m_portDistributionChartView  [private]
```

8.5.3.33 m_portsGroupBox

```
QGroupBox* MainWindow::m_portsGroupBox  [private]
```

8.5.3.34 m_portsLineEdit

```
QLineEdit* MainWindow::m_portsLineEdit  [private]
```

8.5.3.35 m_progressBar

```
QProgressBar* MainWindow::m_progressBar  [private]
```

8.5.3.36 m_rangeGroupBox

```
QGroupBox* MainWindow::m_rangeGroupBox  [private]
```

8.5.3.37 m_resultsTable

```
QTableWidget* MainWindow::m_resultsTable  [private]
```

8.5.3.38 m_saveButton

```
QPushButton* MainWindow::m_saveButton  [private]
```

8.5.3.39 m_saveHistoryAction

```
QAction* MainWindow::m_saveHistoryAction  [private]
```

8.5.3.40 m_saveTopologyAction

```
QAction* MainWindow::m_saveTopologyAction  [private]
```

8.5.3.41 m_scanButton

```
QPushButton* MainWindow::m_scanButton  [private]
```

8.5.3.42 m_scanHistory

```
ScanHistory* MainWindow::m_scanHistory  [private]
```

8.5.3.43 m_scanner

```
NetworkScanner* MainWindow::m_scanner  [private]
```

8.5.3.44 m_scanTab

```
QWidget* MainWindow::m_scanTab  [private]
```

8.5.3.45 m_scheduleScanAction

```
QAction* MainWindow::m_scheduleScanAction  [private]
```

8.5.3.46 m_securityReportText

```
QTextEdit* MainWindow::m_securityReportText  [private]
```

8.5.3.47 m_sessionComboBox

```
QComboBox* MainWindow::m_sessionComboBox  [private]
```

8.5.3.48 m_settingsAction

```
QAction* MainWindow::m_settingsAction  [private]
```

8.5.3.49 m_settingsLayout

```
QVBoxLayout* MainWindow::m_settingsLayout  [private]
```

8.5.3.50 m_settingsTab

```
QWidget* MainWindow::m_settingsTab  [private]
```

8.5.3.51 m_startIPLineEdit

```
QLineEdit* MainWindow::m_startIPLineEdit  [private]
```

8.5.3.52 m_statisticsTab

```
QWidget* MainWindow::m_statisticsTab  [private]
```

8.5.3.53 m_statusBar

```
QStatusBar* MainWindow::m_statusBar  [private]
```

8.5.3.54 m_statusLabel

```
QLabel* MainWindow::m_statusLabel  [private]
```

8.5.3.55 m_stopButton

```
QPushButton* MainWindow::m_stopButton  [private]
```

8.5.3.56 m_tabWidget

```
QTabWidget* MainWindow::m_tabWidget  [private]
```

8.5.3.57 m_timeoutSpinBox

```
QSpinBox* MainWindow::m_timeoutSpinBox  [private]
```

8.5.3.58 m_toolsMenu

```
QMenu* MainWindow::m_toolsMenu  [private]
```

8.5.3.59 m_topologyTab

```
QWidget* MainWindow::m_topologyTab  [private]
```

8.5.3.60 m_vendorChartView

```
QChartView* MainWindow::m_vendorChartView  [private]
```

8.5.3.61 m_viewMenu

```
QMenu* MainWindow::m_viewMenu  [private]
```

该类的文档由以下文件生成:

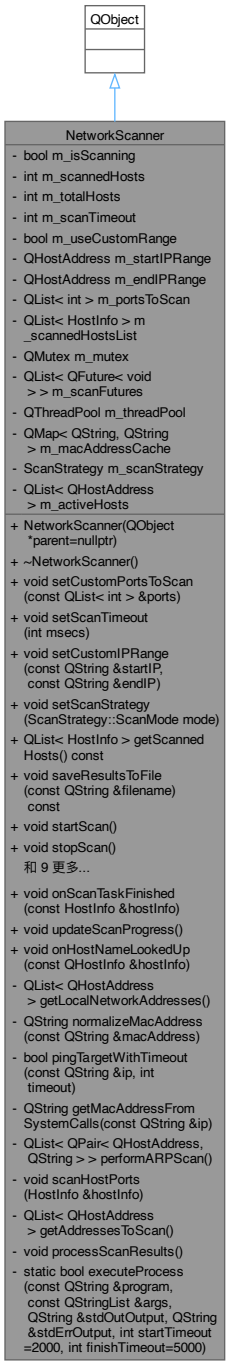
- mainwindow.h
- mainwindow.cpp

8.6 NetworkScanner类 参考

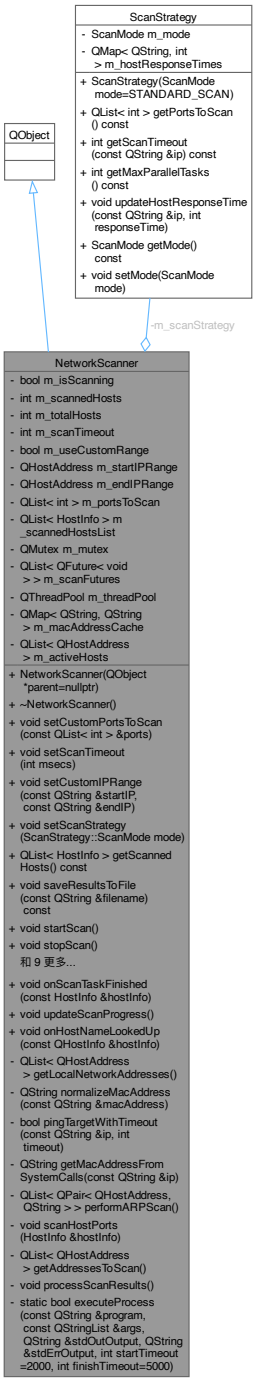
网络扫描器类

`#include <networkscanner.h>`

类 NetworkScanner 继承关系图:



NetworkScanner 的协作图:



Public 槽

- void onScanTaskFinished (const HostInfo &hostInfo)

  处理扫描任务完成
- void updateScanProgress ()

  更新扫描进度
- void onHostNameLookedUp (const QHostInfo &hostInfo)

  主机名查询完成处理

信号

- void hostFound (const HostInfo &host)
  找到主机信号
- void scanStarted ()
  扫描开始信号
- void scanFinished ()
  扫描完成信号
- void scanProgress (int progress)
  扫描进度更新信号
- void scanError (const QString &errorMessage)
  扫描错误信号

Public 成员函数

- NetworkScanner (QObject *parent=nullptr)
  构造函数
- ~NetworkScanner ()
  析构函数
- void setCustomPortsToScan (const QList< int > &ports)
  设置自定义端口扫描列表
- void setScanTimeout (int msecs)
  设置扫描超时时间
- void setCustomIPRange (const QString &startIP, const QString &endIP)
  设置自定义*IP*范围
- void setScanStrategy (ScanStrategy::ScanMode mode)
  设置扫描策略
- QList< HostInfo > getScannedHosts () const
  获取扫描结果
- void saveResultsToFile (const QString &filename) const
  保存结果到文件
- void startScan ()
  开始扫描
- void stopScan ()
  停止扫描
- bool isScanning () const
  检查是否正在扫描
- QList< QHostAddress > quickPingScan (const QList< QHostAddress > &addresses)
  快速*Ping*扫描方法
- bool isHostReachable (const QHostAddress &address, int timeout)
  检查主机是否可达
- bool isReachableOnPorts (const QHostAddress &address, const QList< int > &ports, int timeout)
  检查主机在多个端口上是否可达
- void scanHost (const QHostAddress &address)
  扫描单个主机
- QString lookupHostName (const QHostAddress &address)
  查询主机名
- QString lookupMacAddress (const QHostAddress &address)
  查询*MAC*地址
- QString lookupMacVendor (const QString &macAddress)
  查询*MAC*地址对应的厂商
- QString generatePseudoMACFromIP (const QString &ip)
  根据*IP*地址生成伪*MAC*地址

Private 成员函数

- QList< QHostAddress > getLocalNetworkAddresses ()
  获取本地网络地址列表
- QString normalizeMacAddress (const QString &macAddress)
  *MAC*地址规范化
- bool pingTargetWithTimeout (const QString &ip, int timeout)
  使用*ping*命令检测目标是否可达
- QString getMacAddressFromSystemCalls (const QString &ip)
  通过系统调用获取*MAC*地址
- QList< QPair< QHostAddress, QString > > performARPScan ()
  执行*ARP*扫描
- void scanHostPorts (HostInfo &hostInfo)
  扫描主机端口
- QList< QHostAddress > getAddressesToScan ()
  获取要扫描的地址列表
- void processScanResults ()
  处理扫描结果

静态 **Private** 成员函数

- static bool executeProcess (const QString &program, const QStringList &args, QString &stdOutOutput, QString &stdErrOutput, int startTimeout=2000, int finishTimeout=5000)
  执行外部进程

Private 属性

- bool m_isScanning
  是否正在扫描
- int m_scannedHosts
  已扫描主机数
- int m_totalHosts
  总主机数
- int m_scanTimeout
  扫描超时时间
- bool m_useCustomRange
  是否使用自定义*IP*范围
- QHostAddress m_startIPRange
  起始*IP*地址
- QHostAddress m_endIPRange
  结束*IP*地址
- QList< int > m_portsToScan
  要扫描的端口列表
- QList< HostInfo > m_scannedHostsList
  扫描结果列表
- QMutex m_mutex
  线程同步互斥锁
- QList< QFuture< void > > m_scanFutures
  并行扫描任务
- QThreadPool m_threadPool

线程池

- QMap< QString, QString > m_macAddressCache
  *MAC*地址缓存
- ScanStrategy m_scanStrategy
  扫描策略
- QList< QHostAddress > m_activeHosts
  活跃主机列表

8.6.1 详细描述

网络扫描器类

提供网络设备发现和端口扫描的核心功能

8.6.2 构造及析构函数说明

8.6.2.1 NetworkScanner()

```
NetworkScanner::NetworkScanner (
            QObject * parent = nullptr)
```

构造函数

NetworkScanner类构造函数

参数

parent	父对象
parent	父对象

初始化扫描器参数和线程池

8.6.2.2 ~NetworkScanner()

```
NetworkScanner::~NetworkScanner ()
```

析构函数

NetworkScanner类析构函数

停止扫描并清理资源 函数调用图:

8.6.3 成员函数说明

8.6.3.1 executeProcess()

```
bool NetworkScanner::executeProcess (
            const QString & program,
            const QStringList & args,
            QString & stdOutOutput,
            QString & stdErrOutput,
            int startTimeout = 2000,
            int finishTimeout = 5000)  [static], [private]
```

执行外部进程

参数

program	程序路径
args	参数列表
stdOutOutput	标准输出结果
stdErrOutput	标准错误结果
startTimeout	启动超时(毫秒)
finishTimeout	完成超时(毫秒)

返回

是否执行成功

8.6.3.2 generatePseudoMACFromIP()

```
QString NetworkScanner::generatePseudoMACFromIP (
            const QString & ip)
```

根据IP地址生成伪MAC地址

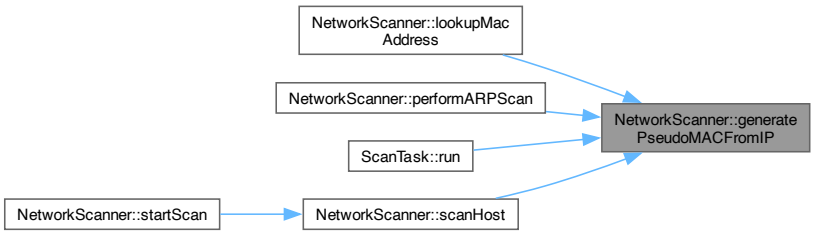
参数

ip	IP地址

返回

生成的伪MAC地址

这是这个函数的调用关系图:



8.6.3.3 getAddressesToScan()

```
QList< QHostAddress > NetworkScanner::getAddressesToScan ()  [private]
```

获取要扫描的地址列表

返回

地址列表

8.6.3.4 getLocalNetworkAddresses()

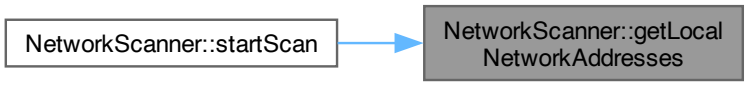
```
QList< QHostAddress > NetworkScanner::getLocalNetworkAddresses ()  [private]
```

获取本地网络地址列表

返回

本地网络地址列表

这是这个函数的调用关系图:



8.6.3.5 getMacAddressFromSystemCalls()

```
QString NetworkScanner::getMacAddressFromSystemCalls (
            const QString & ip)  [private]
```

通过系统调用获取MAC地址

参数

ip	目标IP地址

返回

MAC地址

8.6.3.6 getScannedHosts()

```
QList< HostInfo > NetworkScanner::getScannedHosts () const
```

获取扫描结果

返回

扫描到的主机信息列表

8.6.3.7 hostFound

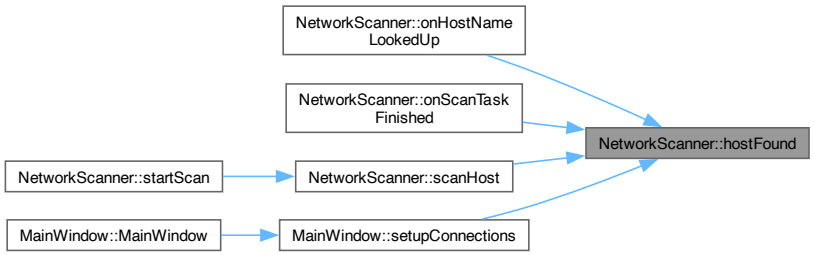
```
void NetworkScanner::hostFound (
            const HostInfo & host)  [signal]
```

找到主机信号

参数

host	主机信息

这是这个函数的调用关系图:



8.6.3.8 isHostReachable()

```
bool NetworkScanner::isHostReachable (
            const QHostAddress & address,
            int timeout)
```

检查主机是否可达

参数

address	主机地址
timeout	超时时间(毫秒)

返回

主机是否可达

8.6.3.9 isReachableOnPorts()

```
bool NetworkScanner::isReachableOnPorts (
            const QHostAddress & address,
            const QList< int > & ports,
            int timeout)
```

检查主机在多个端口上是否可达

参数

address	主机地址
ports	端口列表
timeout	超时时间(毫秒)

返回

是否至少有一个端口可达

8.6.3.10 isScanning()

```
bool NetworkScanner::isScanning () const
```

检查是否正在扫描

返回

是否正在扫描

8.6.3.11 lookupHostName()

```
QString NetworkScanner::lookupHostName (
            const QHostAddress & address)
```

查询主机名

参数

address	主机地址

返回

主机名

这是这个函数的调用关系图:

8.6.3.12 lookupMacAddress()

```
QString NetworkScanner::lookupMacAddress (
            const QHostAddress & address)
```

查询MAC地址

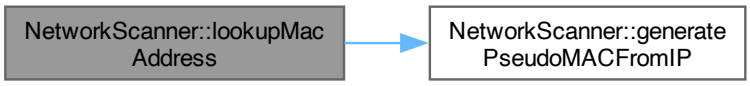
参数

address	主机地址

返回

MAC地址

函数调用图:



8.6.3.13 lookupMacVendor()

```
QString NetworkScanner::lookupMacVendor (
            const QString & macAddress)
```

查询MAC地址对应的厂商

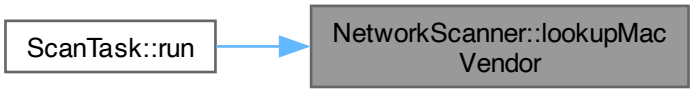
参数

macAddress	MAC地址

返回

厂商名称

这是这个函数的调用关系图:



8.6.3.14 normalizeMacAddress()

```
QString NetworkScanner::normalizeMacAddress (
            const QString & macAddress)  [private]
```

MAC地址规范化

参数

macAddress	原始MAC地址

返回

规范化的MAC地址

这是这个函数的调用关系图:

8.6.3.15 onHostNameLookedUp

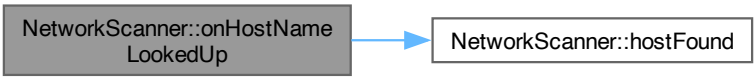
```
void NetworkScanner::onHostNameLookedUp (
            const QHostInfo & hostInfo)  [slot]
```

主机名查询完成处理

参数

hostInfo	主机信息

函数调用图:



8.6.3.16 onScanTaskFinished

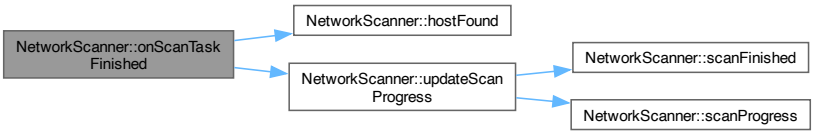
```
void NetworkScanner::onScanTaskFinished (
            const HostInfo & hostInfo)  [slot]
```

处理扫描任务完成

参数

hostInfo	扫描到的主机信息

函数调用图:



8.6.3.17 performARPScan()

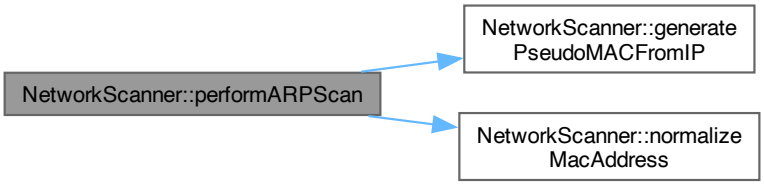
```
QList< QPair< QHostAddress, QString > > NetworkScanner::performARPScan ()  [private]
```

执行ARP扫描

返回

地址和MAC地址对的列表

函数调用图:



8.6.3.18 pingTargetWithTimeout()

```
bool NetworkScanner::pingTargetWithTimeout (
            const QString & ip,
            int timeout)  [private]
```

使用ping命令检测目标是否可达

参数

ip	目标IP地址
timeout	超时时间(毫秒)

返回

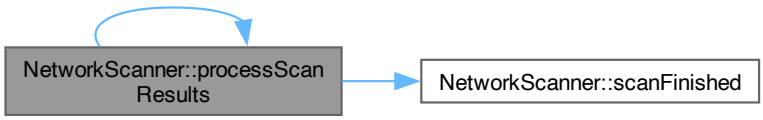
是否可达

8.6.3.19 processScanResults()

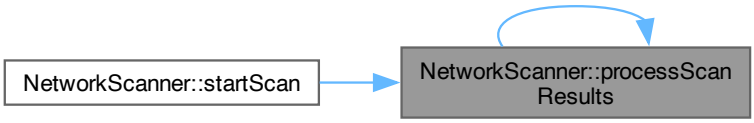
```
void NetworkScanner::processScanResults ()  [private]
```

处理扫描结果

函数调用图:



这是这个函数的调用关系图:



8.6.3.20 quickPingScan()

```
QList< QHostAddress > NetworkScanner::quickPingScan (
            const QList< QHostAddress > & addresses)
```

快速Ping扫描方法

参数

addresses	要扫描的地址列表

返回

活跃的主机地址列表

8.6.3.21 saveResultsToFile()

```
void NetworkScanner::saveResultsToFile (
            const QString & filename) const
```

保存结果到文件

参数

filename	文件名

8.6.3.22 scanError

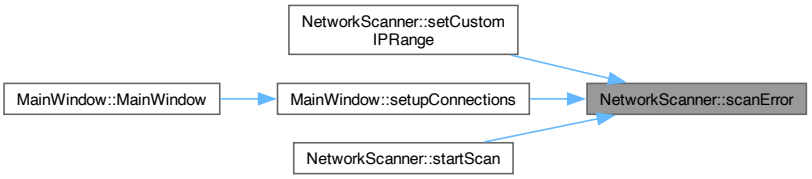
```
void NetworkScanner::scanError (
            const QString & errorMessage)  [signal]
```

扫描错误信号

参数

errorMessage	错误信息

这是这个函数的调用关系图:

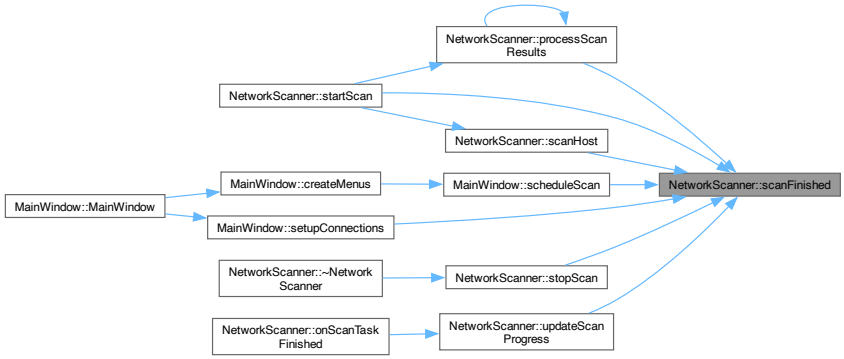


8.6.3.23 scanFinished

```
void NetworkScanner::scanFinished ()  [signal]
```

扫描完成信号

这是这个函数的调用关系图:



8.6.3.24 scanHost()

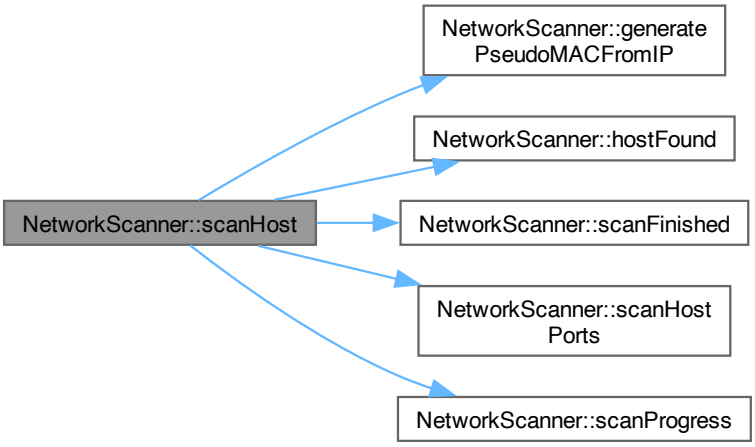
```
void NetworkScanner::scanHost (
            const QHostAddress & address)
```

扫描单个主机

参数

address	主机地址

函数调用图:



这是这个函数的调用关系图:

8.6.3.25 scanHostPorts()

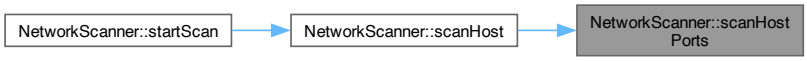
```
void NetworkScanner::scanHostPorts (
            HostInfo & hostInfo)  [private]
```

扫描主机端口

参数

hostInfo	要更新的主机信息结构

这是这个函数的调用关系图:



8.6.3.26 scanProgress

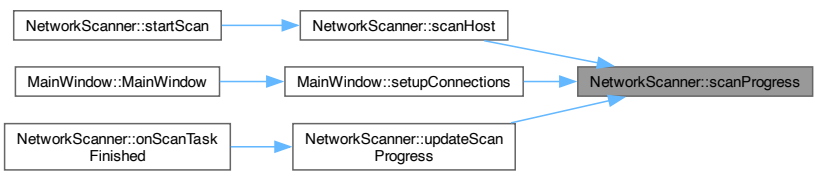
```
void NetworkScanner::scanProgress (
            int progress)  [signal]
```

扫描进度更新信号

参数

progress	进度百分比(0-100)

这是这个函数的调用关系图:

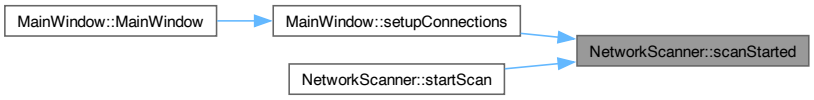


8.6.3.27 scanStarted

```
void NetworkScanner::scanStarted ()  [signal]
```

扫描开始信号

这是这个函数的调用关系图:



8.6.3.28 setCustomIPRange()

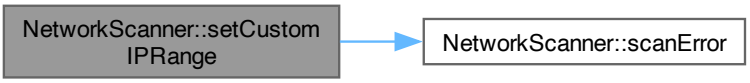
```
void NetworkScanner::setCustomIPRange (
            const QString & startIP,
            const QString & endIP)
```

设置自定义IP范围

参数

start↩ IP	起始IP地址
endIP	结束IP地址

函数调用图:



8.6.3.29 setCustomPortsToScan()

```
void NetworkScanner::setCustomPortsToScan (
            const QList< int > & ports)
```

设置自定义端口扫描列表

参数

ports	要扫描的端口列表

8.6.3.30 setScanStrategy()

```
void NetworkScanner::setScanStrategy (
            ScanStrategy::ScanMode mode)
```

设置扫描策略

参数

mode	扫描模式

8.6.3.31 setScanTimeout()

```
void NetworkScanner::setScanTimeout (
            int msecs)
```

设置扫描超时时间

参数

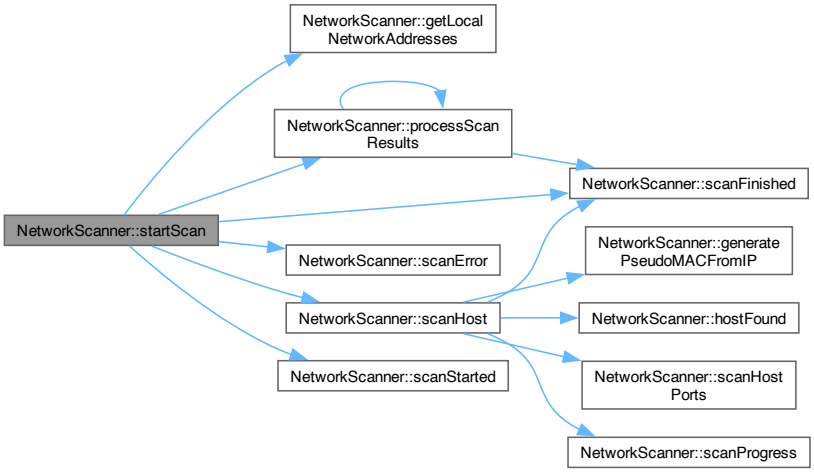
msecs	超时时间(毫秒)
msecs	超时时间（毫秒）

8.6.3.32 startScan()

```
void NetworkScanner::startScan ()
```

开始扫描

函数调用图:

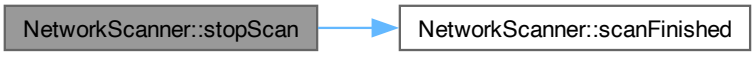


8.6.3.33 stopScan()

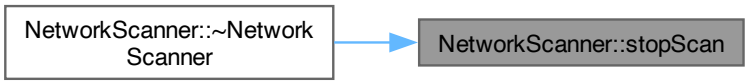
```
void NetworkScanner::stopScan ()
```

停止扫描

函数调用图:



这是这个函数的调用关系图:

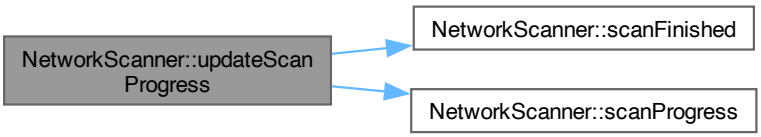


8.6.3.34 updateScanProgress

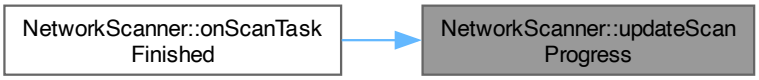
```
void NetworkScanner::updateScanProgress ()  [slot]
```

更新扫描进度

函数调用图:



这是这个函数的调用关系图:



8.6.4 类成员变量说明

8.6.4.1 m_activeHosts

```
QList<QHostAddress> NetworkScanner::m_activeHosts  [private]
```

活跃主机列表

8.6.4.2 m_endIPRange

```
QHostAddress NetworkScanner::m_endIPRange  [private]
```

结束IP地址

8.6.4.3 m_isScanning

```
bool NetworkScanner::m_isScanning  [private]
```

是否正在扫描

8.6.4.4 m_macAddressCache

```
QMap<QString, QString> NetworkScanner::m_macAddressCache  [private]
```

MAC地址缓存

8.6.4.5 m_mutex

```
QMutex NetworkScanner::m_mutex  [private]
```

线程同步互斥锁

8.6.4.6 m_portsToScan

```
QList<int> NetworkScanner::m_portsToScan  [private]
```

要扫描的端口列表

8.6.4.7 m_scanFutures

```
QList<QFuture<void> > NetworkScanner::m_scanFutures  [private]
```

并行扫描任务

8.6.4.8 m_scannedHosts

```
int NetworkScanner::m_scannedHosts  [private]
```

已扫描主机数

8.6.4.9 m_scannedHostsList

```
QList<HostInfo> NetworkScanner::m_scannedHostsList  [private]
```

扫描结果列表

8.6.4.10 m_scanStrategy

```
ScanStrategy NetworkScanner::m_scanStrategy  [private]
```

扫描策略

8.6.4.11 m_scanTimeout

```
int NetworkScanner::m_scanTimeout  [private]
```

扫描超时时间

8.6.4.12 m_startIPRange

```
QHostAddress NetworkScanner::m_startIPRange  [private]
```

起始IP地址

8.6.4.13 m_threadPool

```
QThreadPool NetworkScanner::m_threadPool  [private]
```

线程池

8.6.4.14 m_totalHosts

```
int NetworkScanner::m_totalHosts  [private]
```

总主机数

8.6.4.15 m_useCustomRange

```
bool NetworkScanner::m_useCustomRange  [private]
```

是否使用自定义IP范围

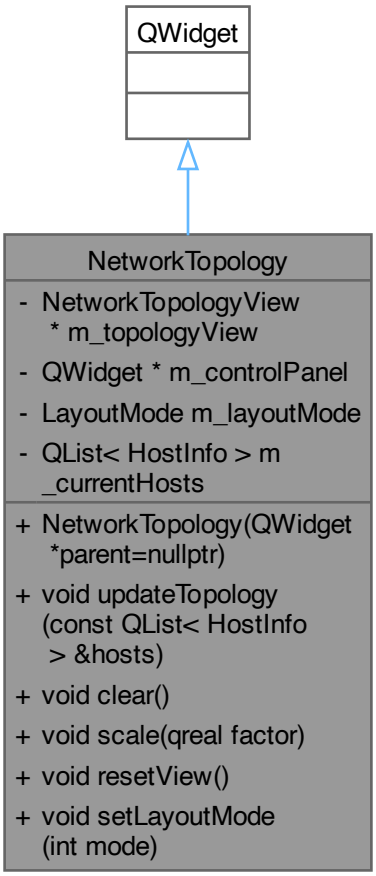
该类的文档由以下文件生成:

- networkscanner.h
- build/NetScanner_autogen/EWIEGA46WW/moc_networkscanner.cpp
- NetScanner_autogen/JRIAJ772TK/moc_networkscanner.cpp
- networkscanner.cpp

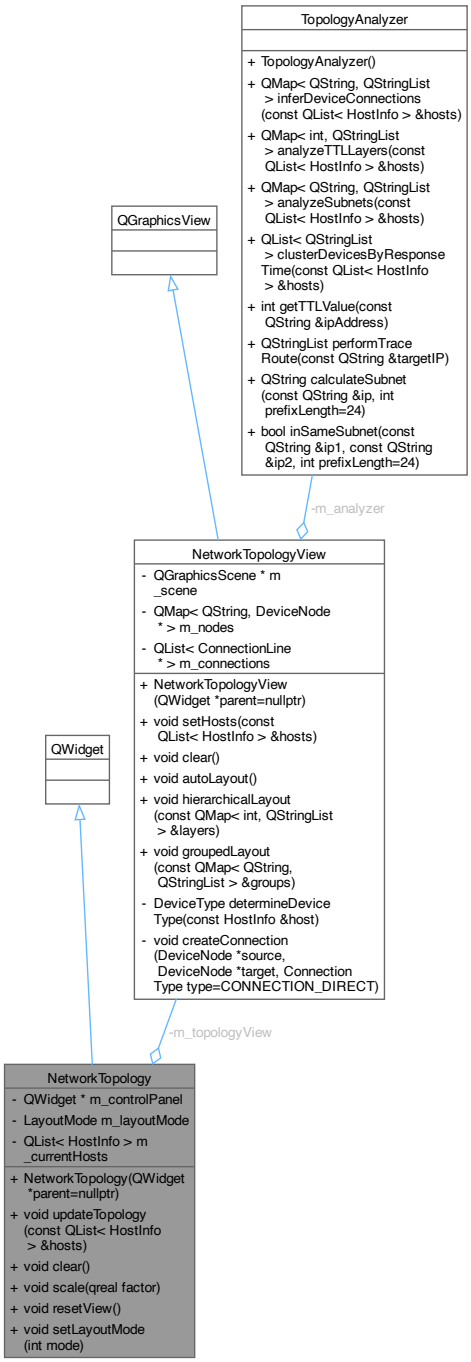
8.7 NetworkTopology类 参考

```
#include <networktopology.h>
```

类 NetworkTopology 继承关系图:



NetworkTopology 的协作图:



信号

- void deviceSelected (const HostInfo &host)

Public 成员函数

- NetworkTopology (QWidget *parent=nullptr)

- void updateTopology (const QList< HostInfo > &hosts)
- void clear ()
- void scale (qreal factor)
- void resetView ()
- void setLayoutMode (int mode)

Private 类型

- enum LayoutMode { LAYOUT_AUTO , LAYOUT_HIERARCHICAL , LAYOUT_GROUPED }

Private 属性

- NetworkTopologyView * m_topologyView
- QWidget * m_controlPanel
- LayoutMode m_layoutMode
- QList< HostInfo > m_currentHosts

8.7.1 成员枚举类型说明

8.7.1.1 LayoutMode

```
enum NetworkTopology::LayoutMode  [private]
```

枚举值

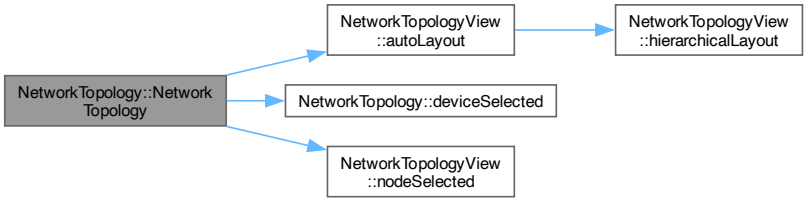
LAYOUT_AUTO	
LAYOUT_HIERARCHICAL	
LAYOUT_GROUPED	

8.7.2 构造及析构函数说明

8.7.2.1 NetworkTopology()

```
NetworkTopology::NetworkTopology (
            QWidget * parent = nullptr)
```

函数调用图:



8.7.3 成员函数说明

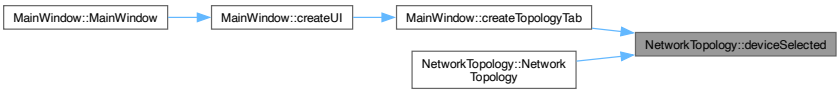
8.7.3.1 clear()

```
void NetworkTopology::clear ()
```

8.7.3.2 deviceSelected

```
void NetworkTopology::deviceSelected (
            const HostInfo & host)  [signal]
```

这是这个函数的调用关系图:



8.7.3.3 resetView()

```
void NetworkTopology::resetView ()
```

8.7.3.4 scale()

```
void NetworkTopology::scale (
            qreal factor)
```

8.7.3.5 setLayoutMode()

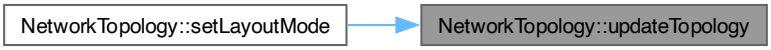
```
void NetworkTopology::setLayoutMode (
            int mode)
```

函数调用图:

8.7.3.6 updateTopology()

```
void NetworkTopology::updateTopology (
            const QList< HostInfo > & hosts)
```

这是这个函数的调用关系图:



8.7.4 类成员变量说明

8.7.4.1 m_controlPanel

```
QWidget* NetworkTopology::m_controlPanel  [private]
```

8.7.4.2 m_currentHosts

```
QList<HostInfo> NetworkTopology::m_currentHosts  [private]
```

8.7.4.3 m_layoutMode

```
LayoutMode NetworkTopology::m_layoutMode  [private]
```

8.7.4.4 m_topologyView

```
NetworkTopologyView* NetworkTopology::m_topologyView  [private]
```

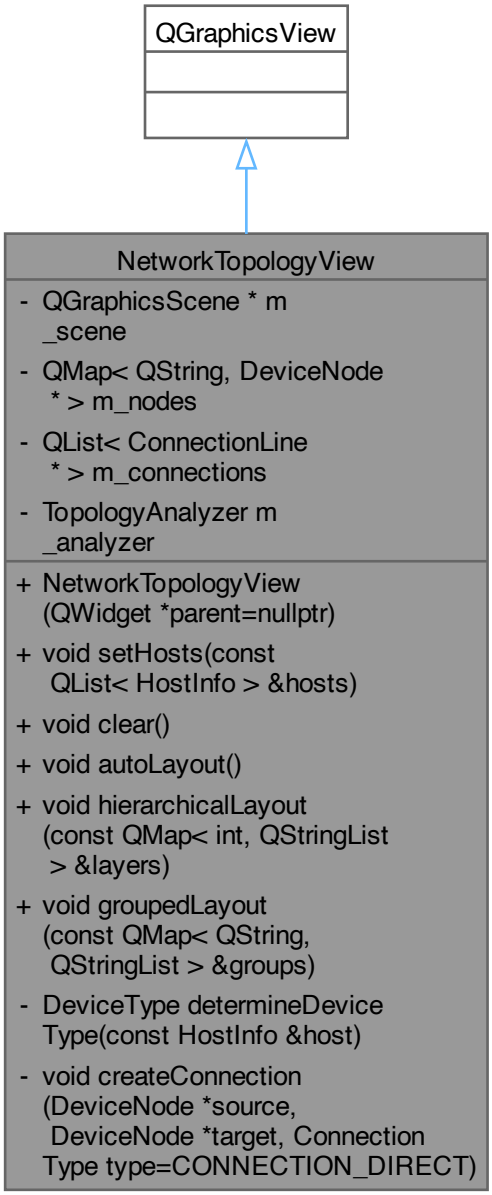
该类的文档由以下文件生成:

- networktopology.h
- build/NetScanner_autogen/EWIEGA46WW/moc_networktopology.cpp
- NetScanner_autogen/JRIAJ772TK/moc_networktopology.cpp
- networktopology.cpp

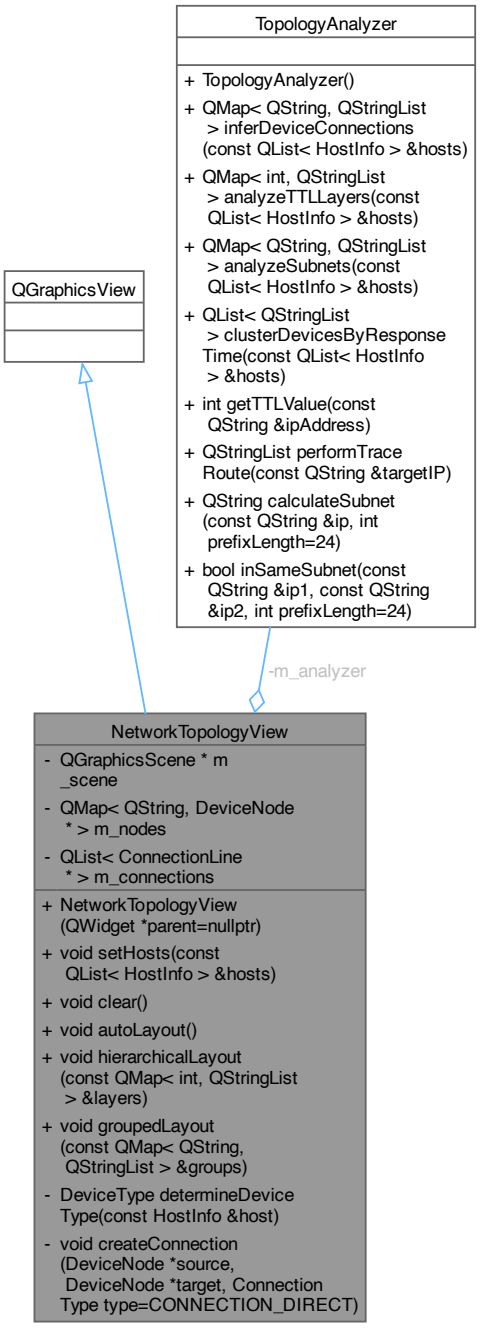
8.8 NetworkTopologyView类 参考

```
#include <networktopology.h>
```

类 NetworkTopologyView 继承关系图:



NetworkTopologyView 的协作图:



信号

- void nodeSelected (const HostInfo &host)

Public 成员函数

- NetworkTopologyView (QWidget ∗parent=nullptr)

- void setHosts (const QList< HostInfo > &hosts)
- void clear ()
- void autoLayout ()
- void hierarchicalLayout (const QMap< int, QStringList > &layers)
- void groupedLayout (const QMap< QString, QStringList > &groups)

Private 成员函数

- DeviceType determineDeviceType (const HostInfo &host)
- void createConnection (DeviceNode *source, DeviceNode *target, ConnectionType type=CONNECTION_DIRECT)

Private 属性

- QGraphicsScene * m_scene
- QMap< QString, DeviceNode * > m_nodes
- QList< ConnectionLine * > m_connections
- TopologyAnalyzer m_analyzer

8.8.1 构造及析构函数说明

8.8.1.1 NetworkTopologyView()

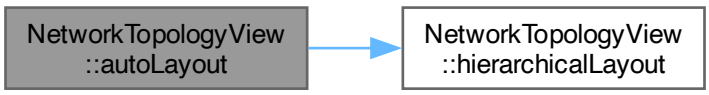
```
NetworkTopologyView::NetworkTopologyView (
            QWidget * parent = nullptr)
```

8.8.2 成员函数说明

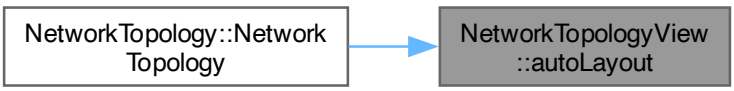
8.8.2.1 autoLayout()

```
void NetworkTopologyView::autoLayout ()
```

函数调用图:



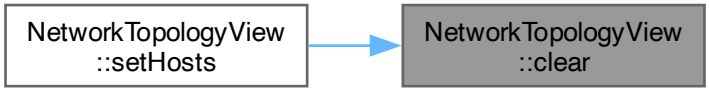
这是这个函数的调用关系图:



8.8.2.2 clear()

```
void NetworkTopologyView::clear ()
```

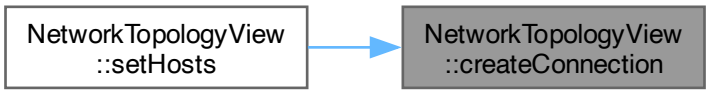
这是这个函数的调用关系图:



8.8.2.3 createConnection()

```
void NetworkTopologyView::createConnection (
            DeviceNode * source,
            DeviceNode * target,
            ConnectionType type = CONNECTION_DIRECT)  [private]
```

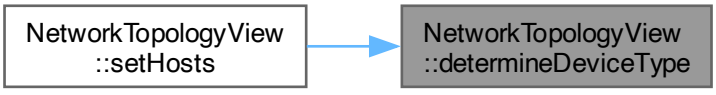
这是这个函数的调用关系图:



8.8.2.4 determineDeviceType()

```
DeviceType NetworkTopologyView::determineDeviceType (
            const HostInfo & host)  [private]
```

这是这个函数的调用关系图:



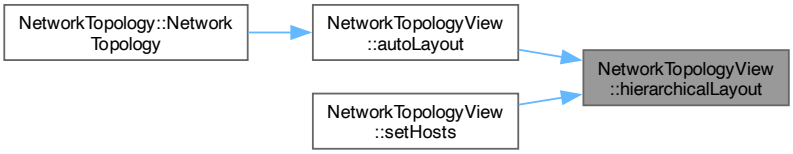
8.8.2.5 groupedLayout()

```
void NetworkTopologyView::groupedLayout (
            const QMap< QString, QStringList > & groups)
```

8.8.2.6 hierarchicalLayout()

```
void NetworkTopologyView::hierarchicalLayout (
            const QMap< int, QStringList > & layers)
```

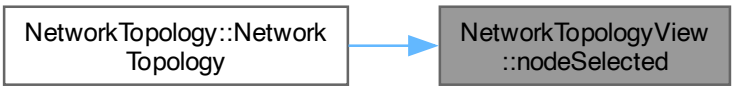
这是这个函数的调用关系图:



8.8.2.7 nodeSelected

```
void NetworkTopologyView::nodeSelected (
            const HostInfo & host)  [signal]
```

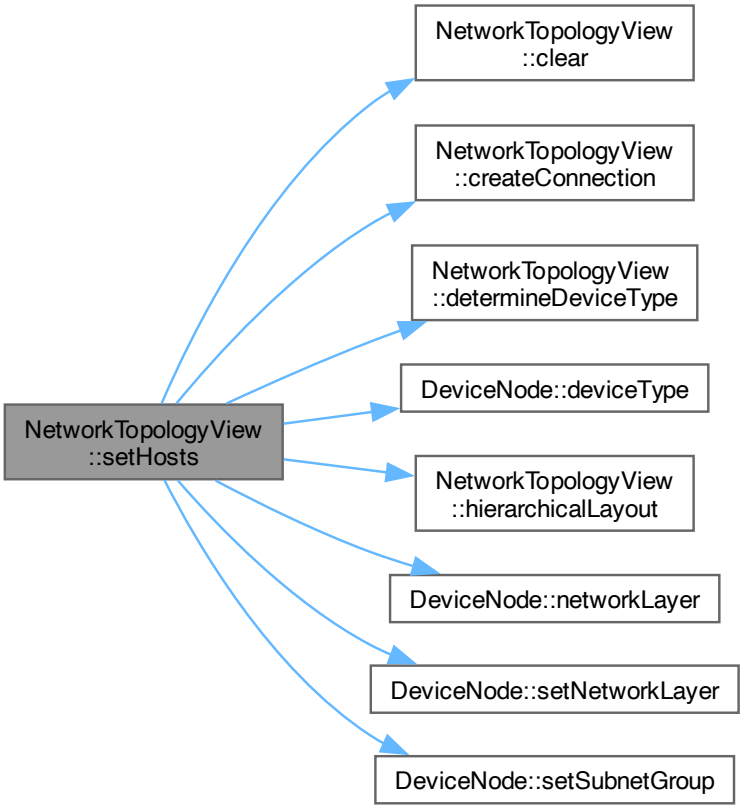
这是这个函数的调用关系图:



8.8.2.8 setHosts()

```
void NetworkTopologyView::setHosts (
            const QList< HostInfo > & hosts)
```

函数调用图:



8.8.3 类成员变量说明

8.8.3.1 m_analyzer

```
TopologyAnalyzer NetworkTopologyView::m_analyzer  [private]
```

8.8.3.2 m_connections

```
QList<ConnectionLine*> NetworkTopologyView::m_connections  [private]
```

8.8.3.3 m_nodes

```
QMap<QString, DeviceNode*> NetworkTopologyView::m_nodes  [private]
```

8.8.3.4 m_scene

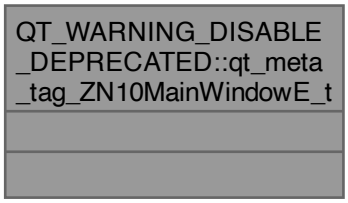
```
QGraphicsScene* NetworkTopologyView::m_scene  [private]
```

该类的文档由以下文件生成:

- networktopology.h
- build/NetScanner_autogen/EWIEGA46WW/moc_networktopology.cpp
- NetScanner_autogen/JRIAJ772TK/moc_networktopology.cpp
- networktopology.cpp

8.9 QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN10MainWindowE_t结构体 参考

QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN10MainWindowE_t 的协作图:

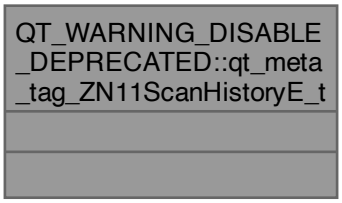


该结构体的文档由以下文件生成:

- build/NetScanner_autogen/EWIEGA46WW/moc_mainwindow.cpp

8.10 QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN11ScanHistoryE_t结构体 参考

QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN11ScanHistoryE_t 的协作图:

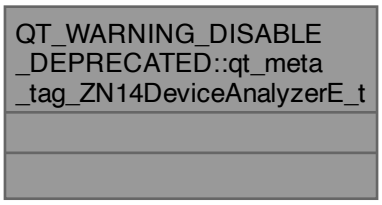


该结构体的文档由以下文件生成:

- build/NetScanner_autogen/EWIEGA46WW/moc_scanhistory.cpp

8.11 QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN14DeviceAnalyzerE_t结构体 参考

QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN14DeviceAnalyzerE_t 的协作图:

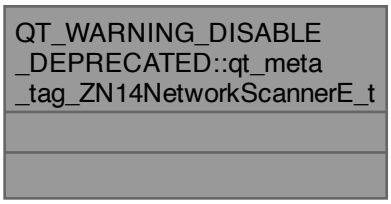


该结构体的文档由以下文件生成:

- build/NetScanner_autogen/EWIEGA46WW/moc_deviceanalyzer.cpp

8.12 QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN14NetworkScannerE_t结构体 参考

QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN14NetworkScannerE_t 的协作图:

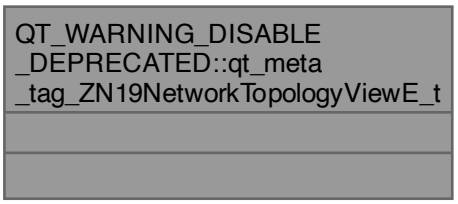


该结构体的文档由以下文件生成:

- build/NetScanner_autogen/EWIEGA46WW/moc_networkscanner.cpp

8.13 QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN19NetworkTopologyViewE_t结构体 参考

QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN19NetworkTopologyViewE_t 的协作图:



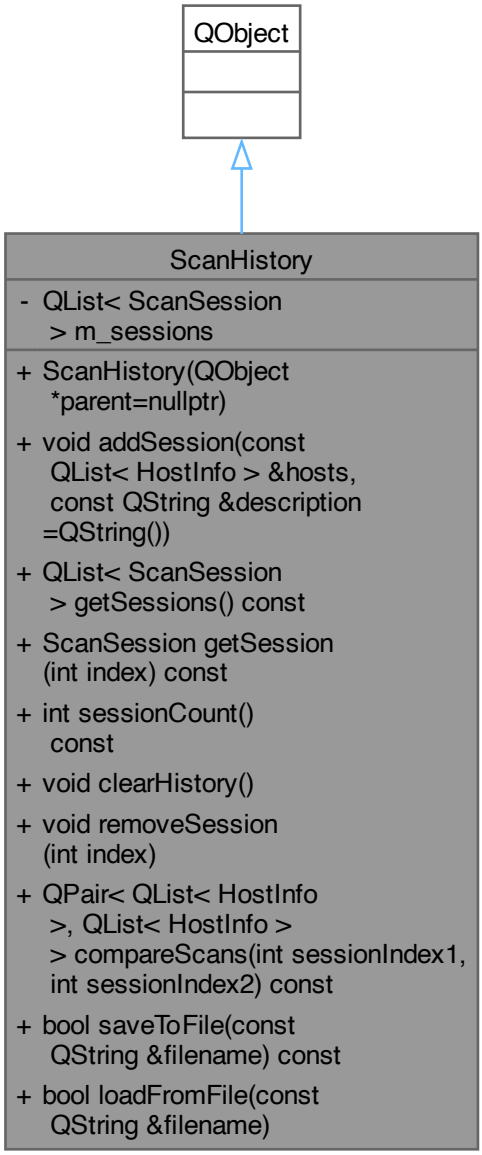
该结构体的文档由以下文件生成:

- build/NetScanner_autogen/EWIEGA46WW/moc_networktopology.cpp

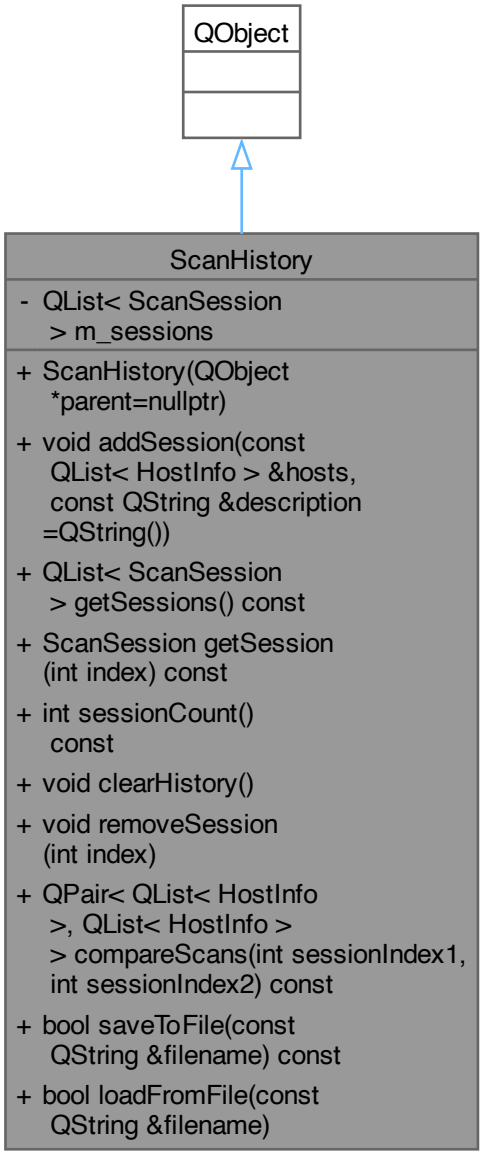
8.14 ScanHistory类 参考

```
#include <scanhistory.h>
```

类 ScanHistory 继承关系图:



ScanHistory 的协作图:



信号

- void historyChanged ()

Public 成员函数

- ScanHistory (QObject *parent=nullptr)
- void addSession (const QList< HostInfo > &hosts, const QString &description=QString())
- QList< ScanSession > getSessions () const

- ScanSession getSession (int index) const
- int sessionCount () const
- void clearHistory ()
- void removeSession (int index)
- QPair< QList< HostInfo >, QList< HostInfo > > compareScans (int sessionIndex1, int sessionIndex2) const
- bool saveToFile (const QString &filename) const
- bool loadFromFile (const QString &filename)

Private 属性

- QList< ScanSession > m_sessions

8.14.1 构造及析构函数说明

8.14.1.1 ScanHistory()

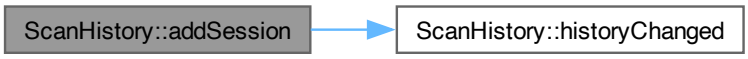
```
ScanHistory::ScanHistory (
            QObject * parent = nullptr)  [explicit]
```

8.14.2 成员函数说明

8.14.2.1 addSession()

```
void ScanHistory::addSession (
            const QList< HostInfo > & hosts,
            const QString & description = QString())
```

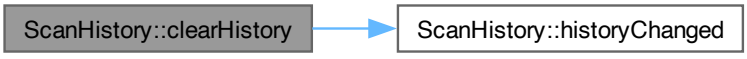
函数调用图:



8.14.2.2 clearHistory()

```
void ScanHistory::clearHistory ()
```

函数调用图:



8.14.2.3 compareScans()

```
QPair< QList< HostInfo >, QList< HostInfo > > ScanHistory::compareScans (
            int sessionIndex1,
            int sessionIndex2) const
```

8.14.2.4 getSession()

```
ScanSession ScanHistory::getSession (
            int index) const
```

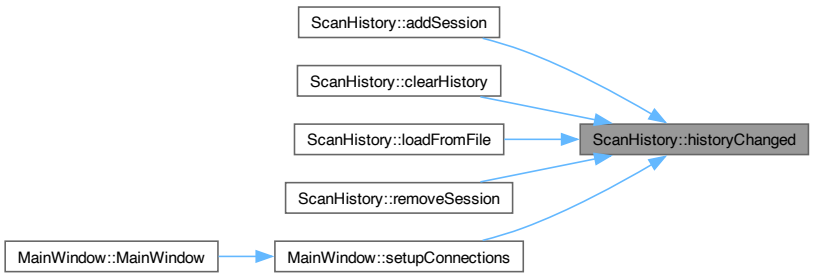
8.14.2.5 getSessions()

```
QList< ScanSession > ScanHistory::getSessions () const  [inline]
```

8.14.2.6 historyChanged

```
void ScanHistory::historyChanged ()  [signal]
```

这是这个函数的调用关系图:



8.14.2.7 loadFromFile()

```
bool ScanHistory::loadFromFile (
            const QString & filename)
```

函数调用图:

8.14.2.8 removeSession()

```
void ScanHistory::removeSession (
            int index)
```

函数调用图:

8.14.2.9 saveToFile()

```
bool ScanHistory::saveToFile (
            const QString & filename) const
```

8.14.2.10 sessionCount()

```
int ScanHistory::sessionCount () const  [inline]
```

8.14.3 类成员变量说明

8.14.3.1 m_sessions

```
QList<ScanSession> ScanHistory::m_sessions  [private]
```

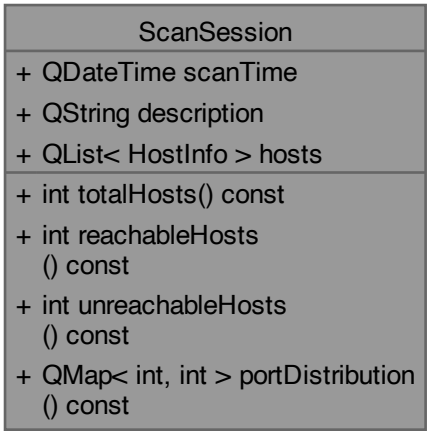
该类的文档由以下文件生成:

- scanhistory.h
- build/NetScanner_autogen/EWIEGA46WW/moc_scanhistory.cpp
- NetScanner_autogen/JRIAJ772TK/moc_scanhistory.cpp
- scanhistory.cpp

8.15 ScanSession结构体 参考

`#include <scanhistory.h>`

ScanSession 的协作图:



Public 成员函数

- int totalHosts () const
- int reachableHosts () const
- int unreachableHosts () const
- QMap< int, int > portDistribution () const

Public 属性

- QDateTime scanTime
- QString description
- QList< HostInfo > hosts

8.15.1 成员函数说明

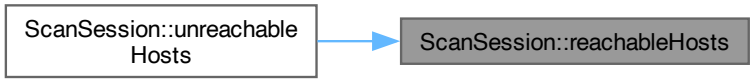
8.15.1.1 portDistribution()

`QMap< int, int > ScanSession::portDistribution () const`

8.15.1.2 reachableHosts()

```
int ScanSession::reachableHosts () const
```

这是这个函数的调用关系图:



8.15.1.3 totalHosts()

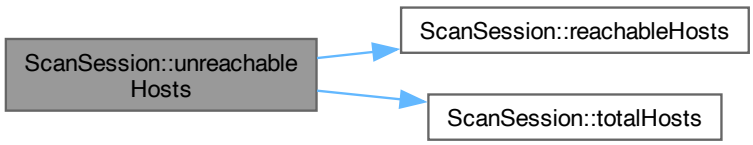
```
int ScanSession::totalHosts () const  [inline]
```

这是这个函数的调用关系图:

8.15.1.4 unreachableHosts()

```
int ScanSession::unreachableHosts () const  [inline]
```

函数调用图:



8.15.2 类成员变量说明

8.15.2.1 description

```
QString ScanSession::description
```

8.15.2.2 hosts

```
QList<HostInfo> ScanSession::hosts
```

8.15.2.3 scanTime

```
QDateTime ScanSession::scanTime
```

该结构体的文档由以下文件生成:

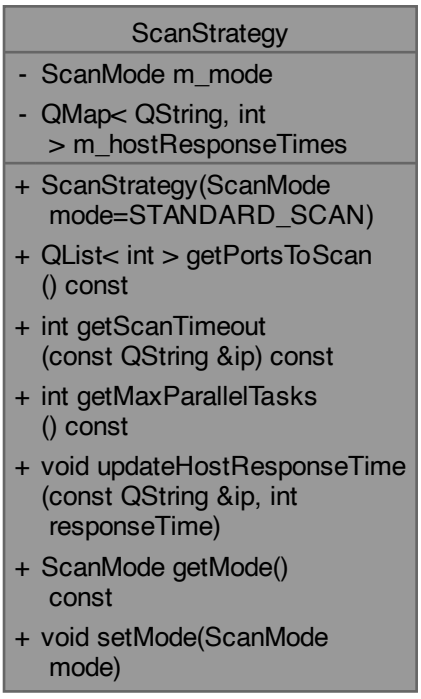
- scanhistory.h
- scanhistory.cpp

8.16 ScanStrategy类 参考

扫描策略类

```
#include <networkscanner.h>
```

ScanStrategy 的协作图:



Public 类型

- enum ScanMode { QUICK_SCAN , STANDARD_SCAN , DEEP_SCAN }
  扫描模式枚举

Public 成员函数

- ScanStrategy (ScanMode mode=STANDARD_SCAN)
  构造函数
- QList< int > getPortsToScan () const
  获取要扫描的端口列表
- int getScanTimeout (const QString &ip) const
  获取扫描超时时间
- int getMaxParallelTasks () const
  获取最大并行任务数
- void updateHostResponseTime (const QString &ip, int responseTime)
  更新主机响应时间记录
- ScanMode getMode () const
  获取当前扫描模式
- void setMode (ScanMode mode)
  设置扫描模式

Private 属性

- ScanMode m_mode
  当前扫描模式
- QMap< QString, int > m_hostResponseTimes
  *IP*地址 -> 响应时间映射

8.16.1 详细描述

扫描策略类

定义不同的扫描模式和参数，如快速扫描、标准扫描和深度扫描

8.16.2 成员枚举类型说明

8.16.2.1 ScanMode

`enum ScanStrategy::ScanMode`

扫描模式枚举

枚举值

QUICK_SCAN	仅检测主机存活
STANDARD_SCAN	扫描常用端口
DEEP_SCAN	全面端口扫描

8.16.3 构造及析构函数说明

8.16.3.1 ScanStrategy()

```
ScanStrategy::ScanStrategy (
            ScanMode mode = STANDARD_SCAN)
```

构造函数

参数

mode	扫描模式，默认为标准扫描

8.16.4 成员函数说明

8.16.4.1 getMaxParallelTasks()

```
int ScanStrategy::getMaxParallelTasks () const
```

获取最大并行任务数

返回

并行任务数量

8.16.4.2 getMode()

```
ScanMode ScanStrategy::getMode () const  [inline]
```

获取当前扫描模式

返回

扫描模式

8.16.4.3 getPortsToScan()

```
QList< int > ScanStrategy::getPortsToScan () const
```

获取要扫描的端口列表

返回

端口号列表

8.16.4.4 getScanTimeout()

```
int ScanStrategy::getScanTimeout (
            const QString & ip) const
```

获取扫描超时时间

参数

ip	目标IP地址

返回

超时时间（毫秒）

8.16.4.5 setMode()

```
void ScanStrategy::setMode (
            ScanMode mode)  [inline]
```

设置扫描模式

参数

mode	要设置的扫描模式

8.16.4.6 updateHostResponseTime()

```
void ScanStrategy::updateHostResponseTime (
            const QString & ip,
            int responseTime)
```

更新主机响应时间记录

参数

ip	主机IP地址
responseTime	响应时间（毫秒）

8.16.5 类成员变量说明

8.16.5.1 m_hostResponseTimes

```
QMap<QString, int> ScanStrategy::m_hostResponseTimes  [private]
```

IP地址 -> 响应时间映射

8.16.5.2 m_mode

```
ScanMode ScanStrategy::m_mode  [private]
```

当前扫描模式

该类的文档由以下文件生成:

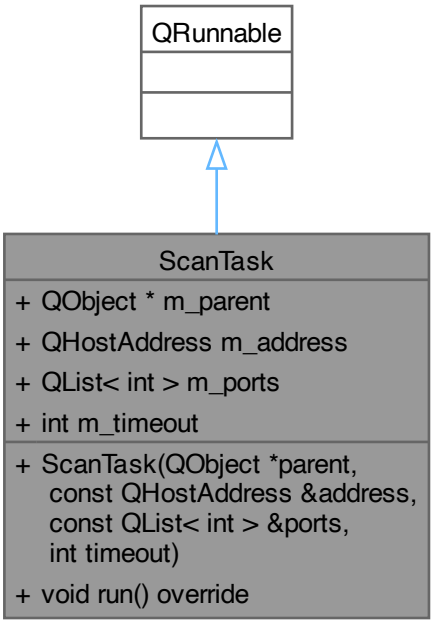
- networkscanner.h
- networkscanner.cpp

8.17 ScanTask类 参考

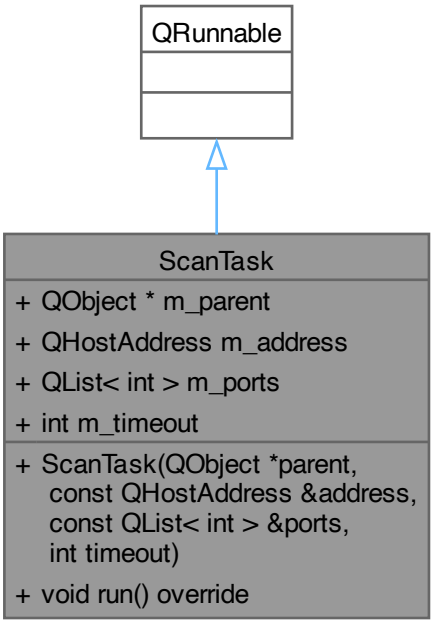
扫描任务类

```
#include <networkscanner.h>
```

类 ScanTask 继承关系图:



ScanTask 的协作图:



Public 成员函数

- ScanTask (QObject ∗parent, const QHostAddress &address, const QList< int > &ports, int timeout)
  构造函数
- void run () override
  执行扫描任务

Public 属性

- QObject ∗ m_parent
  父对象指针
- QHostAddress m_address
  扫描地址
- QList< int > m_ports
  扫描端口列表
- int m_timeout
  超时时间

8.17.1 详细描述

扫描任务类

用于并行执行的单个主机扫描任务

8.17.2 构造及析构函数说明

8.17.2.1 ScanTask()

```
ScanTask::ScanTask (
            QObject * parent,
            const QHostAddress & address,
            const QList< int > & ports,
            int timeout)
```

构造函数

参数

parent	父对象
address	要扫描的地址
ports	要扫描的端口列表
timeout	连接超时时间（毫秒）

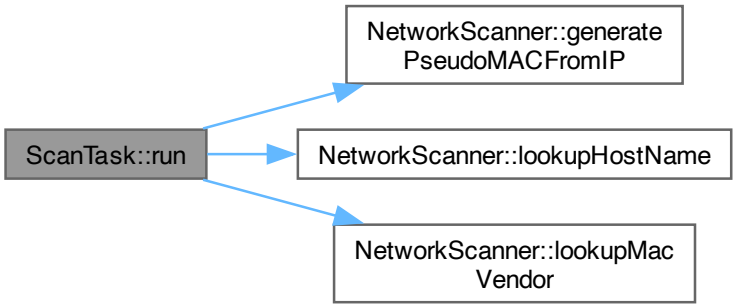
8.17.3 成员函数说明

8.17.3.1 run()

```
void ScanTask::run ()  [override]
```

执行扫描任务

实现QRunnable的抽象方法 函数调用图:



8.17.4 类成员变量说明

8.17.4.1 m_address

```
QHostAddress ScanTask::m_address
```

扫描地址

8.17.4.2 m_parent

```
QObject* ScanTask::m_parent
```

父对象指针

8.17.4.3 m_ports

```
QList<int> ScanTask::m_ports
```

扫描端口列表

8.17.4.4 m_timeout

```
int ScanTask::m_timeout
```

超时时间

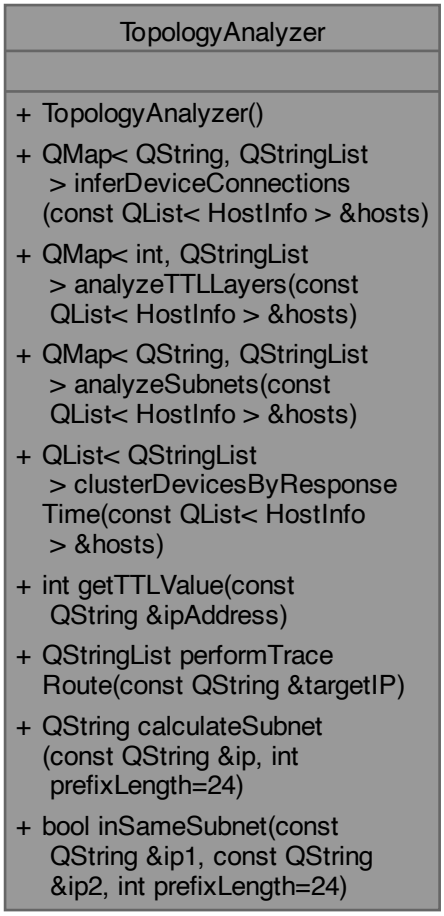
该类的文档由以下文件生成:

- networkscanner.h
- networkscanner.cpp

8.18 TopologyAnalyzer类 参考

```
#include <networktopology.h>
```

TopologyAnalyzer 的协作图:



Public 成员函数

- TopologyAnalyzer ()
- QMap< QString, QStringList > inferDeviceConnections (const QList< HostInfo > &hosts)
- QMap< int, QStringList > analyzeTTLLayers (const QList< HostInfo > &hosts)
- QMap< QString, QStringList > analyzeSubnets (const QList< HostInfo > &hosts)
- QList< QStringList > clusterDevicesByResponseTime (const QList< HostInfo > &hosts)
- int getTTLValue (const QString &ipAddress)
- QStringList performTraceRoute (const QString &targetIP)
- QString calculateSubnet (const QString &ip, int prefixLength=24)
- bool inSameSubnet (const QString &ip1, const QString &ip2, int prefixLength=24)

8.18.1 构造及析构函数说明

8.18.1.1 TopologyAnalyzer()

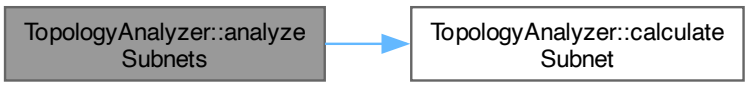
```
TopologyAnalyzer::TopologyAnalyzer ()
```

8.18.2 成员函数说明

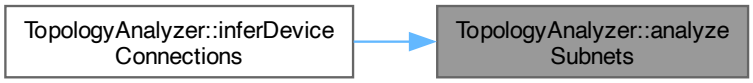
8.18.2.1 analyzeSubnets()

```
QMap< QString, QStringList > TopologyAnalyzer::analyzeSubnets (
            const QList< HostInfo > & hosts)
```

函数调用图:



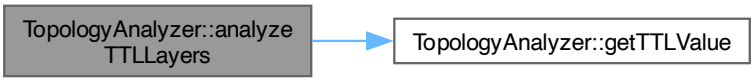
这是这个函数的调用关系图:



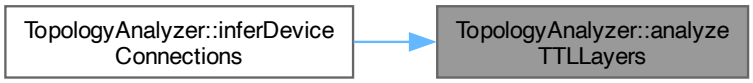
8.18.2.2 analyzeTTLLayers()

```
QMap< int, QStringList > TopologyAnalyzer::analyzeTTLLayers (
            const QList< HostInfo > & hosts)
```

函数调用图:



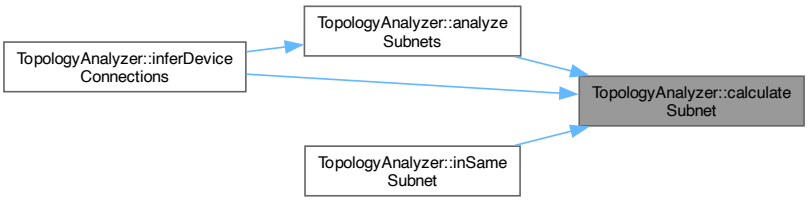
这是这个函数的调用关系图:



8.18.2.3 calculateSubnet()

```
QString TopologyAnalyzer::calculateSubnet (
            const QString & ip,
            int prefixLength = 24)
```

这是这个函数的调用关系图:



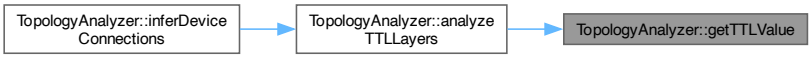
8.18.2.4 clusterDevicesByResponseTime()

```
QList< QStringList > TopologyAnalyzer::clusterDevicesByResponseTime (
            const QList< HostInfo > & hosts)
```

8.18.2.5 getTTLValue()

```
int TopologyAnalyzer::getTTLValue (
            const QString & ipAddress)
```

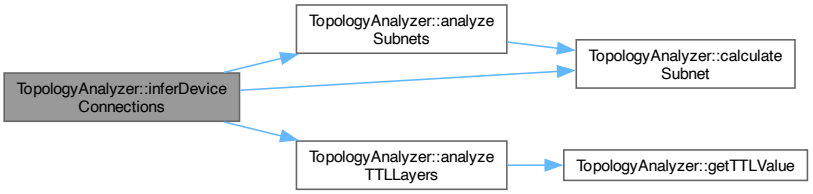
这是这个函数的调用关系图:



8.18.2.6 inferDeviceConnections()

```
QMap< QString, QStringList > TopologyAnalyzer::inferDeviceConnections (
            const QList< HostInfo > & hosts)
```

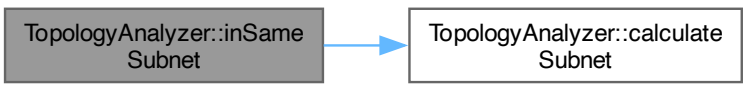
函数调用图:



8.18.2.7 inSameSubnet()

```
bool TopologyAnalyzer::inSameSubnet (
            const QString & ip1,
            const QString & ip2,
            int prefixLength = 24)
```

函数调用图:



8.18.2.8 performTraceRoute()

```
QStringList TopologyAnalyzer::performTraceRoute (
            const QString & targetIP)
```

该类的文档由以下文件生成:

- networktopology.h
- networktopology.cpp

Chapter 9

文件说明

9.1 build/CMakeFiles/3.31.5/CompilerIdCXX/CMakeCXXCompilerId.cpp 文件参考

宏定义

- #define __has_include(x)
- #define COMPILER_ID ""
- #define STRINGIFY_HELPER(X)
- #define STRINGIFY(X)
- #define PLATFORM_ID
- #define ARCHITECTURE_ID
- #define DEC(n)
- #define HEX(n)
- #define CXX_STD_98 199711L
- #define CXX_STD_11 201103L
- #define CXX_STD_14 201402L
- #define CXX_STD_17 201703L
- #define CXX_STD_20 202002L
- #define CXX_STD_23 202302L
- #define CXX_STD __cplusplus

函数

- int main (int argc, char *argv[ ])

变量

- char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
- char const * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
- char const * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
- const char * info_language_standard_default
- const char * info_language_extensions_default

9.1.1 宏定义说明

9.1.1.1 __has_include

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

值:

```
0
```

9.1.1.2 ARCHITECTURE_ID

```
#define ARCHITECTURE_ID
```

9.1.1.3 COMPILER_ID

```
#define COMPILER_ID ""
```

9.1.1.4 CXX_STD

```
#define CXX_STD __cplusplus
```

9.1.1.5 CXX_STD_11

```
#define CXX_STD_11 201103L
```

9.1.1.6 CXX_STD_14

```
#define CXX_STD_14 201402L
```

9.1.1.7 CXX_STD_17

```
#define CXX_STD_17 201703L
```

9.1.1.8 CXX_STD_20

```
#define CXX_STD_20 202002L
```

9.1.1.9 CXX_STD_23

```
#define CXX_STD_23 202302L
```

9.1.1.10 CXX_STD_98

```
#define CXX_STD_98 199711L
```

9.1.1.11 DEC

```
#define DEC(
              n)
```

值:

```
  ('0' + (((n) / 10000000)%10)), \
  ('0' + (((n) / 1000000)%10)),  \
  ('0' + (((n) / 100000)%10)),   \
  ('0' + (((n) / 10000)%10)),    \
  ('0' + (((n) / 1000)%10)),     \
  ('0' + (((n) / 100)%10)),      \
  ('0' + (((n) / 10)%10)),       \
  ('0' +  ((n) % 10))
```

9.1.1.12 HEX

```
#define HEX(
              n)
```

值:

```
  ('0' + ((n)>>28 & 0xF)), \
  ('0' + ((n)>>24 & 0xF)), \
  ('0' + ((n)>>20 & 0xF)), \
  ('0' + ((n)>>16 & 0xF)), \
  ('0' + ((n)>>12 & 0xF)), \
  ('0' + ((n)>>8  & 0xF)), \
  ('0' + ((n)>>4  & 0xF)), \
  ('0' + ((n)     & 0xF))
```

9.1.1.13 PLATFORM_ID

```
#define PLATFORM_ID
```

9.1.1.14 STRINGIFY

```
#define STRINGIFY(
              X)
```

值:

```
STRINGIFY_HELPER(X)
```

9.1.1.15 STRINGIFY_HELPER

```
#define STRINGIFY_HELPER(
              X)
```

值:

```
#X
```

9.1.2 函数说明

9.1.2.1 main()

```
int main (
             int argc,
             char * argv[])
```

9.1.3 变量说明

9.1.3.1 info_arch

```
char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
```

9.1.3.2 info_compiler

```
char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
```

9.1.3.3 info_language_extensions_default

```
const char* info_language_extensions_default
```

初始值:

```
= "INFO" ":" "extensions_default["



  "OFF"
"]"
```

9.1.3.4 info_language_standard_default

```
const char* info_language_standard_default
```

初始值:

```
= "INFO" ":" "standard_default["









  "98"
"]"
```

9.1.3.5 info_platform

```
char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
```

9.2 build/CMakeFiles/NetScanner.dir/deviceanalyzer.cpp.o.d 文件参考

9.3 build/CMakeFiles/NetScanner.dir/main.cpp.o.d 文件参考

9.4 build/CMakeFiles/NetScanner.dir/mainwindow.cpp.o.d 文件参考

9.5 build/CMakeFiles/NetScanner_autogen/mocs_compilation.cpp.o.d 文件参考

9.6 build/CMakeFiles/NetScanner.dir/networkscanner.cpp.o.d 文件参考

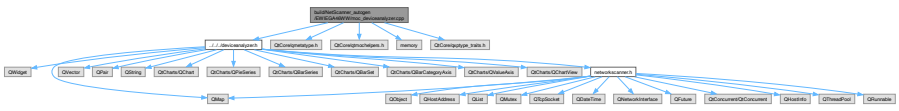
9.7 build/CMakeFiles/NetScanner.dir/networktopology.cpp.o.d 文件参考

9.8 build/CMakeFiles/NetScanner.dir/scanhistory.cpp.o.d 文件参考

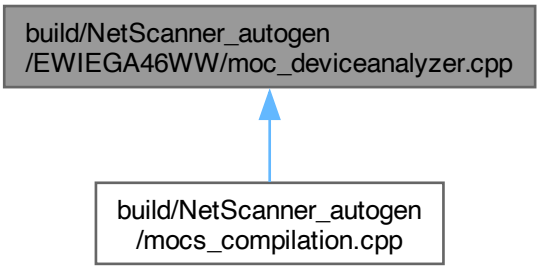
9.9 build/NetScanner_autogen/EWIEGA46WW/moc_deviceanalyzer.cpp 文件参考

```
#include "../../../deviceanalyzer.h"
#include <QtCore/qmetatype.h>
#include <QtCore/qtmochelpers.h>
#include <memory>
#include <QtCore/qxptype_traits.h>
```

moc_deviceanalyzer.cpp 的引用(Include)关系图:



此图展示该文件被哪些文件直接或间接地引用了:



类

- struct QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN14DeviceAnalyzerE_t

命名空间

- namespace QT_WARNING_DISABLE_DEPRECATED

宏定义

- #define Q_CONSTINIT

9.9.1 宏定义说明

9.9.1.1 Q_CONSTINIT

```
#define Q_CONSTINIT
```

9.10 moc_deviceanalyzer.cpp

浏览该文件的文档.

```
00001 /****************************************************************************
00002 ** Meta object code from reading C++ file 'deviceanalyzer.h'
00003 **
00004 ** Created by: The Qt Meta Object Compiler version 69 (Qt 6.9.0)
00005 **
00006 ** WARNING! All changes made in this file will be lost!
00007 *****************************************************************************/
00008
00009 #include "../../../deviceanalyzer.h"
00010 #include <QtCore/qmetatype.h>
00011
00012 #include <QtCore/qtmochelpers.h>
00013
00014 #include <memory>
00015
```

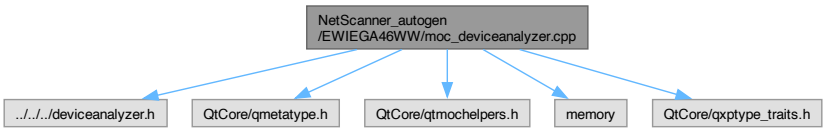
```
00016
00017 #include <QtCore/qxptype_traits.h>
00018 #if !defined(Q_MOC_OUTPUT_REVISION)
00019 #error "The header file 'deviceanalyzer.h' doesn't include <QObject>."
00020 #elif Q_MOC_OUTPUT_REVISION != 69
00021 #error "This file was generated using the moc from 6.9.0. It"
00022 #error "cannot be used with the include files from this version of Qt."
00023 #error "(The moc has changed too much.)"
00024 #endif
00025
00026 #ifndef Q_CONSTINIT
00027 #define Q_CONSTINIT
00028 #endif
00029
00030 QT_WARNING_PUSH
00031 QT_WARNING_DISABLE_DEPRECATED
00032 QT_WARNING_DISABLE_GCC("-Wuseless-cast")
00033 namespace {
00034 struct qt_meta_tag_ZN14DeviceAnalyzerE_t {};
00035 } // unnamed namespace
00036
00037 template <> constexpr inline auto
      DeviceAnalyzer::qt_create_metaobjectdata<qt_meta_tag_ZN14DeviceAnalyzerE_t>()
00038 {
00039     namespace QMC = QtMocConstants;
00040     QtMocHelpers::StringRefStorage qt_stringData {
00041         "DeviceAnalyzer",
00042         "analysisCompleted",
00043         ""
00044     };
00045
00046     QtMocHelpers::UintData qt_methods {
00047         // Signal 'analysisCompleted'
00048         QtMocHelpers::SignalData<void()>(1, 2, QMC::AccessPublic, QMetaType::Void),
00049     };
00050     QtMocHelpers::UintData qt_properties {
00051     };
00052     QtMocHelpers::UintData qt_enums {
00053     };
00054     return QtMocHelpers::metaObjectData<DeviceAnalyzer,
      qt_meta_tag_ZN14DeviceAnalyzerE_t>(QMC::MetaObjectFlag{}, qt_stringData,
00055             qt_methods, qt_properties, qt_enums);
00056 }
00057 Q_CONSTINIT const QMetaObject DeviceAnalyzer::staticMetaObject = { {
00058     QMetaObject::SuperData::link<QWidget::staticMetaObject>(),
00059     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN14DeviceAnalyzerE_t>.stringdata,
00060     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN14DeviceAnalyzerE_t>.data,
00061     qt_static_metacall,
00062     nullptr,
00063     qt_staticMetaObjectRelocatingContent<qt_meta_tag_ZN14DeviceAnalyzerE_t>.metaTypes,
00064     nullptr
00065 } };
00066
00067 void DeviceAnalyzer::qt_static_metacall(QObject *_o, QMetaObject::Call _c, int _id, void **_a)
00068 {
00069     auto *_t = static_cast<DeviceAnalyzer *>(_o);
00070     if (_c == QMetaObject::InvokeMetaMethod) {
00071         switch (_id) {
00072         case 0: _t->analysisCompleted(); break;
00073         default: ;
00074         }
00075     }
00076     if (_c == QMetaObject::IndexOfMethod) {
00077         if (QtMocHelpers::indexOfMethod<void (DeviceAnalyzer::*)()>(_a,
      &DeviceAnalyzer::analysisCompleted, 0))
00078             return;
00079     }
00080 }
00081
00082 const QMetaObject *DeviceAnalyzer::metaObject() const
00083 {
00084     return QObject::d_ptr->metaObject ? QObject::d_ptr->dynamicMetaObject() : &staticMetaObject;
00085 }
00086
00087 void *DeviceAnalyzer::qt_metacast(const char *_clname)
00088 {
00089     if (!_clname) return nullptr;
00090     if (!strcmp(_clname, qt_staticMetaObjectStaticContent<qt_meta_tag_ZN14DeviceAnalyzerE_t>.strings))
00091         return static_cast<void*>(this);
00092     return QWidget::qt_metacast(_clname);
00093 }
00094
00095 int DeviceAnalyzer::qt_metacall(QMetaObject::Call _c, int _id, void **_a)
00096 {
00097     _id = QWidget::qt_metacall(_c, _id, _a);
00098     if (_id < 0)
00099         return _id;
```

```
00100     if (_c == QMetaObject::InvokeMetaMethod) {
00101         if (_id < 1)
00102             qt_static_metacall(this, _c, _id, _a);
00103         _id -= 1;
00104     }
00105     if (_c == QMetaObject::RegisterMethodArgumentMetaType) {
00106         if (_id < 1)
00107             *reinterpret_cast<QMetaType *>(_a[0]) = QMetaType();
00108         _id -= 1;
00109     }
00110     return _id;
00111 }
00112
00113 // SIGNAL 0
00114 void DeviceAnalyzer::analysisCompleted()
00115 {
00116     QMetaObject::activate(this, &staticMetaObject, 0, nullptr);
00117 }
00118 QT_WARNING_POP
```

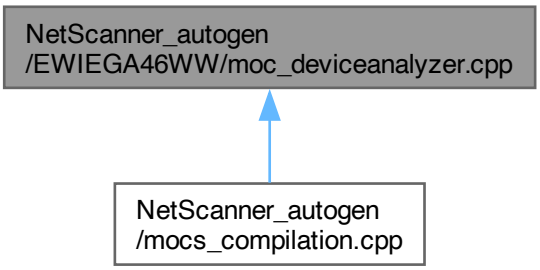
9.11 NetScanner_autogen/EWIEGA46WW/moc_deviceanalyzer.cpp 文件参考

```
#include "../../../deviceanalyzer.h"
#include <QtCore/qmetatype.h>
#include <QtCore/qtmochelpers.h>
#include <memory>
#include <QtCore/qxptype_traits.h>
```

moc_deviceanalyzer.cpp 的引用(Include)关系图:



此图展示该文件被哪些文件直接或间接地引用了:



类

- struct QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN14DeviceAnalyzerE_t

命名空间

- namespace QT_WARNING_DISABLE_DEPRECATED

宏定义

- #define Q_CONSTINIT

9.11.1 宏定义说明

9.11.1.1 Q_CONSTINIT

```
#define Q_CONSTINIT
```

9.12 moc_deviceanalyzer.cpp

浏览该文件的文档.

```
00001 /****************************************************************************
00002 ** Meta object code from reading C++ file 'deviceanalyzer.h'
00003 **
00004 ** Created by: The Qt Meta Object Compiler version 69 (Qt 6.9.0)
00005 **
00006 ** WARNING! All changes made in this file will be lost!
00007 *****************************************************************************/
00008
00009 #include "../../../deviceanalyzer.h"
00010 #include <QtCore/qmetatype.h>
00011
00012 #include <QtCore/qtmochelpers.h>
00013
00014 #include <memory>
00015
00016
00017 #include <QtCore/qxptype_traits.h>
00018 #if !defined(Q_MOC_OUTPUT_REVISION)
00019 #error "The header file 'deviceanalyzer.h' doesn't include <QObject>."
00020 #elif Q_MOC_OUTPUT_REVISION != 69
00021 #error "This file was generated using the moc from 6.9.0. It"
00022 #error "cannot be used with the include files from this version of Qt."
00023 #error "(The moc has changed too much.)"
00024 #endif
00025
00026 #ifndef Q_CONSTINIT
00027 #define Q_CONSTINIT
00028 #endif
00029
00030 QT_WARNING_PUSH
00031 QT_WARNING_DISABLE_DEPRECATED
00032 QT_WARNING_DISABLE_GCC("-Wuseless-cast")
00033 namespace {
00034 struct qt_meta_tag_ZN14DeviceAnalyzerE_t {};
00035 } // unnamed namespace
00036
00037 template <> constexpr inline auto
      DeviceAnalyzer::qt_create_metaobjectdata<qt_meta_tag_ZN14DeviceAnalyzerE_t>()
00038 {
00039     namespace QMC = QtMocConstants;
00040     QtMocHelpers::StringRefStorage qt_stringData {
00041         "DeviceAnalyzer",
00042         "analysisCompleted",
00043         ""
00044     };
00045
00046     QtMocHelpers::UintData qt_methods {
00047         // Signal 'analysisCompleted'
00048         QtMocHelpers::SignalData<void()>(1, 2, QMC::AccessPublic, QMetaType::Void),
00049     };
00050     QtMocHelpers::UintData qt_properties {
00051     };
```

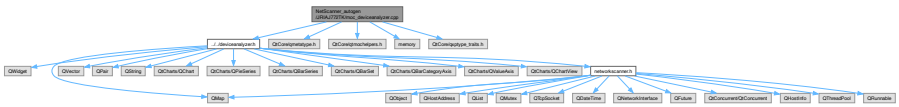
```
00052     QtMocHelpers::UintData qt_enums {
00053     };
00054     return QtMocHelpers::metaObjectData<DeviceAnalyzer,
      qt_meta_tag_ZN14DeviceAnalyzerE_t>(QMC::MetaObjectFlag{}, qt_stringData,
00055             qt_methods, qt_properties, qt_enums);
00056 }
00057 Q_CONSTINIT const QMetaObject DeviceAnalyzer::staticMetaObject = { {
00058     QMetaObject::SuperData::link<QWidget::staticMetaObject>(),
00059     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN14DeviceAnalyzerE_t>.stringdata,
00060     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN14DeviceAnalyzerE_t>.data,
00061     qt_static_metacall,
00062     nullptr,
00063     qt_staticMetaObjectRelocatingContent<qt_meta_tag_ZN14DeviceAnalyzerE_t>.metaTypes,
00064     nullptr
00065 } };
00066
00067 void DeviceAnalyzer::qt_static_metacall(QObject *_o, QMetaObject::Call _c, int _id, void **_a)
00068 {
00069     auto *_t = static_cast<DeviceAnalyzer *>(_o);
00070     if (_c == QMetaObject::InvokeMetaMethod) {
00071         switch (_id) {
00072         case 0: _t->analysisCompleted(); break;
00073         default: ;
00074         }
00075     }
00076     if (_c == QMetaObject::IndexOfMethod) {
00077         if (QtMocHelpers::indexOfMethod<void (DeviceAnalyzer::*)()>(_a,
      &DeviceAnalyzer::analysisCompleted, 0))
00078             return;
00079     }
00080 }
00081
00082 const QMetaObject *DeviceAnalyzer::metaObject() const
00083 {
00084     return QObject::d_ptr->metaObject ? QObject::d_ptr->dynamicMetaObject() : &staticMetaObject;
00085 }
00086
00087 void *DeviceAnalyzer::qt_metacast(const char *_clname)
00088 {
00089     if (!_clname) return nullptr;
00090     if (!strcmp(_clname, qt_staticMetaObjectStaticContent<qt_meta_tag_ZN14DeviceAnalyzerE_t>.strings))
00091         return static_cast<void*>(this);
00092     return QWidget::qt_metacast(_clname);
00093 }
00094
00095 int DeviceAnalyzer::qt_metacall(QMetaObject::Call _c, int _id, void **_a)
00096 {
00097     _id = QWidget::qt_metacall(_c, _id, _a);
00098     if (_id < 0)
00099         return _id;
00100     if (_c == QMetaObject::InvokeMetaMethod) {
00101         if (_id < 1)
00102             qt_static_metacall(this, _c, _id, _a);
00103         _id -= 1;
00104     }
00105     if (_c == QMetaObject::RegisterMethodArgumentMetaType) {
00106         if (_id < 1)
00107             *reinterpret_cast<QMetaType *>(_a[0]) = QMetaType();
00108         _id -= 1;
00109     }
00110     return _id;
00111 }
00112
00113 // SIGNAL 0
00114 void DeviceAnalyzer::analysisCompleted()
00115 {
00116     QMetaObject::activate(this, &staticMetaObject, 0, nullptr);
00117 }
00118 QT_WARNING_POP
```

9.13 NetScanner_autogen/JRIAJ772TK/moc_deviceanalyzer.cpp 文件参考

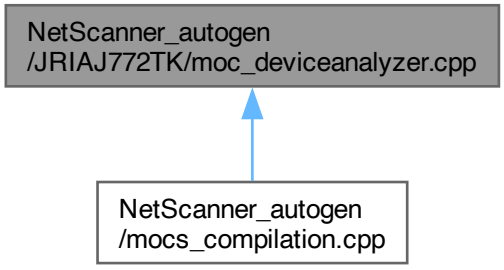
```
#include "../../deviceanalyzer.h"
#include <QtCore/qmetatype.h>
#include <QtCore/qtmochelpers.h>
#include <memory>
```

```
#include <QtCore/qxptype_traits.h>
```

moc_deviceanalyzer.cpp 的引用(Include)关系图:



此图展示该文件被哪些文件直接或间接地引用了:



类

- struct QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN14DeviceAnalyzerE_t

命名空间

- namespace QT_WARNING_DISABLE_DEPRECATED

宏定义

- #define Q_CONSTINIT

9.13.1 宏定义说明

9.13.1.1 Q_CONSTINIT

```
#define Q_CONSTINIT
```

9.14 moc_deviceanalyzer.cpp

浏览该文件的文档.

```
00001 /****************************************************************************
00002 ** Meta object code from reading C++ file 'deviceanalyzer.h'
00003 **
00004 ** Created by: The Qt Meta Object Compiler version 69 (Qt 6.9.0)
00005 **
00006 ** WARNING! All changes made in this file will be lost!
00007 *****************************************************************************/
00008
00009 #include "../../deviceanalyzer.h"
00010 #include <QtCore/qmetatype.h>
00011
00012 #include <QtCore/qtmochelpers.h>
00013
00014 #include <memory>
00015
00016
00017 #include <QtCore/qxptype_traits.h>
00018 #if !defined(Q_MOC_OUTPUT_REVISION)
00019 #error "The header file 'deviceanalyzer.h' doesn't include <QObject>."
00020 #elif Q_MOC_OUTPUT_REVISION != 69
00021 #error "This file was generated using the moc from 6.9.0. It"
00022 #error "cannot be used with the include files from this version of Qt."
00023 #error "(The moc has changed too much.)"
00024 #endif
00025
00026 #ifndef Q_CONSTINIT
00027 #define Q_CONSTINIT
00028 #endif
00029
00030 QT_WARNING_PUSH
00031 QT_WARNING_DISABLE_DEPRECATED
00032 QT_WARNING_DISABLE_GCC("-Wuseless-cast")
00033 namespace {
00034 struct qt_meta_tag_ZN14DeviceAnalyzerE_t {};
00035 } // unnamed namespace
00036
00037 template <> constexpr inline auto
      DeviceAnalyzer::qt_create_metaobjectdata<qt_meta_tag_ZN14DeviceAnalyzerE_t>()
00038 {
00039     namespace QMC = QtMocConstants;
00040     QtMocHelpers::StringRefStorage qt_stringData {
00041         "DeviceAnalyzer",
00042         "analysisCompleted",
00043         ""
00044     };
00045
00046     QtMocHelpers::UintData qt_methods {
00047         // Signal 'analysisCompleted'
00048         QtMocHelpers::SignalData<void()>(1, 2, QMC::AccessPublic, QMetaType::Void),
00049     };
00050     QtMocHelpers::UintData qt_properties {
00051     };
00052     QtMocHelpers::UintData qt_enums {
00053     };
00054     return QtMocHelpers::metaObjectData<DeviceAnalyzer,
      qt_meta_tag_ZN14DeviceAnalyzerE_t>(QMC::MetaObjectFlag{}, qt_stringData,
00055             qt_methods, qt_properties, qt_enums);
00056 }
00057 Q_CONSTINIT const QMetaObject DeviceAnalyzer::staticMetaObject = { {
00058     QMetaObject::SuperData::link<QWidget::staticMetaObject>(),
00059     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN14DeviceAnalyzerE_t>.stringdata,
00060     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN14DeviceAnalyzerE_t>.data,
00061     qt_static_metacall,
00062     nullptr,
00063     qt_staticMetaObjectRelocatingContent<qt_meta_tag_ZN14DeviceAnalyzerE_t>.metaTypes,
00064     nullptr
00065 } };
00066
00067 void DeviceAnalyzer::qt_static_metacall(QObject *_o, QMetaObject::Call _c, int _id, void **_a)
00068 {
00069     auto *_t = static_cast<DeviceAnalyzer *>(_o);
00070     if (_c == QMetaObject::InvokeMetaMethod) {
00071         switch (_id) {
00072         case 0: _t->analysisCompleted(); break;
00073         default: ;
00074         }
00075     }
00076     if (_c == QMetaObject::IndexOfMethod) {
00077         if (QtMocHelpers::indexOfMethod<void (DeviceAnalyzer::*)()>(_a,
      &DeviceAnalyzer::analysisCompleted, 0))
00078             return;
00079     }
```

```
00080 }
00081
00082 const QMetaObject *DeviceAnalyzer::metaObject() const
00083 {
00084     return QObject::d_ptr->metaObject ? QObject::d_ptr->dynamicMetaObject() : &staticMetaObject;
00085 }
00086
00087 void *DeviceAnalyzer::qt_metacast(const char *_clname)
00088 {
00089     if (!_clname) return nullptr;
00090     if (!strcmp(_clname, qt_staticMetaObjectStaticContent<qt_meta_tag_ZN14DeviceAnalyzerE_t>.strings))
00091         return static_cast<void*>(this);
00092     return QWidget::qt_metacast(_clname);
00093 }
00094
00095 int DeviceAnalyzer::qt_metacall(QMetaObject::Call _c, int _id, void **_a)
00096 {
00097     _id = QWidget::qt_metacall(_c, _id, _a);
00098     if (_id < 0)
00099         return _id;
00100     if (_c == QMetaObject::InvokeMetaMethod) {
00101         if (_id < 1)
00102             qt_static_metacall(this, _c, _id, _a);
00103         _id -= 1;
00104     }
00105     if (_c == QMetaObject::RegisterMethodArgumentMetaType) {
00106         if (_id < 1)
00107             *reinterpret_cast<QMetaType *>(_a[0]) = QMetaType();
00108         _id -= 1;
00109     }
00110     return _id;
00111 }
00112
00113 // SIGNAL 0
00114 void DeviceAnalyzer::analysisCompleted()
00115 {
00116     QMetaObject::activate(this, &staticMetaObject, 0, nullptr);
00117 }
00118 QT_WARNING_POP
```

9.15 build/NetScanner_autogen/EWIEGA46WW/moc_↩ deviceanalyzer.cpp.d 文件参考

9.16 NetScanner_autogen/EWIEGA46WW/moc_deviceanalyzer.cpp.d 文件参考

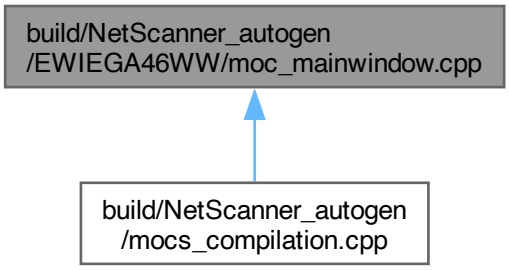
9.17 NetScanner_autogen/JRIAJ772TK/moc_deviceanalyzer.cpp.d 文件参考

9.18 build/NetScanner_autogen/EWIEGA46WW/moc_mainwindow.cpp 文件参考

```
#include "../../../mainwindow.h"
#include <QtGui/qtextcursor.h>
#include <QtCore/qmetatype.h>
#include <QtCore/qtmochelpers.h>
#include <memory>
#include <QtCore/qxptype_traits.h>
```

moc_mainwindow.cpp 的引用(Include)关系图:

此图展示该文件被哪些文件直接或间接地引用了:



类

- struct QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN10MainWindowE_t

命名空间

- namespace QT_WARNING_DISABLE_DEPRECATED

宏定义

- #define Q_CONSTINIT

9.18.1 宏定义说明

9.18.1.1 Q_CONSTINIT

```
#define Q_CONSTINIT
```

9.19 moc_mainwindow.cpp

浏览该文件的文档.

```
00001 /****************************************************************************
00002 ** Meta object code from reading C++ file 'mainwindow.h'
00003 **
00004 ** Created by: The Qt Meta Object Compiler version 69 (Qt 6.9.0)
00005 **
00006 ** WARNING! All changes made in this file will be lost!
00007 *****************************************************************************/
00008
00009 #include "../../../mainwindow.h"
00010 #include <QtGui/qtextcursor.h>
00011 #include <QtCore/qmetatype.h>
00012
00013 #include <QtCore/qtmochelpers.h>
00014
00015 #include <memory>
```

```
00016
00017
00018 #include <QtCore/qxptype_traits.h>
00019 #if !defined(Q_MOC_OUTPUT_REVISION)
00020 #error "The header file 'mainwindow.h' doesn't include <QObject>."
00021 #elif Q_MOC_OUTPUT_REVISION != 69
00022 #error "This file was generated using the moc from 6.9.0. It"
00023 #error "cannot be used with the include files from this version of Qt."
00024 #error "(The moc has changed too much.)"
00025 #endif
00026
00027 #ifndef Q_CONSTINIT
00028 #define Q_CONSTINIT
00029 #endif
00030
00031 QT_WARNING_PUSH
00032 QT_WARNING_DISABLE_DEPRECATED
00033 QT_WARNING_DISABLE_GCC("-Wuseless-cast")
00034 namespace {
00035 struct qt_meta_tag_ZN10MainWindowE_t {};
00036 } // unnamed namespace
00037
00038 template <> constexpr inline auto MainWindow::qt_create_metaobjectdata<qt_meta_tag_ZN10MainWindowE_t>()
00039 {
00040     namespace QMC = QtMocConstants;
00041     QtMocHelpers::StringRefStorage qt_stringData {
00042         "MainWindow",
00043         "startScan",
00044         "",
00045         "stopScan",
00046         "onHostFound",
00047         "HostInfo",
00048         "host",
00049         "onScanStarted",
00050         "onScanFinished",
00051         "onScanProgress",
00052         "progress",
00053         "onScanError",
00054         "errorMessage",
00055         "saveResults",
00056         "clearResults",
00057         "showSettings",
00058         "applySettings",
00059         "showAbout",
00060         "showHostDetails",
00061         "row",
00062         "column",
00063         "exportToCSV",
00064         "togglePortScanOptions",
00065         "checked",
00066         "toggleRangeOptions",
00067         "showTopologyView",
00068         "showStatisticsView",
00069         "showHistoryView",
00070         "generateSecurityReport",
00071         "saveTopologyImage",
00072         "toggleDarkMode",
00073         "enable",
00074         "compareScanResults",
00075         "scheduleScan",
00076         "saveHistoryToFile",
00077         "loadHistoryFromFile",
00078         "updateNetworkTopology",
00079         "refreshTopology",
00080         "filterResults",
00081         "clearFilters",
00082         "onThemeChanged"
00083     };
00084
00085     QtMocHelpers::UintData qt_methods {
00086         // Slot 'startScan'
00087         QtMocHelpers::SlotData<void()>(1, 2, QMC::AccessPrivate, QMetaType::Void),
00088         // Slot 'stopScan'
00089         QtMocHelpers::SlotData<void()>(3, 2, QMC::AccessPrivate, QMetaType::Void),
00090         // Slot 'onHostFound'
00091         QtMocHelpers::SlotData<void(const HostInfo &)>(4, 2, QMC::AccessPrivate, QMetaType::Void, {{
00092             { 0x80000000 | 5, 6 },
00093         }}),
00094         // Slot 'onScanStarted'
00095         QtMocHelpers::SlotData<void()>(7, 2, QMC::AccessPrivate, QMetaType::Void),
00096         // Slot 'onScanFinished'
00097         QtMocHelpers::SlotData<void()>(8, 2, QMC::AccessPrivate, QMetaType::Void),
00098         // Slot 'onScanProgress'
00099         QtMocHelpers::SlotData<void(int)>(9, 2, QMC::AccessPrivate, QMetaType::Void, {{
00100             { QMetaType::Int, 10 },
00101         }}),
00102         // Slot 'onScanError'
```

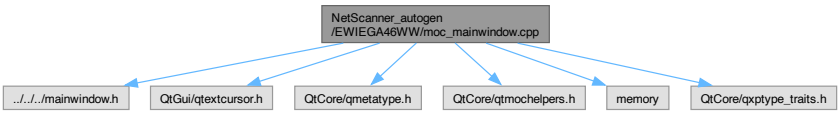
```
00103         QtMocHelpers::SlotData<void(const QString &)>(11, 2, QMC::AccessPrivate, QMetaType::Void, {{
00104             { QMetaType::QString, 12 },
00105         }}),
00106         // Slot 'saveResults'
00107         QtMocHelpers::SlotData<void()>(13, 2, QMC::AccessPrivate, QMetaType::Void),
00108         // Slot 'clearResults'
00109         QtMocHelpers::SlotData<void()>(14, 2, QMC::AccessPrivate, QMetaType::Void),
00110         // Slot 'showSettings'
00111         QtMocHelpers::SlotData<void()>(15, 2, QMC::AccessPrivate, QMetaType::Void),
00112         // Slot 'applySettings'
00113         QtMocHelpers::SlotData<void()>(16, 2, QMC::AccessPrivate, QMetaType::Void),
00114         // Slot 'showAbout'
00115         QtMocHelpers::SlotData<void()>(17, 2, QMC::AccessPrivate, QMetaType::Void),
00116         // Slot 'showHostDetails'
00117         QtMocHelpers::SlotData<void(int, int)>(18, 2, QMC::AccessPrivate, QMetaType::Void, {{
00118             { QMetaType::Int, 19 }, { QMetaType::Int, 20 },
00119         }}),
00120         // Slot 'exportToCSV'
00121         QtMocHelpers::SlotData<void()>(21, 2, QMC::AccessPrivate, QMetaType::Void),
00122         // Slot 'togglePortScanOptions'
00123         QtMocHelpers::SlotData<void(bool)>(22, 2, QMC::AccessPrivate, QMetaType::Void, {{
00124             { QMetaType::Bool, 23 },
00125         }}),
00126         // Slot 'toggleRangeOptions'
00127         QtMocHelpers::SlotData<void(bool)>(24, 2, QMC::AccessPrivate, QMetaType::Void, {{
00128             { QMetaType::Bool, 23 },
00129         }}),
00130         // Slot 'showTopologyView'
00131         QtMocHelpers::SlotData<void()>(25, 2, QMC::AccessPrivate, QMetaType::Void),
00132         // Slot 'showStatisticsView'
00133         QtMocHelpers::SlotData<void()>(26, 2, QMC::AccessPrivate, QMetaType::Void),
00134         // Slot 'showHistoryView'
00135         QtMocHelpers::SlotData<void()>(27, 2, QMC::AccessPrivate, QMetaType::Void),
00136         // Slot 'generateSecurityReport'
00137         QtMocHelpers::SlotData<void()>(28, 2, QMC::AccessPrivate, QMetaType::Void),
00138         // Slot 'saveTopologyImage'
00139         QtMocHelpers::SlotData<void()>(29, 2, QMC::AccessPrivate, QMetaType::Void),
00140         // Slot 'toggleDarkMode'
00141         QtMocHelpers::SlotData<void(bool)>(30, 2, QMC::AccessPrivate, QMetaType::Void, {{
00142             { QMetaType::Bool, 31 },
00143         }}),
00144         // Slot 'compareScanResults'
00145         QtMocHelpers::SlotData<void()>(32, 2, QMC::AccessPrivate, QMetaType::Void),
00146         // Slot 'scheduleScan'
00147         QtMocHelpers::SlotData<void()>(33, 2, QMC::AccessPrivate, QMetaType::Void),
00148         // Slot 'saveHistoryToFile'
00149         QtMocHelpers::SlotData<void()>(34, 2, QMC::AccessPrivate, QMetaType::Void),
00150         // Slot 'loadHistoryFromFile'
00151         QtMocHelpers::SlotData<void()>(35, 2, QMC::AccessPrivate, QMetaType::Void),
00152         // Slot 'updateNetworkTopology'
00153         QtMocHelpers::SlotData<void()>(36, 2, QMC::AccessPrivate, QMetaType::Void),
00154         // Slot 'refreshTopology'
00155         QtMocHelpers::SlotData<void()>(37, 2, QMC::AccessPrivate, QMetaType::Void),
00156         // Slot 'filterResults'
00157         QtMocHelpers::SlotData<void()>(38, 2, QMC::AccessPrivate, QMetaType::Void),
00158         // Slot 'clearFilters'
00159         QtMocHelpers::SlotData<void()>(39, 2, QMC::AccessPrivate, QMetaType::Void),
00160         // Slot 'onThemeChanged'
00161         QtMocHelpers::SlotData<void()>(40, 2, QMC::AccessPrivate, QMetaType::Void),
00162     };
00163     QtMocHelpers::UintData qt_properties {
00164     };
00165     QtMocHelpers::UintData qt_enums {
00166     };
00167     return QtMocHelpers::metaObjectData<MainWindow,
      qt_meta_tag_ZN10MainWindowE_t>(QMC::MetaObjectFlag{}, qt_stringData,
00168             qt_methods, qt_properties, qt_enums);
00169 }
00170 Q_CONSTINIT const QMetaObject MainWindow::staticMetaObject = { {
00171     QMetaObject::SuperData::link<QMainWindow::staticMetaObject>(),
00172     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN10MainWindowE_t>.stringdata,
00173     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN10MainWindowE_t>.data,
00174     qt_static_metacall,
00175     nullptr,
00176     qt_staticMetaObjectRelocatingContent<qt_meta_tag_ZN10MainWindowE_t>.metaTypes,
00177     nullptr
00178 } };
00179
00180 void MainWindow::qt_static_metacall(QObject *_o, QMetaObject::Call _c, int _id, void **_a)
00181 {
00182     auto *_t = static_cast<MainWindow *>(_o);
00183     if (_c == QMetaObject::InvokeMetaMethod) {
00184         switch (_id) {
00185         case 0: _t->startScan(); break;
00186         case 1: _t->stopScan(); break;
00187         case 2: _t->onHostFound((*reinterpret_cast< std::add_pointer_t<HostInfo>>(_a[1]))); break;
00188         case 3: _t->onScanStarted(); break;
```

```
00189         case 4: _t->onScanFinished(); break;
00190         case 5: _t->onScanProgress((*reinterpret_cast< std::add_pointer_t<int>>(_a[1]))); break;
00191         case 6: _t->onScanError((*reinterpret_cast< std::add_pointer_t<QString>>(_a[1]))); break;
00192         case 7: _t->saveResults(); break;
00193         case 8: _t->clearResults(); break;
00194         case 9: _t->showSettings(); break;
00195         case 10: _t->applySettings(); break;
00196         case 11: _t->showAbout(); break;
00197         case 12: _t->showHostDetails((*reinterpret_cast<
      std::add_pointer_t<int>>(_a[1])),(*reinterpret_cast< std::add_pointer_t<int>>(_a[2]))); break;
00198         case 13: _t->exportToCSV(); break;
00199         case 14: _t->togglePortScanOptions((*reinterpret_cast< std::add_pointer_t<bool>>(_a[1])));
      break;
00200         case 15: _t->toggleRangeOptions((*reinterpret_cast< std::add_pointer_t<bool>>(_a[1]))); break;
00201         case 16: _t->showTopologyView(); break;
00202         case 17: _t->showStatisticsView(); break;
00203         case 18: _t->showHistoryView(); break;
00204         case 19: _t->generateSecurityReport(); break;
00205         case 20: _t->saveTopologyImage(); break;
00206         case 21: _t->toggleDarkMode((*reinterpret_cast< std::add_pointer_t<bool>>(_a[1]))); break;
00207         case 22: _t->compareScanResults(); break;
00208         case 23: _t->scheduleScan(); break;
00209         case 24: _t->saveHistoryToFile(); break;
00210         case 25: _t->loadHistoryFromFile(); break;
00211         case 26: _t->updateNetworkTopology(); break;
00212         case 27: _t->refreshTopology(); break;
00213         case 28: _t->filterResults(); break;
00214         case 29: _t->clearFilters(); break;
00215         case 30: _t->onThemeChanged(); break;
00216         default: ;
00217         }
00218     }
00219 }
00220
00221 const QMetaObject *MainWindow::metaObject() const
00222 {
00223     return QObject::d_ptr->metaObject ? QObject::d_ptr->dynamicMetaObject() : &staticMetaObject;
00224 }
00225
00226 void *MainWindow::qt_metacast(const char *_clname)
00227 {
00228     if (!_clname) return nullptr;
00229     if (!strcmp(_clname, qt_staticMetaObjectStaticContent<qt_meta_tag_ZN10MainWindowE_t>.strings))
00230         return static_cast<void*>(this);
00231     return QMainWindow::qt_metacast(_clname);
00232 }
00233
00234 int MainWindow::qt_metacall(QMetaObject::Call _c, int _id, void **_a)
00235 {
00236     _id = QMainWindow::qt_metacall(_c, _id, _a);
00237     if (_id < 0)
00238         return _id;
00239     if (_c == QMetaObject::InvokeMetaMethod) {
00240         if (_id < 31)
00241             qt_static_metacall(this, _c, _id, _a);
00242         _id -= 31;
00243     }
00244     if (_c == QMetaObject::RegisterMethodArgumentMetaType) {
00245         if (_id < 31)
00246             *reinterpret_cast<QMetaType *>(_a[0]) = QMetaType();
00247         _id -= 31;
00248     }
00249     return _id;
00250 }
00251 QT_WARNING_POP
```

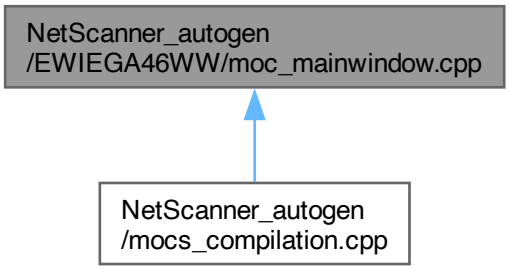
9.20 NetScanner_autogen/EWIEGA46WW/moc_mainwindow.cpp 文件参考

```
#include "../../../mainwindow.h"
#include <QtGui/qtextcursor.h>
#include <QtCore/qmetatype.h>
#include <QtCore/qtmochelpers.h>
#include <memory>
#include <QtCore/qxptype_traits.h>
```

moc_mainwindow.cpp 的引用(Include)关系图:



此图展示该文件被哪些文件直接或间接地引用了:



类

- struct QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN10MainWindowE_t

命名空间

- namespace QT_WARNING_DISABLE_DEPRECATED

宏定义

- #define Q_CONSTINIT

9.20.1 宏定义说明

9.20.1.1 Q_CONSTINIT

```
#define Q_CONSTINIT
```

9.21 moc_mainwindow.cpp

浏览该文件的文档.

```
00001 /****************************************************************************
00002 ** Meta object code from reading C++ file 'mainwindow.h'
00003 **
00004 ** Created by: The Qt Meta Object Compiler version 69 (Qt 6.9.0)
00005 **
00006 ** WARNING! All changes made in this file will be lost!
00007 *****************************************************************************/
00008
00009 #include "../../../mainwindow.h"
00010 #include <QtGui/qtextcursor.h>
00011 #include <QtCore/qmetatype.h>
00012
00013 #include <QtCore/qtmochelpers.h>
00014
00015 #include <memory>
00016
00017
00018 #include <QtCore/qxptype_traits.h>
00019 #if !defined(Q_MOC_OUTPUT_REVISION)
00020 #error "The header file 'mainwindow.h' doesn't include <QObject>."
00021 #elif Q_MOC_OUTPUT_REVISION != 69
00022 #error "This file was generated using the moc from 6.9.0. It"
00023 #error "cannot be used with the include files from this version of Qt."
00024 #error "(The moc has changed too much.)"
00025 #endif
00026
00027 #ifndef Q_CONSTINIT
00028 #define Q_CONSTINIT
00029 #endif
00030
00031 QT_WARNING_PUSH
00032 QT_WARNING_DISABLE_DEPRECATED
00033 QT_WARNING_DISABLE_GCC("-Wuseless-cast")
00034 namespace {
00035 struct qt_meta_tag_ZN10MainWindowE_t {};
00036 } // unnamed namespace
00037
00038 template <> constexpr inline auto MainWindow::qt_create_metaobjectdata<qt_meta_tag_ZN10MainWindowE_t>()
00039 {
00040     namespace QMC = QtMocConstants;
00041     QtMocHelpers::StringRefStorage qt_stringData {
00042         "MainWindow",
00043         "startScan",
00044         "",
00045         "stopScan",
00046         "onHostFound",
00047         "HostInfo",
00048         "host",
00049         "onScanStarted",
00050         "onScanFinished",
00051         "onScanProgress",
00052         "progress",
00053         "onScanError",
00054         "errorMessage",
00055         "saveResults",
00056         "clearResults",
00057         "showSettings",
00058         "applySettings",
00059         "showAbout",
00060         "showHostDetails",
00061         "row",
00062         "column",
00063         "exportToCSV",
00064         "togglePortScanOptions",
00065         "checked",
00066         "toggleRangeOptions",
00067         "showTopologyView",
00068         "showStatisticsView",
00069         "showHistoryView",
00070         "generateSecurityReport",
00071         "saveTopologyImage",
00072         "toggleDarkMode",
00073         "enable",
00074         "compareScanResults",
00075         "scheduleScan",
00076         "saveHistoryToFile",
00077         "loadHistoryFromFile",
00078         "updateNetworkTopology",
00079         "refreshTopology",
00080         "filterResults",
00081         "clearFilters",
00082         "onThemeChanged"
```

```
00083     };
00084
00085     QtMocHelpers::UintData qt_methods {
00086         // Slot 'startScan'
00087         QtMocHelpers::SlotData<void()>(1, 2, QMC::AccessPrivate, QMetaType::Void),
00088         // Slot 'stopScan'
00089         QtMocHelpers::SlotData<void()>(3, 2, QMC::AccessPrivate, QMetaType::Void),
00090         // Slot 'onHostFound'
00091         QtMocHelpers::SlotData<void(const HostInfo &)>(4, 2, QMC::AccessPrivate, QMetaType::Void, {{
00092             { 0x80000000 | 5, 6 },
00093         }}),
00094         // Slot 'onScanStarted'
00095         QtMocHelpers::SlotData<void()>(7, 2, QMC::AccessPrivate, QMetaType::Void),
00096         // Slot 'onScanFinished'
00097         QtMocHelpers::SlotData<void()>(8, 2, QMC::AccessPrivate, QMetaType::Void),
00098         // Slot 'onScanProgress'
00099         QtMocHelpers::SlotData<void(int)>(9, 2, QMC::AccessPrivate, QMetaType::Void, {{
00100             { QMetaType::Int, 10 },
00101         }}),
00102         // Slot 'onScanError'
00103         QtMocHelpers::SlotData<void(const QString &)>(11, 2, QMC::AccessPrivate, QMetaType::Void, {{
00104             { QMetaType::QString, 12 },
00105         }}),
00106         // Slot 'saveResults'
00107         QtMocHelpers::SlotData<void()>(13, 2, QMC::AccessPrivate, QMetaType::Void),
00108         // Slot 'clearResults'
00109         QtMocHelpers::SlotData<void()>(14, 2, QMC::AccessPrivate, QMetaType::Void),
00110         // Slot 'showSettings'
00111         QtMocHelpers::SlotData<void()>(15, 2, QMC::AccessPrivate, QMetaType::Void),
00112         // Slot 'applySettings'
00113         QtMocHelpers::SlotData<void()>(16, 2, QMC::AccessPrivate, QMetaType::Void),
00114         // Slot 'showAbout'
00115         QtMocHelpers::SlotData<void()>(17, 2, QMC::AccessPrivate, QMetaType::Void),
00116         // Slot 'showHostDetails'
00117         QtMocHelpers::SlotData<void(int, int)>(18, 2, QMC::AccessPrivate, QMetaType::Void, {{
00118             { QMetaType::Int, 19 }, { QMetaType::Int, 20 },
00119         }}),
00120         // Slot 'exportToCSV'
00121         QtMocHelpers::SlotData<void()>(21, 2, QMC::AccessPrivate, QMetaType::Void),
00122         // Slot 'togglePortScanOptions'
00123         QtMocHelpers::SlotData<void(bool)>(22, 2, QMC::AccessPrivate, QMetaType::Void, {{
00124             { QMetaType::Bool, 23 },
00125         }}),
00126         // Slot 'toggleRangeOptions'
00127         QtMocHelpers::SlotData<void(bool)>(24, 2, QMC::AccessPrivate, QMetaType::Void, {{
00128             { QMetaType::Bool, 23 },
00129         }}),
00130         // Slot 'showTopologyView'
00131         QtMocHelpers::SlotData<void()>(25, 2, QMC::AccessPrivate, QMetaType::Void),
00132         // Slot 'showStatisticsView'
00133         QtMocHelpers::SlotData<void()>(26, 2, QMC::AccessPrivate, QMetaType::Void),
00134         // Slot 'showHistoryView'
00135         QtMocHelpers::SlotData<void()>(27, 2, QMC::AccessPrivate, QMetaType::Void),
00136         // Slot 'generateSecurityReport'
00137         QtMocHelpers::SlotData<void()>(28, 2, QMC::AccessPrivate, QMetaType::Void),
00138         // Slot 'saveTopologyImage'
00139         QtMocHelpers::SlotData<void()>(29, 2, QMC::AccessPrivate, QMetaType::Void),
00140         // Slot 'toggleDarkMode'
00141         QtMocHelpers::SlotData<void(bool)>(30, 2, QMC::AccessPrivate, QMetaType::Void, {{
00142             { QMetaType::Bool, 31 },
00143         }}),
00144         // Slot 'compareScanResults'
00145         QtMocHelpers::SlotData<void()>(32, 2, QMC::AccessPrivate, QMetaType::Void),
00146         // Slot 'scheduleScan'
00147         QtMocHelpers::SlotData<void()>(33, 2, QMC::AccessPrivate, QMetaType::Void),
00148         // Slot 'saveHistoryToFile'
00149         QtMocHelpers::SlotData<void()>(34, 2, QMC::AccessPrivate, QMetaType::Void),
00150         // Slot 'loadHistoryFromFile'
00151         QtMocHelpers::SlotData<void()>(35, 2, QMC::AccessPrivate, QMetaType::Void),
00152         // Slot 'updateNetworkTopology'
00153         QtMocHelpers::SlotData<void()>(36, 2, QMC::AccessPrivate, QMetaType::Void),
00154         // Slot 'refreshTopology'
00155         QtMocHelpers::SlotData<void()>(37, 2, QMC::AccessPrivate, QMetaType::Void),
00156         // Slot 'filterResults'
00157         QtMocHelpers::SlotData<void()>(38, 2, QMC::AccessPrivate, QMetaType::Void),
00158         // Slot 'clearFilters'
00159         QtMocHelpers::SlotData<void()>(39, 2, QMC::AccessPrivate, QMetaType::Void),
00160         // Slot 'onThemeChanged'
00161         QtMocHelpers::SlotData<void()>(40, 2, QMC::AccessPrivate, QMetaType::Void),
00162     };
00163     QtMocHelpers::UintData qt_properties {
00164     };
00165     QtMocHelpers::UintData qt_enums {
00166     };
00167     return QtMocHelpers::metaObjectData<MainWindow,
      qt_meta_tag_ZN10MainWindowE_t>(QMC::MetaObjectFlag{}, qt_stringData,
00168             qt_methods, qt_properties, qt_enums);
```

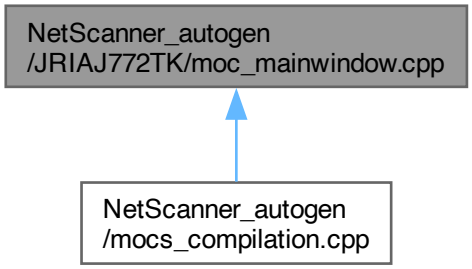
```
00169 }
00170 Q_CONSTINIT const QMetaObject MainWindow::staticMetaObject = { {
00171     QMetaObject::SuperData::link<QMainWindow::staticMetaObject>(),
00172     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN10MainWindowE_t>.stringdata,
00173     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN10MainWindowE_t>.data,
00174     qt_static_metacall,
00175     nullptr,
00176     qt_staticMetaObjectRelocatingContent<qt_meta_tag_ZN10MainWindowE_t>.metaTypes,
00177     nullptr
00178 } };
00179
00180 void MainWindow::qt_static_metacall(QObject *_o, QMetaObject::Call _c, int _id, void **_a)
00181 {
00182     auto *_t = static_cast<MainWindow *>(_o);
00183     if (_c == QMetaObject::InvokeMetaMethod) {
00184         switch (_id) {
00185         case 0: _t->startScan(); break;
00186         case 1: _t->stopScan(); break;
00187         case 2: _t->onHostFound((*reinterpret_cast< std::add_pointer_t<HostInfo>>(_a[1]))); break;
00188         case 3: _t->onScanStarted(); break;
00189         case 4: _t->onScanFinished(); break;
00190         case 5: _t->onScanProgress((*reinterpret_cast< std::add_pointer_t<int>>(_a[1]))); break;
00191         case 6: _t->onScanError((*reinterpret_cast< std::add_pointer_t<QString>>(_a[1]))); break;
00192         case 7: _t->saveResults(); break;
00193         case 8: _t->clearResults(); break;
00194         case 9: _t->showSettings(); break;
00195         case 10: _t->applySettings(); break;
00196         case 11: _t->showAbout(); break;
00197         case 12: _t->showHostDetails((*reinterpret_cast<
      std::add_pointer_t<int>>(_a[1])),(*reinterpret_cast< std::add_pointer_t<int>>(_a[2]))); break;
00198         case 13: _t->exportToCSV(); break;
00199         case 14: _t->togglePortScanOptions((*reinterpret_cast< std::add_pointer_t<bool>>(_a[1])));
      break;
00200         case 15: _t->toggleRangeOptions((*reinterpret_cast< std::add_pointer_t<bool>>(_a[1]))); break;
00201         case 16: _t->showTopologyView(); break;
00202         case 17: _t->showStatisticsView(); break;
00203         case 18: _t->showHistoryView(); break;
00204         case 19: _t->generateSecurityReport(); break;
00205         case 20: _t->saveTopologyImage(); break;
00206         case 21: _t->toggleDarkMode((*reinterpret_cast< std::add_pointer_t<bool>>(_a[1]))); break;
00207         case 22: _t->compareScanResults(); break;
00208         case 23: _t->scheduleScan(); break;
00209         case 24: _t->saveHistoryToFile(); break;
00210         case 25: _t->loadHistoryFromFile(); break;
00211         case 26: _t->updateNetworkTopology(); break;
00212         case 27: _t->refreshTopology(); break;
00213         case 28: _t->filterResults(); break;
00214         case 29: _t->clearFilters(); break;
00215         case 30: _t->onThemeChanged(); break;
00216         default: ;
00217         }
00218     }
00219 }
00220
00221 const QMetaObject *MainWindow::metaObject() const
00222 {
00223     return QObject::d_ptr->metaObject ? QObject::d_ptr->dynamicMetaObject() : &staticMetaObject;
00224 }
00225
00226 void *MainWindow::qt_metacast(const char *_clname)
00227 {
00228     if (!_clname) return nullptr;
00229     if (!strcmp(_clname, qt_staticMetaObjectStaticContent<qt_meta_tag_ZN10MainWindowE_t>.strings))
00230         return static_cast<void*>(this);
00231     return QMainWindow::qt_metacast(_clname);
00232 }
00233
00234 int MainWindow::qt_metacall(QMetaObject::Call _c, int _id, void **_a)
00235 {
00236     _id = QMainWindow::qt_metacall(_c, _id, _a);
00237     if (_id < 0)
00238         return _id;
00239     if (_c == QMetaObject::InvokeMetaMethod) {
00240         if (_id < 31)
00241             qt_static_metacall(this, _c, _id, _a);
00242         _id -= 31;
00243     }
00244     if (_c == QMetaObject::RegisterMethodArgumentMetaType) {
00245         if (_id < 31)
00246             *reinterpret_cast<QMetaType *>(_a[0]) = QMetaType();
00247         _id -= 31;
00248     }
00249     return _id;
00250 }
00251 QT_WARNING_POP
```

9.22 NetScanner_autogen/JRIAJ772TK/moc_mainwindow.cpp 文件参考

```
#include "../../mainwindow.h"
#include <QtGui/qtextcursor.h>
#include <QtCore/qmetatype.h>
#include <QtCore/qtmochelpers.h>
#include <memory>
#include <QtCore/qxptype_traits.h>
```

moc_mainwindow.cpp 的引用(Include)关系图:

此图展示该文件被哪些文件直接或间接地引用了:



类

- struct QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN10MainWindowE_t

命名空间

- namespace QT_WARNING_DISABLE_DEPRECATED

宏定义

- #define Q_CONSTINIT

9.22.1 宏定义说明

9.22.1.1 Q_CONSTINIT

```
#define Q_CONSTINIT
```

9.23 moc_mainwindow.cpp

浏览该文件的文档.

```
00001 /****************************************************************************
00002 ** Meta object code from reading C++ file 'mainwindow.h'
00003 **
00004 ** Created by: The Qt Meta Object Compiler version 69 (Qt 6.9.0)
00005 **
00006 ** WARNING! All changes made in this file will be lost!
00007 *****************************************************************************/
00008
00009 #include "../../mainwindow.h"
00010 #include <QtGui/qtextcursor.h>
00011 #include <QtCore/qmetatype.h>
00012
00013 #include <QtCore/qtmochelpers.h>
00014
00015 #include <memory>
00016
00017
00018 #include <QtCore/qxptype_traits.h>
00019 #if !defined(Q_MOC_OUTPUT_REVISION)
00020 #error "The header file 'mainwindow.h' doesn't include <QObject>."
00021 #elif Q_MOC_OUTPUT_REVISION != 69
00022 #error "This file was generated using the moc from 6.9.0. It"
00023 #error "cannot be used with the include files from this version of Qt."
00024 #error "(The moc has changed too much.)"
00025 #endif
00026
00027 #ifndef Q_CONSTINIT
00028 #define Q_CONSTINIT
00029 #endif
00030
00031 QT_WARNING_PUSH
00032 QT_WARNING_DISABLE_DEPRECATED
00033 QT_WARNING_DISABLE_GCC("-Wuseless-cast")
00034 namespace {
00035 struct qt_meta_tag_ZN10MainWindowE_t {};
00036 } // unnamed namespace
00037
00038 template <> constexpr inline auto MainWindow::qt_create_metaobjectdata<qt_meta_tag_ZN10MainWindowE_t>()
00039 {
00040     namespace QMC = QtMocConstants;
00041     QtMocHelpers::StringRefStorage qt_stringData {
00042         "MainWindow",
00043         "startScan",
00044         "",
00045         "stopScan",
00046         "onHostFound",
00047         "HostInfo",
00048         "host",
00049         "onScanStarted",
00050         "onScanFinished",
00051         "onScanProgress",
00052         "progress",
00053         "onScanError",
00054         "errorMessage",
00055         "saveResults",
00056         "clearResults",
00057         "showSettings",
00058         "applySettings",
00059         "showAbout",
00060         "showHostDetails",
00061         "row",
00062         "column",
00063         "exportToCSV",
00064         "togglePortScanOptions",
00065         "checked",
00066         "toggleRangeOptions",
00067         "showTopologyView",
00068         "showStatisticsView",
00069         "showHistoryView",
00070         "generateSecurityReport",
00071         "saveTopologyImage",
00072         "toggleDarkMode",
00073         "enable",
00074         "compareScanResults",
00075         "scheduleScan",
00076         "saveHistoryToFile",
00077         "loadHistoryFromFile",
00078         "updateNetworkTopology",
00079         "refreshTopology",
00080         "filterResults",
00081         "clearFilters",
00082         "onThemeChanged"
```

```
00083     };
00084
00085     QtMocHelpers::UintData qt_methods {
00086         // Slot 'startScan'
00087         QtMocHelpers::SlotData<void()>(1, 2, QMC::AccessPrivate, QMetaType::Void),
00088         // Slot 'stopScan'
00089         QtMocHelpers::SlotData<void()>(3, 2, QMC::AccessPrivate, QMetaType::Void),
00090         // Slot 'onHostFound'
00091         QtMocHelpers::SlotData<void(const HostInfo &)>(4, 2, QMC::AccessPrivate, QMetaType::Void, {{
00092             { 0x80000000 | 5, 6 },
00093         }}),
00094         // Slot 'onScanStarted'
00095         QtMocHelpers::SlotData<void()>(7, 2, QMC::AccessPrivate, QMetaType::Void),
00096         // Slot 'onScanFinished'
00097         QtMocHelpers::SlotData<void()>(8, 2, QMC::AccessPrivate, QMetaType::Void),
00098         // Slot 'onScanProgress'
00099         QtMocHelpers::SlotData<void(int)>(9, 2, QMC::AccessPrivate, QMetaType::Void, {{
00100             { QMetaType::Int, 10 },
00101         }}),
00102         // Slot 'onScanError'
00103         QtMocHelpers::SlotData<void(const QString &)>(11, 2, QMC::AccessPrivate, QMetaType::Void, {{
00104             { QMetaType::QString, 12 },
00105         }}),
00106         // Slot 'saveResults'
00107         QtMocHelpers::SlotData<void()>(13, 2, QMC::AccessPrivate, QMetaType::Void),
00108         // Slot 'clearResults'
00109         QtMocHelpers::SlotData<void()>(14, 2, QMC::AccessPrivate, QMetaType::Void),
00110         // Slot 'showSettings'
00111         QtMocHelpers::SlotData<void()>(15, 2, QMC::AccessPrivate, QMetaType::Void),
00112         // Slot 'applySettings'
00113         QtMocHelpers::SlotData<void()>(16, 2, QMC::AccessPrivate, QMetaType::Void),
00114         // Slot 'showAbout'
00115         QtMocHelpers::SlotData<void()>(17, 2, QMC::AccessPrivate, QMetaType::Void),
00116         // Slot 'showHostDetails'
00117         QtMocHelpers::SlotData<void(int, int)>(18, 2, QMC::AccessPrivate, QMetaType::Void, {{
00118             { QMetaType::Int, 19 }, { QMetaType::Int, 20 },
00119         }}),
00120         // Slot 'exportToCSV'
00121         QtMocHelpers::SlotData<void()>(21, 2, QMC::AccessPrivate, QMetaType::Void),
00122         // Slot 'togglePortScanOptions'
00123         QtMocHelpers::SlotData<void(bool)>(22, 2, QMC::AccessPrivate, QMetaType::Void, {{
00124             { QMetaType::Bool, 23 },
00125         }}),
00126         // Slot 'toggleRangeOptions'
00127         QtMocHelpers::SlotData<void(bool)>(24, 2, QMC::AccessPrivate, QMetaType::Void, {{
00128             { QMetaType::Bool, 23 },
00129         }}),
00130         // Slot 'showTopologyView'
00131         QtMocHelpers::SlotData<void()>(25, 2, QMC::AccessPrivate, QMetaType::Void),
00132         // Slot 'showStatisticsView'
00133         QtMocHelpers::SlotData<void()>(26, 2, QMC::AccessPrivate, QMetaType::Void),
00134         // Slot 'showHistoryView'
00135         QtMocHelpers::SlotData<void()>(27, 2, QMC::AccessPrivate, QMetaType::Void),
00136         // Slot 'generateSecurityReport'
00137         QtMocHelpers::SlotData<void()>(28, 2, QMC::AccessPrivate, QMetaType::Void),
00138         // Slot 'saveTopologyImage'
00139         QtMocHelpers::SlotData<void()>(29, 2, QMC::AccessPrivate, QMetaType::Void),
00140         // Slot 'toggleDarkMode'
00141         QtMocHelpers::SlotData<void(bool)>(30, 2, QMC::AccessPrivate, QMetaType::Void, {{
00142             { QMetaType::Bool, 31 },
00143         }}),
00144         // Slot 'compareScanResults'
00145         QtMocHelpers::SlotData<void()>(32, 2, QMC::AccessPrivate, QMetaType::Void),
00146         // Slot 'scheduleScan'
00147         QtMocHelpers::SlotData<void()>(33, 2, QMC::AccessPrivate, QMetaType::Void),
00148         // Slot 'saveHistoryToFile'
00149         QtMocHelpers::SlotData<void()>(34, 2, QMC::AccessPrivate, QMetaType::Void),
00150         // Slot 'loadHistoryFromFile'
00151         QtMocHelpers::SlotData<void()>(35, 2, QMC::AccessPrivate, QMetaType::Void),
00152         // Slot 'updateNetworkTopology'
00153         QtMocHelpers::SlotData<void()>(36, 2, QMC::AccessPrivate, QMetaType::Void),
00154         // Slot 'refreshTopology'
00155         QtMocHelpers::SlotData<void()>(37, 2, QMC::AccessPrivate, QMetaType::Void),
00156         // Slot 'filterResults'
00157         QtMocHelpers::SlotData<void()>(38, 2, QMC::AccessPrivate, QMetaType::Void),
00158         // Slot 'clearFilters'
00159         QtMocHelpers::SlotData<void()>(39, 2, QMC::AccessPrivate, QMetaType::Void),
00160         // Slot 'onThemeChanged'
00161         QtMocHelpers::SlotData<void()>(40, 2, QMC::AccessPrivate, QMetaType::Void),
00162     };
00163     QtMocHelpers::UintData qt_properties {
00164     };
00165     QtMocHelpers::UintData qt_enums {
00166     };
00167     return QtMocHelpers::metaObjectData<MainWindow,
      qt_meta_tag_ZN10MainWindowE_t>(QMC::MetaObjectFlag{}, qt_stringData,
00168             qt_methods, qt_properties, qt_enums);
```

```
00169 }
00170 Q_CONSTINIT const QMetaObject MainWindow::staticMetaObject = { {
00171     QMetaObject::SuperData::link<QMainWindow::staticMetaObject>(),
00172     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN10MainWindowE_t>.stringdata,
00173     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN10MainWindowE_t>.data,
00174     qt_static_metacall,
00175     nullptr,
00176     qt_staticMetaObjectRelocatingContent<qt_meta_tag_ZN10MainWindowE_t>.metaTypes,
00177     nullptr
00178 } };
00179
00180 void MainWindow::qt_static_metacall(QObject *_o, QMetaObject::Call _c, int _id, void **_a)
00181 {
00182     auto *_t = static_cast<MainWindow *>(_o);
00183     if (_c == QMetaObject::InvokeMetaMethod) {
00184         switch (_id) {
00185         case 0: _t->startScan(); break;
00186         case 1: _t->stopScan(); break;
00187         case 2: _t->onHostFound((*reinterpret_cast< std::add_pointer_t<HostInfo>>(_a[1]))); break;
00188         case 3: _t->onScanStarted(); break;
00189         case 4: _t->onScanFinished(); break;
00190         case 5: _t->onScanProgress((*reinterpret_cast< std::add_pointer_t<int>>(_a[1]))); break;
00191         case 6: _t->onScanError((*reinterpret_cast< std::add_pointer_t<QString>>(_a[1]))); break;
00192         case 7: _t->saveResults(); break;
00193         case 8: _t->clearResults(); break;
00194         case 9: _t->showSettings(); break;
00195         case 10: _t->applySettings(); break;
00196         case 11: _t->showAbout(); break;
00197         case 12: _t->showHostDetails((*reinterpret_cast<
      std::add_pointer_t<int>>(_a[1])),(*reinterpret_cast< std::add_pointer_t<int>>(_a[2]))); break;
00198         case 13: _t->exportToCSV(); break;
00199         case 14: _t->togglePortScanOptions((*reinterpret_cast< std::add_pointer_t<bool>>(_a[1])));
      break;
00200         case 15: _t->toggleRangeOptions((*reinterpret_cast< std::add_pointer_t<bool>>(_a[1]))); break;
00201         case 16: _t->showTopologyView(); break;
00202         case 17: _t->showStatisticsView(); break;
00203         case 18: _t->showHistoryView(); break;
00204         case 19: _t->generateSecurityReport(); break;
00205         case 20: _t->saveTopologyImage(); break;
00206         case 21: _t->toggleDarkMode((*reinterpret_cast< std::add_pointer_t<bool>>(_a[1]))); break;
00207         case 22: _t->compareScanResults(); break;
00208         case 23: _t->scheduleScan(); break;
00209         case 24: _t->saveHistoryToFile(); break;
00210         case 25: _t->loadHistoryFromFile(); break;
00211         case 26: _t->updateNetworkTopology(); break;
00212         case 27: _t->refreshTopology(); break;
00213         case 28: _t->filterResults(); break;
00214         case 29: _t->clearFilters(); break;
00215         case 30: _t->onThemeChanged(); break;
00216         default: ;
00217         }
00218     }
00219 }
00220
00221 const QMetaObject *MainWindow::metaObject() const
00222 {
00223     return QObject::d_ptr->metaObject ? QObject::d_ptr->dynamicMetaObject() : &staticMetaObject;
00224 }
00225
00226 void *MainWindow::qt_metacast(const char *_clname)
00227 {
00228     if (!_clname) return nullptr;
00229     if (!strcmp(_clname, qt_staticMetaObjectStaticContent<qt_meta_tag_ZN10MainWindowE_t>.strings))
00230         return static_cast<void*>(this);
00231     return QMainWindow::qt_metacast(_clname);
00232 }
00233
00234 int MainWindow::qt_metacall(QMetaObject::Call _c, int _id, void **_a)
00235 {
00236     _id = QMainWindow::qt_metacall(_c, _id, _a);
00237     if (_id < 0)
00238         return _id;
00239     if (_c == QMetaObject::InvokeMetaMethod) {
00240         if (_id < 31)
00241             qt_static_metacall(this, _c, _id, _a);
00242         _id -= 31;
00243     }
00244     if (_c == QMetaObject::RegisterMethodArgumentMetaType) {
00245         if (_id < 31)
00246             *reinterpret_cast<QMetaType *>(_a[0]) = QMetaType();
00247         _id -= 31;
00248     }
00249     return _id;
00250 }
00251 QT_WARNING_POP
```

9.24 build/NetScanner_autogen/EWIEGA46WW/moc_mainwindow.cpp.d 文件参考

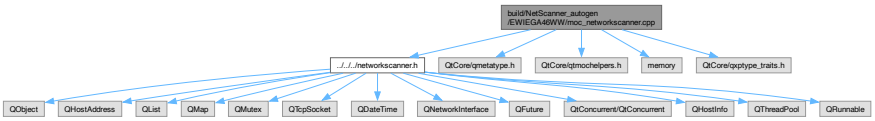
9.25 NetScanner_autogen/EWIEGA46WW/moc_mainwindow.cpp.d 文件参考

9.26 NetScanner_autogen/JRIAJ772TK/moc_mainwindow.cpp.d 文件参考

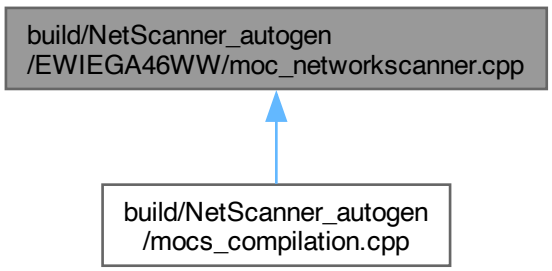
9.27 build/NetScanner_autogen/EWIEGA46WW/moc_networkscanner.cpp 文件参考

```
#include "../../../networkscanner.h"
#include <QtCore/qmetatype.h>
#include <QtCore/qtmochelpers.h>
#include <memory>
#include <QtCore/qxptype_traits.h>
```

moc_networkscanner.cpp 的引用(Include)关系图:



此图展示该文件被哪些文件直接或间接地引用了:



类

- struct QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN14NetworkScannerE_t

命名空间

- namespace QT_WARNING_DISABLE_DEPRECATED

宏定义

- #define Q_CONSTINIT

9.27.1 宏定义说明

9.27.1.1 Q_CONSTINIT

```
#define Q_CONSTINIT
```

9.28 moc_networkscanner.cpp

浏览该文件的文档.

```
00001 /****************************************************************************
00002 ** Meta object code from reading C++ file 'networkscanner.h'
00003 **
00004 ** Created by: The Qt Meta Object Compiler version 69 (Qt 6.9.0)
00005 **
00006 ** WARNING! All changes made in this file will be lost!
00007 *****************************************************************************/
00008
00009 #include "../../../networkscanner.h"
00010 #include <QtCore/qmetatype.h>
00011
00012 #include <QtCore/qtmochelpers.h>
00013
00014 #include <memory>
00015
00016
00017 #include <QtCore/qxptype_traits.h>
00018 #if !defined(Q_MOC_OUTPUT_REVISION)
00019 #error "The header file 'networkscanner.h' doesn't include <QObject>."
00020 #elif Q_MOC_OUTPUT_REVISION != 69
00021 #error "This file was generated using the moc from 6.9.0. It"
00022 #error "cannot be used with the include files from this version of Qt."
00023 #error "(The moc has changed too much.)"
00024 #endif
00025
00026 #ifndef Q_CONSTINIT
00027 #define Q_CONSTINIT
00028 #endif
00029
00030 QT_WARNING_PUSH
00031 QT_WARNING_DISABLE_DEPRECATED
00032 QT_WARNING_DISABLE_GCC("-Wuseless-cast")
00033 namespace {
00034 struct qt_meta_tag_ZN14NetworkScannerE_t {};
00035 } // unnamed namespace
00036
00037 template <> constexpr inline auto
      NetworkScanner::qt_create_metaobjectdata<qt_meta_tag_ZN14NetworkScannerE_t>()
00038 {
00039     namespace QMC = QtMocConstants;
00040     QtMocHelpers::StringRefStorage qt_stringData {
00041         "NetworkScanner",
00042         "hostFound",
00043         "",
00044         "HostInfo",
00045         "host",
00046         "scanStarted",
00047         "scanFinished",
00048         "scanProgress",
00049         "progress",
00050         "scanError",
00051         "errorMessage",
```

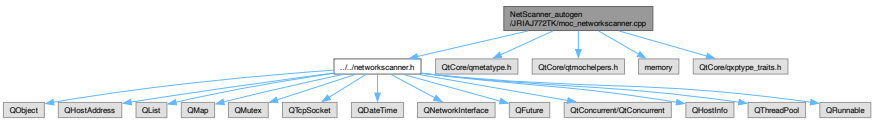
```
00052         "onScanTaskFinished",
00053         "hostInfo",
00054         "updateScanProgress",
00055         "onHostNameLookedUp",
00056         "QHostInfo"
00057     };
00058
00059     QtMocHelpers::UintData qt_methods {
00060         // Signal 'hostFound'
00061         QtMocHelpers::SignalData<void(const HostInfo &)>(1, 2, QMC::AccessPublic, QMetaType::Void, {{
00062             { 0x80000000 | 3, 4 },
00063         }}),
00064         // Signal 'scanStarted'
00065         QtMocHelpers::SignalData<void()>(5, 2, QMC::AccessPublic, QMetaType::Void),
00066         // Signal 'scanFinished'
00067         QtMocHelpers::SignalData<void()>(6, 2, QMC::AccessPublic, QMetaType::Void),
00068         // Signal 'scanProgress'
00069         QtMocHelpers::SignalData<void(int)>(7, 2, QMC::AccessPublic, QMetaType::Void, {{
00070             { QMetaType::Int, 8 },
00071         }}),
00072         // Signal 'scanError'
00073         QtMocHelpers::SignalData<void(const QString &)>(9, 2, QMC::AccessPublic, QMetaType::Void, {{
00074             { QMetaType::QString, 10 },
00075         }}),
00076         // Slot 'onScanTaskFinished'
00077         QtMocHelpers::SlotData<void(const HostInfo &)>(11, 2, QMC::AccessPublic, QMetaType::Void, {{
00078             { 0x80000000 | 3, 12 },
00079         }}),
00080         // Slot 'updateScanProgress'
00081         QtMocHelpers::SlotData<void()>(13, 2, QMC::AccessPublic, QMetaType::Void),
00082         // Slot 'onHostNameLookedUp'
00083         QtMocHelpers::SlotData<void(const QHostInfo &)>(14, 2, QMC::AccessPublic, QMetaType::Void, {{
00084             { 0x80000000 | 15, 12 },
00085         }}),
00086     };
00087     QtMocHelpers::UintData qt_properties {
00088     };
00089     QtMocHelpers::UintData qt_enums {
00090     };
00091     return QtMocHelpers::metaObjectData<NetworkScanner,
    qt_meta_tag_ZN14NetworkScannerE_t>(QMC::MetaObjectFlag{}, qt_stringData,
00092             qt_methods, qt_properties, qt_enums);
00093 }
00094 Q_CONSTINIT const QMetaObject NetworkScanner::staticMetaObject = { {
00095     QMetaObject::SuperData::link<QObject::staticMetaObject>(),
00096     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN14NetworkScannerE_t>.stringdata,
00097     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN14NetworkScannerE_t>.data,
00098     qt_static_metacall,
00099     nullptr,
00100     qt_staticMetaObjectRelocatingContent<qt_meta_tag_ZN14NetworkScannerE_t>.metaTypes,
00101     nullptr
00102 } };
00103
00104 void NetworkScanner::qt_static_metacall(QObject *_o, QMetaObject::Call _c, int _id, void **_a)
00105 {
00106     auto *_t = static_cast<NetworkScanner *>(_o);
00107     if (_c == QMetaObject::InvokeMetaMethod) {
00108         switch (_id) {
00109         case 0: _t->hostFound((*reinterpret_cast< std::add_pointer_t<HostInfo>>(_a[1]))); break;
00110         case 1: _t->scanStarted(); break;
00111         case 2: _t->scanFinished(); break;
00112         case 3: _t->scanProgress((*reinterpret_cast< std::add_pointer_t<int>>(_a[1]))); break;
00113         case 4: _t->scanError((*reinterpret_cast< std::add_pointer_t<QString>>(_a[1]))); break;
00114         case 5: _t->onScanTaskFinished((*reinterpret_cast< std::add_pointer_t<HostInfo>>(_a[1])));
    break;
00115         case 6: _t->updateScanProgress(); break;
00116         case 7: _t->onHostNameLookedUp((*reinterpret_cast< std::add_pointer_t<QHostInfo>>(_a[1])));
    break;
00117         default: ;
00118         }
00119     }
00120     if (_c == QMetaObject::RegisterMethodArgumentMetaType) {
00121         switch (_id) {
00122         default: *reinterpret_cast<QMetaType *>(_a[0]) = QMetaType(); break;
00123         case 7:
00124             switch (*reinterpret_cast<int*>(_a[1])) {
00125             default: *reinterpret_cast<QMetaType *>(_a[0]) = QMetaType(); break;
00126             case 0:
00127                 *reinterpret_cast<QMetaType *>(_a[0]) = QMetaType::fromType< QHostInfo >(); break;
00128             }
00129             break;
00130         }
00131     }
00132     if (_c == QMetaObject::IndexOfMethod) {
00133         if (QtMocHelpers::indexOfMethod<void (NetworkScanner::*)(const HostInfo & )>(_a,
    &NetworkScanner::hostFound, 0))
00134             return;
```

```
00135         if (QtMocHelpers::indexOfMethod<void (NetworkScanner::*)()>(_a, &NetworkScanner::scanStarted,
      1))
00136             return;
00137         if (QtMocHelpers::indexOfMethod<void (NetworkScanner::*)()>(_a, &NetworkScanner::scanFinished,
      2))
00138             return;
00139         if (QtMocHelpers::indexOfMethod<void (NetworkScanner::*)(int )>(_a,
      &NetworkScanner::scanProgress, 3))
00140             return;
00141         if (QtMocHelpers::indexOfMethod<void (NetworkScanner::*)(const QString & )>(_a,
      &NetworkScanner::scanError, 4))
00142             return;
00143     }
00144 }
00145
00146 const QMetaObject *NetworkScanner::metaObject() const
00147 {
00148     return QObject::d_ptr->metaObject ? QObject::d_ptr->dynamicMetaObject() : &staticMetaObject;
00149 }
00150
00151 void *NetworkScanner::qt_metacast(const char *_clname)
00152 {
00153     if (!_clname) return nullptr;
00154     if (!strcmp(_clname, qt_staticMetaObjectStaticContent<qt_meta_tag_ZN14NetworkScannerE_t>.strings))
00155         return static_cast<void*>(this);
00156     return QObject::qt_metacast(_clname);
00157 }
00158
00159 int NetworkScanner::qt_metacall(QMetaObject::Call _c, int _id, void **_a)
00160 {
00161     _id = QObject::qt_metacall(_c, _id, _a);
00162     if (_id < 0)
00163         return _id;
00164     if (_c == QMetaObject::InvokeMetaMethod) {
00165         if (_id < 8)
00166             qt_static_metacall(this, _c, _id, _a);
00167         _id -= 8;
00168     }
00169     if (_c == QMetaObject::RegisterMethodArgumentMetaType) {
00170         if (_id < 8)
00171             qt_static_metacall(this, _c, _id, _a);
00172         _id -= 8;
00173     }
00174     return _id;
00175 }
00176
00177 // SIGNAL 0
00178 void NetworkScanner::hostFound(const HostInfo & _t1)
00179 {
00180     QMetaObject::activate<void>(this, &staticMetaObject, 0, nullptr, _t1);
00181 }
00182
00183 // SIGNAL 1
00184 void NetworkScanner::scanStarted()
00185 {
00186     QMetaObject::activate(this, &staticMetaObject, 1, nullptr);
00187 }
00188
00189 // SIGNAL 2
00190 void NetworkScanner::scanFinished()
00191 {
00192     QMetaObject::activate(this, &staticMetaObject, 2, nullptr);
00193 }
00194
00195 // SIGNAL 3
00196 void NetworkScanner::scanProgress(int _t1)
00197 {
00198     QMetaObject::activate<void>(this, &staticMetaObject, 3, nullptr, _t1);
00199 }
00200
00201 // SIGNAL 4
00202 void NetworkScanner::scanError(const QString & _t1)
00203 {
00204     QMetaObject::activate<void>(this, &staticMetaObject, 4, nullptr, _t1);
00205 }
00206 QT_WARNING_POP
```

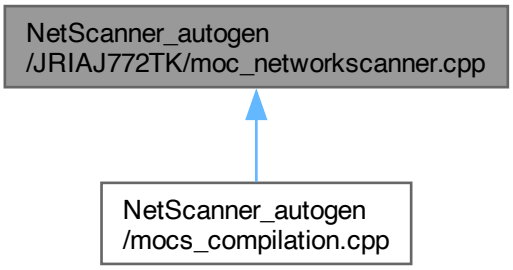
9.29 NetScanner_autogen/JRIAJ772TK/moc_networkscanner.cpp 文件参考

```
#include "../../networkscanner.h"
#include <QtCore/qmetatype.h>
#include <QtCore/qtmochelpers.h>
#include <memory>
#include <QtCore/qxptype_traits.h>
```

moc_networkscanner.cpp 的引用(Include)关系图:



此图展示该文件被哪些文件直接或间接地引用了:



类

- struct QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN14NetworkScannerE_t

命名空间

- namespace QT_WARNING_DISABLE_DEPRECATED

宏定义

- #define Q_CONSTINIT

9.29.1 宏定义说明

9.29.1.1 Q_CONSTINIT

```
#define Q_CONSTINIT
```

9.30 moc_networkscanner.cpp

浏览该文件的文档.

```
00001 /****************************************************************************
00002 ** Meta object code from reading C++ file 'networkscanner.h'
00003 **
00004 ** Created by: The Qt Meta Object Compiler version 69 (Qt 6.9.0)
00005 **
00006 ** WARNING! All changes made in this file will be lost!
00007 *****************************************************************************/
00008
00009 #include "../../networkscanner.h"
00010 #include <QtCore/qmetatype.h>
00011
00012 #include <QtCore/qtmochelpers.h>
00013
00014 #include <memory>
00015
00016
00017 #include <QtCore/qxptype_traits.h>
00018 #if !defined(Q_MOC_OUTPUT_REVISION)
00019 #error "The header file 'networkscanner.h' doesn't include <QObject>."
00020 #elif Q_MOC_OUTPUT_REVISION != 69
00021 #error "This file was generated using the moc from 6.9.0. It"
00022 #error "cannot be used with the include files from this version of Qt."
00023 #error "(The moc has changed too much.)"
00024 #endif
00025
00026 #ifndef Q_CONSTINIT
00027 #define Q_CONSTINIT
00028 #endif
00029
00030 QT_WARNING_PUSH
00031 QT_WARNING_DISABLE_DEPRECATED
00032 QT_WARNING_DISABLE_GCC("-Wuseless-cast")
00033 namespace {
00034 struct qt_meta_tag_ZN14NetworkScannerE_t {};
00035 } // unnamed namespace
00036
00037 template <> constexpr inline auto
      NetworkScanner::qt_create_metaobjectdata<qt_meta_tag_ZN14NetworkScannerE_t>()
00038 {
00039     namespace QMC = QtMocConstants;
00040     QtMocHelpers::StringRefStorage qt_stringData {
00041         "NetworkScanner",
00042         "scanStarted",
00043         "",
00044         "scanFinished",
00045         "hostFound",
00046         "HostInfo",
00047         "host",
00048         "scanProgress",
00049         "progress",
00050         "scanError",
00051         "errorMessage",
00052         "processScanResults"
00053     };
00054
00055     QtMocHelpers::UintData qt_methods {
00056         // Signal 'scanStarted'
00057         QtMocHelpers::SignalData<void()>(1, 2, QMC::AccessPublic, QMetaType::Void),
00058         // Signal 'scanFinished'
00059         QtMocHelpers::SignalData<void()>(3, 2, QMC::AccessPublic, QMetaType::Void),
00060         // Signal 'hostFound'
00061         QtMocHelpers::SignalData<void(const HostInfo &)>(4, 2, QMC::AccessPublic, QMetaType::Void, {{
00062             { 0x80000000 | 5, 6 },
00063         }}),
00064         // Signal 'scanProgress'
00065         QtMocHelpers::SignalData<void(int)>(7, 2, QMC::AccessPublic, QMetaType::Void, {{
00066             { QMetaType::Int, 8 },
00067         }}),
00068         // Signal 'scanError'
00069         QtMocHelpers::SignalData<void(const QString &)>(9, 2, QMC::AccessPublic, QMetaType::Void, {{
00070             { QMetaType::QString, 10 },
00071         }}),
00072         // Slot 'processScanResults'
00073         QtMocHelpers::SlotData<void()>(11, 2, QMC::AccessPrivate, QMetaType::Void),
00074     };
00075     QtMocHelpers::UintData qt_properties {
00076     };
00077     QtMocHelpers::UintData qt_enums {
00078     };
00079     return QtMocHelpers::metaObjectData<NetworkScanner,
      qt_meta_tag_ZN14NetworkScannerE_t>(QMC::MetaObjectFlag{}, qt_stringData,
00080             qt_methods, qt_properties, qt_enums);
```

```
00081 }
00082 Q_CONSTINIT const QMetaObject NetworkScanner::staticMetaObject = { {
00083     QMetaObject::SuperData::link<QObject::staticMetaObject>(),
00084     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN14NetworkScannerE_t>.stringdata,
00085     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN14NetworkScannerE_t>.data,
00086     qt_static_metacall,
00087     nullptr,
00088     qt_staticMetaObjectRelocatingContent<qt_meta_tag_ZN14NetworkScannerE_t>.metaTypes,
00089     nullptr
00090 } };
00091
00092 void NetworkScanner::qt_static_metacall(QObject *_o, QMetaObject::Call _c, int _id, void **_a)
00093 {
00094     auto *_t = static_cast<NetworkScanner *>(_o);
00095     if (_c == QMetaObject::InvokeMetaMethod) {
00096         switch (_id) {
00097         case 0: _t->scanStarted(); break;
00098         case 1: _t->scanFinished(); break;
00099         case 2: _t->hostFound((*reinterpret_cast< std::add_pointer_t<HostInfo>>(_a[1]))); break;
00100         case 3: _t->scanProgress((*reinterpret_cast< std::add_pointer_t<int>>(_a[1]))); break;
00101         case 4: _t->scanError((*reinterpret_cast< std::add_pointer_t<QString>>(_a[1]))); break;
00102         case 5: _t->processScanResults(); break;
00103         default: ;
00104         }
00105     }
00106     if (_c == QMetaObject::IndexOfMethod) {
00107         if (QtMocHelpers::indexOfMethod<void (NetworkScanner::*)()>(_a, &NetworkScanner::scanStarted,
      0))
00108             return;
00109         if (QtMocHelpers::indexOfMethod<void (NetworkScanner::*)()>(_a, &NetworkScanner::scanFinished,
      1))
00110             return;
00111         if (QtMocHelpers::indexOfMethod<void (NetworkScanner::*)(const HostInfo & )>(_a,
      &NetworkScanner::hostFound, 2))
00112             return;
00113         if (QtMocHelpers::indexOfMethod<void (NetworkScanner::*)(int )>(_a,
      &NetworkScanner::scanProgress, 3))
00114             return;
00115         if (QtMocHelpers::indexOfMethod<void (NetworkScanner::*)(const QString & )>(_a,
      &NetworkScanner::scanError, 4))
00116             return;
00117     }
00118 }
00119
00120 const QMetaObject *NetworkScanner::metaObject() const
00121 {
00122     return QObject::d_ptr->metaObject ? QObject::d_ptr->dynamicMetaObject() : &staticMetaObject;
00123 }
00124
00125 void *NetworkScanner::qt_metacast(const char *_clname)
00126 {
00127     if (!_clname) return nullptr;
00128     if (!strcmp(_clname, qt_staticMetaObjectStaticContent<qt_meta_tag_ZN14NetworkScannerE_t>.strings))
00129         return static_cast<void*>(this);
00130     return QObject::qt_metacast(_clname);
00131 }
00132
00133 int NetworkScanner::qt_metacall(QMetaObject::Call _c, int _id, void **_a)
00134 {
00135     _id = QObject::qt_metacall(_c, _id, _a);
00136     if (_id < 0)
00137         return _id;
00138     if (_c == QMetaObject::InvokeMetaMethod) {
00139         if (_id < 6)
00140             qt_static_metacall(this, _c, _id, _a);
00141         _id -= 6;
00142     }
00143     if (_c == QMetaObject::RegisterMethodArgumentMetaType) {
00144         if (_id < 6)
00145             *reinterpret_cast<QMetaType *>(_a[0]) = QMetaType();
00146         _id -= 6;
00147     }
00148     return _id;
00149 }
00150
00151 // SIGNAL 0
00152 void NetworkScanner::scanStarted()
00153 {
00154     QMetaObject::activate(this, &staticMetaObject, 0, nullptr);
00155 }
00156
00157 // SIGNAL 1
00158 void NetworkScanner::scanFinished()
00159 {
00160     QMetaObject::activate(this, &staticMetaObject, 1, nullptr);
00161 }
00162
```

```
00163 // SIGNAL 2
00164 void NetworkScanner::hostFound(const HostInfo & _t1)
00165 {
00166     QMetaObject::activate<void>(this, &staticMetaObject, 2, nullptr, _t1);
00167 }
00168
00169 // SIGNAL 3
00170 void NetworkScanner::scanProgress(int _t1)
00171 {
00172     QMetaObject::activate<void>(this, &staticMetaObject, 3, nullptr, _t1);
00173 }
00174
00175 // SIGNAL 4
00176 void NetworkScanner::scanError(const QString & _t1)
00177 {
00178     QMetaObject::activate<void>(this, &staticMetaObject, 4, nullptr, _t1);
00179 }
00180 QT_WARNING_POP
```

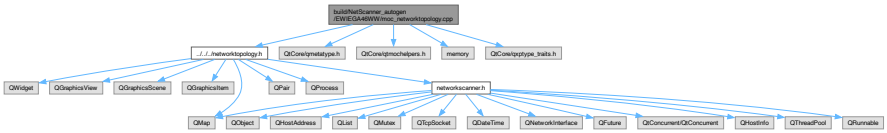
9.31 build/NetScanner_autogen/EWIEGA46WW/moc_networkscanner.cpp.d 文件参考

9.32 NetScanner_autogen/JRIAJ772TK/moc_networkscanner.cpp.d 文件参考

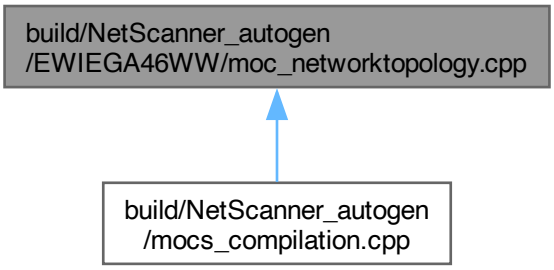
9.33 build/NetScanner_autogen/EWIEGA46WW/moc_networktopology.cpp 文件参考

```
#include "../../../networktopology.h"
#include <QtCore/qmetatype.h>
#include <QtCore/qtmochelpers.h>
#include <memory>
#include <QtCore/qxptype_traits.h>
```

moc_networktopology.cpp 的引用(Include)关系图:



此图展示该文件被哪些文件直接或间接地引用了:



类

- struct QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN19NetworkTopologyViewE_t

命名空间

- namespace QT_WARNING_DISABLE_DEPRECATED

宏定义

- #define Q_CONSTINIT

9.33.1 宏定义说明

9.33.1.1 Q_CONSTINIT

```
#define Q_CONSTINIT
```

9.34 moc_networktopology.cpp

浏览该文件的文档.

```
00001 /****************************************************************************
00002 ** Meta object code from reading C++ file 'networktopology.h'
00003 **
00004 ** Created by: The Qt Meta Object Compiler version 69 (Qt 6.9.0)
00005 **
00006 ** WARNING! All changes made in this file will be lost!
00007 *****************************************************************************/
00008
00009 #include "../../../networktopology.h"
00010 #include <QtCore/qmetatype.h>
00011
00012 #include <QtCore/qtmochelpers.h>
00013
00014 #include <memory>
00015
```

```
00016
00017 #include <QtCore/qxptype_traits.h>
00018 #if !defined(Q_MOC_OUTPUT_REVISION)
00019 #error "The header file 'networktopology.h' doesn't include <QObject>."
00020 #elif Q_MOC_OUTPUT_REVISION != 69
00021 #error "This file was generated using the moc from 6.9.0. It"
00022 #error "cannot be used with the include files from this version of Qt."
00023 #error "(The moc has changed too much.)"
00024 #endif
00025
00026 #ifndef Q_CONSTINIT
00027 #define Q_CONSTINIT
00028 #endif
00029
00030 QT_WARNING_PUSH
00031 QT_WARNING_DISABLE_DEPRECATED
00032 QT_WARNING_DISABLE_GCC("-Wuseless-cast")
00033 namespace {
00034 struct qt_meta_tag_ZN19NetworkTopologyViewE_t {};
00035 } // unnamed namespace
00036
00037 template <> constexpr inline auto
      NetworkTopologyView::qt_create_metaobjectdata<qt_meta_tag_ZN19NetworkTopologyViewE_t>()
00038 {
00039     namespace QMC = QtMocConstants;
00040     QtMocHelpers::StringRefStorage qt_stringData {
00041         "NetworkTopologyView",
00042         "nodeSelected",
00043         "",
00044         "HostInfo",
00045         "host"
00046     };
00047
00048     QtMocHelpers::UintData qt_methods {
00049         // Signal 'nodeSelected'
00050         QtMocHelpers::SignalData<void(const HostInfo &)>(1, 2, QMC::AccessPublic, QMetaType::Void, {{
00051             { 0x80000000 | 3, 4 },
00052         }}),
00053     };
00054     QtMocHelpers::UintData qt_properties {
00055     };
00056     QtMocHelpers::UintData qt_enums {
00057     };
00058     return QtMocHelpers::metaObjectData<NetworkTopologyView,
      qt_meta_tag_ZN19NetworkTopologyViewE_t>(QMC::MetaObjectFlag{}, qt_stringData,
00059             qt_methods, qt_properties, qt_enums);
00060 }
00061 Q_CONSTINIT const QMetaObject NetworkTopologyView::staticMetaObject = { {
00062     QMetaObject::SuperData::link<QGraphicsView::staticMetaObject>(),
00063     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN19NetworkTopologyViewE_t>.stringdata,
00064     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN19NetworkTopologyViewE_t>.data,
00065     qt_static_metacall,
00066     nullptr,
00067     qt_staticMetaObjectRelocatingContent<qt_meta_tag_ZN19NetworkTopologyViewE_t>.metaTypes,
00068     nullptr
00069 } };
00070
00071 void NetworkTopologyView::qt_static_metacall(QObject *_o, QMetaObject::Call _c, int _id, void **_a)
00072 {
00073     auto *_t = static_cast<NetworkTopologyView *>(_o);
00074     if (_c == QMetaObject::InvokeMetaMethod) {
00075         switch (_id) {
00076         case 0: _t->nodeSelected((*reinterpret_cast< std::add_pointer_t<HostInfo>>(_a[1]))); break;
00077         default: ;
00078         }
00079     }
00080     if (_c == QMetaObject::IndexOfMethod) {
00081         if (QtMocHelpers::indexOfMethod<void (NetworkTopologyView::*)(const HostInfo & )>(_a,
      &NetworkTopologyView::nodeSelected, 0))
00082             return;
00083     }
00084 }
00085
00086 const QMetaObject *NetworkTopologyView::metaObject() const
00087 {
00088     return QObject::d_ptr->metaObject ? QObject::d_ptr->dynamicMetaObject() : &staticMetaObject;
00089 }
00090
00091 void *NetworkTopologyView::qt_metacast(const char *_clname)
00092 {
00093     if (!_clname) return nullptr;
00094     if (!strcmp(_clname,
      qt_staticMetaObjectStaticContent<qt_meta_tag_ZN19NetworkTopologyViewE_t>.strings))
00095         return static_cast<void*>(this);
00096     return QGraphicsView::qt_metacast(_clname);
00097 }
00098
```

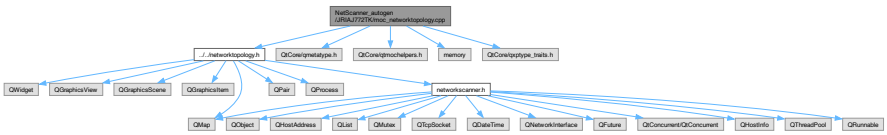
```
00099 int NetworkTopologyView::qt_metacall(QMetaObject::Call _c, int _id, void **_a)
00100 {
00101     _id = QGraphicsView::qt_metacall(_c, _id, _a);
00102     if (_id < 0)
00103         return _id;
00104     if (_c == QMetaObject::InvokeMetaMethod) {
00105         if (_id < 1)
00106             qt_static_metacall(this, _c, _id, _a);
00107         _id -= 1;
00108     }
00109     if (_c == QMetaObject::RegisterMethodArgumentMetaType) {
00110         if (_id < 1)
00111             *reinterpret_cast<QMetaType *>(_a[0]) = QMetaType();
00112         _id -= 1;
00113     }
00114     return _id;
00115 }
00116
00117 // SIGNAL 0
00118 void NetworkTopologyView::nodeSelected(const HostInfo & _t1)
00119 {
00120     QMetaObject::activate<void>(this, &staticMetaObject, 0, nullptr, _t1);
00121 }
00122 namespace {
00123 struct qt_meta_tag_ZN15NetworkTopologyE_t {};
00124 } // unnamed namespace
00125
00126 template <> constexpr inline auto
      NetworkTopology::qt_create_metaobjectdata<qt_meta_tag_ZN15NetworkTopologyE_t>()
00127 {
00128     namespace QMC = QtMocConstants;
00129     QtMocHelpers::StringRefStorage qt_stringData {
00130         "NetworkTopology",
00131         "deviceSelected",
00132         "",
00133         "HostInfo",
00134         "host"
00135     };
00136
00137     QtMocHelpers::UintData qt_methods {
00138         // Signal 'deviceSelected'
00139         QtMocHelpers::SignalData<void(const HostInfo &)>(1, 2, QMC::AccessPublic, QMetaType::Void, {{
00140             { 0x80000000 | 3, 4 },
00141         }}),
00142     };
00143     QtMocHelpers::UintData qt_properties {
00144     };
00145     QtMocHelpers::UintData qt_enums {
00146     };
00147     return QtMocHelpers::metaObjectData<NetworkTopology,
      qt_meta_tag_ZN15NetworkTopologyE_t>(QMC::MetaObjectFlag{}, qt_stringData,
00148             qt_methods, qt_properties, qt_enums);
00149 }
00150 Q_CONSTINIT const QMetaObject NetworkTopology::staticMetaObject = { {
00151     QMetaObject::SuperData::link<QWidget::staticMetaObject>(),
00152     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN15NetworkTopologyE_t>.stringdata,
00153     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN15NetworkTopologyE_t>.data,
00154     qt_static_metacall,
00155     nullptr,
00156     qt_staticMetaObjectRelocatingContent<qt_meta_tag_ZN15NetworkTopologyE_t>.metaTypes,
00157     nullptr
00158 } };
00159
00160 void NetworkTopology::qt_static_metacall(QObject *_o, QMetaObject::Call _c, int _id, void **_a)
00161 {
00162     auto *_t = static_cast<NetworkTopology *>(_o);
00163     if (_c == QMetaObject::InvokeMetaMethod) {
00164         switch (_id) {
00165         case 0: _t->deviceSelected((*reinterpret_cast< std::add_pointer_t<HostInfo>>(_a[1]))); break;
00166         default: ;
00167         }
00168     }
00169     if (_c == QMetaObject::IndexOfMethod) {
00170         if (QtMocHelpers::indexOfMethod<void (NetworkTopology::*)(const HostInfo & )>(_a,
      &NetworkTopology::deviceSelected, 0))
00171             return;
00172     }
00173 }
00174
00175 const QMetaObject *NetworkTopology::metaObject() const
00176 {
00177     return QObject::d_ptr->metaObject ? QObject::d_ptr->dynamicMetaObject() : &staticMetaObject;
00178 }
00179
00180 void *NetworkTopology::qt_metacast(const char *_clname)
00181 {
00182     if (!_clname) return nullptr;
```

```
00183     if (!strcmp(_clname, qt_staticMetaObjectStaticContent<qt_meta_tag_ZN15NetworkTopologyE_t>.strings))
00184         return static_cast<void*>(this);
00185     return QWidget::qt_metacast(_clname);
00186 }
00187
00188 int NetworkTopology::qt_metacall(QMetaObject::Call _c, int _id, void **_a)
00189 {
00190     _id = QWidget::qt_metacall(_c, _id, _a);
00191     if (_id < 0)
00192         return _id;
00193     if (_c == QMetaObject::InvokeMetaMethod) {
00194         if (_id < 1)
00195             qt_static_metacall(this, _c, _id, _a);
00196         _id -= 1;
00197     }
00198     if (_c == QMetaObject::RegisterMethodArgumentMetaType) {
00199         if (_id < 1)
00200             *reinterpret_cast<QMetaType *>(_a[0]) = QMetaType();
00201         _id -= 1;
00202     }
00203     return _id;
00204 }
00205
00206 // SIGNAL 0
00207 void NetworkTopology::deviceSelected(const HostInfo & _t1)
00208 {
00209     QMetaObject::activate<void>(this, &staticMetaObject, 0, nullptr, _t1);
00210 }
00211 QT_WARNING_POP
```

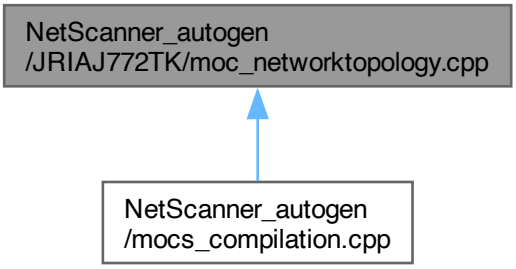
9.35 NetScanner_autogen/JRIAJ772TK/moc_networktopology.cpp 文件参考

```
#include "../../networktopology.h"
#include <QtCore/qmetatype.h>
#include <QtCore/qtmochelpers.h>
#include <memory>
#include <QtCore/qxptype_traits.h>
```

moc_networktopology.cpp 的引用(Include)关系图:



此图展示该文件被哪些文件直接或间接地引用了:



类

- struct QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN19NetworkTopologyViewE_t

命名空间

- namespace QT_WARNING_DISABLE_DEPRECATED

宏定义

- #define Q_CONSTINIT

9.35.1 宏定义说明

9.35.1.1 Q_CONSTINIT

```
#define Q_CONSTINIT
```

9.36 moc_networktopology.cpp

浏览该文件的文档.

```
00001 /****************************************************************************
00002 ** Meta object code from reading C++ file 'networktopology.h'
00003 **
00004 ** Created by: The Qt Meta Object Compiler version 69 (Qt 6.9.0)
00005 **
00006 ** WARNING! All changes made in this file will be lost!
00007 *****************************************************************************/
00008
00009 #include "../../networktopology.h"
00010 #include <QtCore/qmetatype.h>
00011
00012 #include <QtCore/qtmochelpers.h>
00013
00014 #include <memory>
00015
00016
00017 #include <QtCore/qxptype_traits.h>
00018 #if !defined(Q_MOC_OUTPUT_REVISION)
00019 #error "The header file 'networktopology.h' doesn't include <QObject>."
00020 #elif Q_MOC_OUTPUT_REVISION != 69
00021 #error "This file was generated using the moc from 6.9.0. It"
00022 #error "cannot be used with the include files from this version of Qt."
00023 #error "(The moc has changed too much.)"
00024 #endif
00025
00026 #ifndef Q_CONSTINIT
00027 #define Q_CONSTINIT
00028 #endif
00029
00030 QT_WARNING_PUSH
00031 QT_WARNING_DISABLE_DEPRECATED
00032 QT_WARNING_DISABLE_GCC("-Wuseless-cast")
00033 namespace {
00034 struct qt_meta_tag_ZN19NetworkTopologyViewE_t {};
00035 } // unnamed namespace
00036
00037 template <> constexpr inline auto
      NetworkTopologyView::qt_create_metaobjectdata<qt_meta_tag_ZN19NetworkTopologyViewE_t>()
00038 {
00039     namespace QMC = QtMocConstants;
00040     QtMocHelpers::StringRefStorage qt_stringData {
00041         "NetworkTopologyView",
00042         "nodeSelected",
00043         "",
```

```
00044         "HostInfo",
00045         "host"
00046     };
00047
00048     QtMocHelpers::UintData qt_methods {
00049         // Signal 'nodeSelected'
00050         QtMocHelpers::SignalData<void(const HostInfo &)>(1, 2, QMC::AccessPublic, QMetaType::Void, {{
00051             { 0x80000000 | 3, 4 },
00052         }}),
00053     };
00054     QtMocHelpers::UintData qt_properties {
00055     };
00056     QtMocHelpers::UintData qt_enums {
00057     };
00058     return QtMocHelpers::metaObjectData<NetworkTopologyView,
    qt_meta_tag_ZN19NetworkTopologyViewE_t>(QMC::MetaObjectFlag{}, qt_stringData,
00059             qt_methods, qt_properties, qt_enums);
00060 }
00061 Q_CONSTINIT const QMetaObject NetworkTopologyView::staticMetaObject = { {
00062     QMetaObject::SuperData::link<QGraphicsView::staticMetaObject>(),
00063     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN19NetworkTopologyViewE_t>.stringdata,
00064     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN19NetworkTopologyViewE_t>.data,
00065     qt_static_metacall,
00066     nullptr,
00067     qt_staticMetaObjectRelocatingContent<qt_meta_tag_ZN19NetworkTopologyViewE_t>.metaTypes,
00068     nullptr
00069 } };
00070
00071 void NetworkTopologyView::qt_static_metacall(QObject *_o, QMetaObject::Call _c, int _id, void **_a)
00072 {
00073     auto *_t = static_cast<NetworkTopologyView *>(_o);
00074     if (_c == QMetaObject::InvokeMetaMethod) {
00075         switch (_id) {
00076         case 0: _t->nodeSelected((*reinterpret_cast< std::add_pointer_t<HostInfo>>(_a[1]))); break;
00077         default: ;
00078         }
00079     }
00080     if (_c == QMetaObject::IndexOfMethod) {
00081         if (QtMocHelpers::indexOfMethod<void (NetworkTopologyView::*)(const HostInfo & )>(_a,
    &NetworkTopologyView::nodeSelected, 0))
00082             return;
00083     }
00084 }
00085
00086 const QMetaObject *NetworkTopologyView::metaObject() const
00087 {
00088     return QObject::d_ptr->metaObject ? QObject::d_ptr->dynamicMetaObject() : &staticMetaObject;
00089 }
00090
00091 void *NetworkTopologyView::qt_metacast(const char *_clname)
00092 {
00093     if (!_clname) return nullptr;
00094     if (!strcmp(_clname,
    qt_staticMetaObjectStaticContent<qt_meta_tag_ZN19NetworkTopologyViewE_t>.strings))
00095         return static_cast<void*>(this);
00096     return QGraphicsView::qt_metacast(_clname);
00097 }
00098
00099 int NetworkTopologyView::qt_metacall(QMetaObject::Call _c, int _id, void **_a)
00100 {
00101     _id = QGraphicsView::qt_metacall(_c, _id, _a);
00102     if (_id < 0)
00103         return _id;
00104     if (_c == QMetaObject::InvokeMetaMethod) {
00105         if (_id < 1)
00106             qt_static_metacall(this, _c, _id, _a);
00107         _id -= 1;
00108     }
00109     if (_c == QMetaObject::RegisterMethodArgumentMetaType) {
00110         if (_id < 1)
00111             *reinterpret_cast<QMetaType *>(_a[0]) = QMetaType();
00112         _id -= 1;
00113     }
00114     return _id;
00115 }
00116
00117 // SIGNAL 0
00118 void NetworkTopologyView::nodeSelected(const HostInfo & _t1)
00119 {
00120     QMetaObject::activate<void>(this, &staticMetaObject, 0, nullptr, _t1);
00121 }
00122 namespace {
00123 struct qt_meta_tag_ZN15NetworkTopologyE_t {};
00124 } // unnamed namespace
00125
00126 template <> constexpr inline auto
    NetworkTopology::qt_create_metaobjectdata<qt_meta_tag_ZN15NetworkTopologyE_t>()
```

```
00127 {
00128     namespace QMC = QtMocConstants;
00129     QtMocHelpers::StringRefStorage qt_stringData {
00130         "NetworkTopology",
00131         "deviceSelected",
00132         "",
00133         "HostInfo",
00134         "host"
00135     };
00136 
00137     QtMocHelpers::UintData qt_methods {
00138         // Signal 'deviceSelected'
00139         QtMocHelpers::SignalData<void(const HostInfo &)>(1, 2, QMC::AccessPublic, QMetaType::Void, {{
00140             { 0x80000000 | 3, 4 },
00141         }}),
00142     };
00143     QtMocHelpers::UintData qt_properties {
00144     };
00145     QtMocHelpers::UintData qt_enums {
00146     };
00147     return QtMocHelpers::metaObjectData<NetworkTopology,
    qt_meta_tag_ZN15NetworkTopologyE_t>(QMC::MetaObjectFlag{}, qt_stringData,
00148             qt_methods, qt_properties, qt_enums);
00149 }
00150 Q_CONSTINIT const QMetaObject NetworkTopology::staticMetaObject = { {
00151     QMetaObject::SuperData::link<QWidget::staticMetaObject>(),
00152     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN15NetworkTopologyE_t>.stringdata,
00153     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN15NetworkTopologyE_t>.data,
00154     qt_static_metacall,
00155     nullptr,
00156     qt_staticMetaObjectRelocatingContent<qt_meta_tag_ZN15NetworkTopologyE_t>.metaTypes,
00157     nullptr
00158 } };
00159 
00160 void NetworkTopology::qt_static_metacall(QObject *_o, QMetaObject::Call _c, int _id, void **_a)
00161 {
00162     auto *_t = static_cast<NetworkTopology *>(_o);
00163     if (_c == QMetaObject::InvokeMetaMethod) {
00164         switch (_id) {
00165         case 0: _t->deviceSelected((*reinterpret_cast< std::add_pointer_t<HostInfo>>(_a[1]))); break;
00166         default: ;
00167         }
00168     }
00169     if (_c == QMetaObject::IndexOfMethod) {
00170         if (QtMocHelpers::indexOfMethod<void (NetworkTopology::*)(const HostInfo & )>(_a,
    &NetworkTopology::deviceSelected, 0))
00171             return;
00172     }
00173 }
00174 
00175 const QMetaObject *NetworkTopology::metaObject() const
00176 {
00177     return QObject::d_ptr->metaObject ? QObject::d_ptr->dynamicMetaObject() : &staticMetaObject;
00178 }
00179 
00180 void *NetworkTopology::qt_metacast(const char *_clname)
00181 {
00182     if (!_clname) return nullptr;
00183     if (!strcmp(_clname, qt_staticMetaObjectStaticContent<qt_meta_tag_ZN15NetworkTopologyE_t>.strings))
00184         return static_cast<void*>(this);
00185     return QWidget::qt_metacast(_clname);
00186 }
00187 
00188 int NetworkTopology::qt_metacall(QMetaObject::Call _c, int _id, void **_a)
00189 {
00190     _id = QWidget::qt_metacall(_c, _id, _a);
00191     if (_id < 0)
00192         return _id;
00193     if (_c == QMetaObject::InvokeMetaMethod) {
00194         if (_id < 1)
00195             qt_static_metacall(this, _c, _id, _a);
00196         _id -= 1;
00197     }
00198     if (_c == QMetaObject::RegisterMethodArgumentMetaType) {
00199         if (_id < 1)
00200             *reinterpret_cast<QMetaType *>(_a[0]) = QMetaType();
00201         _id -= 1;
00202     }
00203     return _id;
00204 }
00205 
00206 // SIGNAL 0
00207 void NetworkTopology::deviceSelected(const HostInfo & _t1)
00208 {
00209     QMetaObject::activate<void>(this, &staticMetaObject, 0, nullptr, _t1);
00210 }
00211 QT_WARNING_POP
```

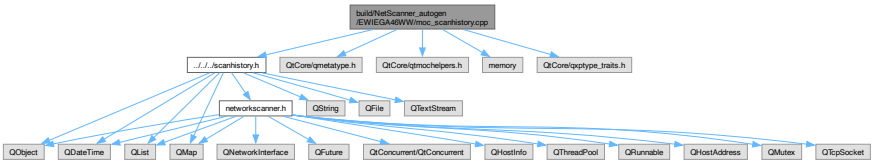
9.37 build/NetScanner_autogen/EWIEGA46WW/moc_networktopology.cpp.d 文件参考

9.38 NetScanner_autogen/JRIAJ772TK/moc_networktopology.cpp.d 文件参考

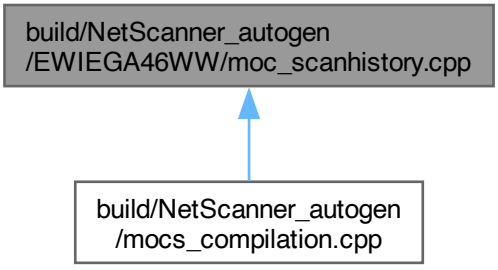
9.39 build/NetScanner_autogen/EWIEGA46WW/moc_scanhistory.cpp 文件参考

```
#include "../../../scanhistory.h"
#include <QtCore/qmetatype.h>
#include <QtCore/qtmochelpers.h>
#include <memory>
#include <QtCore/qxptype_traits.h>
```

moc_scanhistory.cpp 的引用(Include)关系图:



此图展示该文件被哪些文件直接或间接地引用了:



类

- struct QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN11ScanHistoryE_t

命名空间

- namespace QT_WARNING_DISABLE_DEPRECATED

宏定义

- #define Q_CONSTINIT

9.39.1 宏定义说明

9.39.1.1 Q_CONSTINIT

```
#define Q_CONSTINIT
```

9.40 moc_scanhistory.cpp

浏览该文件的文档.

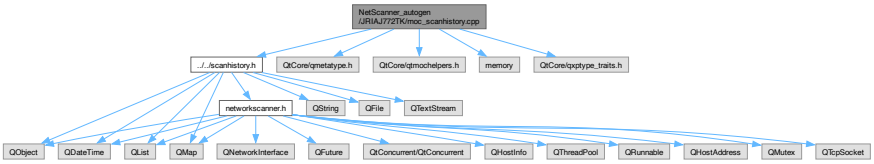
```
00001 /****************************************************************************
00002 ** Meta object code from reading C++ file 'scanhistory.h'
00003 **
00004 ** Created by: The Qt Meta Object Compiler version 69 (Qt 6.9.0)
00005 **
00006 ** WARNING! All changes made in this file will be lost!
00007 *****************************************************************************/
00008
00009 #include "../../../scanhistory.h"
00010 #include <QtCore/qmetatype.h>
00011
00012 #include <QtCore/qtmochelpers.h>
00013
00014 #include <memory>
00015
00016
00017 #include <QtCore/qxptype_traits.h>
00018 #if !defined(Q_MOC_OUTPUT_REVISION)
00019 #error "The header file 'scanhistory.h' doesn't include <QObject>."
00020 #elif Q_MOC_OUTPUT_REVISION != 69
00021 #error "This file was generated using the moc from 6.9.0. It"
00022 #error "cannot be used with the include files from this version of Qt."
00023 #error "(The moc has changed too much.)"
00024 #endif
00025
00026 #ifndef Q_CONSTINIT
00027 #define Q_CONSTINIT
00028 #endif
00029
00030 QT_WARNING_PUSH
00031 QT_WARNING_DISABLE_DEPRECATED
00032 QT_WARNING_DISABLE_GCC("-Wuseless-cast")
00033 namespace {
00034 struct qt_meta_tag_ZN11ScanHistoryE_t {};
00035 } // unnamed namespace
00036
00037 template <> constexpr inline auto
      ScanHistory::qt_create_metaobjectdata<qt_meta_tag_ZN11ScanHistoryE_t>()
00038 {
00039     namespace QMC = QtMocConstants;
00040     QtMocHelpers::StringRefStorage qt_stringData {
00041         "ScanHistory",
00042         "historyChanged",
00043         ""
00044     };
00045
00046     QtMocHelpers::UintData qt_methods {
00047         // Signal 'historyChanged'
00048         QtMocHelpers::SignalData<void()>(1, 2, QMC::AccessPublic, QMetaType::Void),
00049     };
00050     QtMocHelpers::UintData qt_properties {
00051     };
00052     QtMocHelpers::UintData qt_enums {
00053     };
00054     return QtMocHelpers::metaObjectData<ScanHistory,
      qt_meta_tag_ZN11ScanHistoryE_t>(QMC::MetaObjectFlag{}, qt_stringData,
00055             qt_methods, qt_properties, qt_enums);
00056 }
00057 Q_CONSTINIT const QMetaObject ScanHistory::staticMetaObject = { {
00058     QMetaObject::SuperData::link<QObject::staticMetaObject>(),
```

```
00059     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN11ScanHistoryE_t>.stringdata,
00060     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN11ScanHistoryE_t>.data,
00061     qt_static_metacall,
00062     nullptr,
00063     qt_staticMetaObjectRelocatingContent<qt_meta_tag_ZN11ScanHistoryE_t>.metaTypes,
00064     nullptr
00065 } };
00066
00067 void ScanHistory::qt_static_metacall(QObject *_o, QMetaObject::Call _c, int _id, void **_a)
00068 {
00069     auto *_t = static_cast<ScanHistory *>(_o);
00070     if (_c == QMetaObject::InvokeMetaMethod) {
00071         switch (_id) {
00072         case 0: _t->historyChanged(); break;
00073         default: ;
00074         }
00075     }
00076     if (_c == QMetaObject::IndexOfMethod) {
00077         if (QtMocHelpers::indexOfMethod<void (ScanHistory::*)()>(_a, &ScanHistory::historyChanged, 0))
00078             return;
00079     }
00080 }
00081
00082 const QMetaObject *ScanHistory::metaObject() const
00083 {
00084     return QObject::d_ptr->metaObject ? QObject::d_ptr->dynamicMetaObject() : &staticMetaObject;
00085 }
00086
00087 void *ScanHistory::qt_metacast(const char *_clname)
00088 {
00089     if (!_clname) return nullptr;
00090     if (!strcmp(_clname, qt_staticMetaObjectStaticContent<qt_meta_tag_ZN11ScanHistoryE_t>.strings))
00091         return static_cast<void*>(this);
00092     return QObject::qt_metacast(_clname);
00093 }
00094
00095 int ScanHistory::qt_metacall(QMetaObject::Call _c, int _id, void **_a)
00096 {
00097     _id = QObject::qt_metacall(_c, _id, _a);
00098     if (_id < 0)
00099         return _id;
00100     if (_c == QMetaObject::InvokeMetaMethod) {
00101         if (_id < 1)
00102             qt_static_metacall(this, _c, _id, _a);
00103         _id -= 1;
00104     }
00105     if (_c == QMetaObject::RegisterMethodArgumentMetaType) {
00106         if (_id < 1)
00107             *reinterpret_cast<QMetaType *>(_a[0]) = QMetaType();
00108         _id -= 1;
00109     }
00110     return _id;
00111 }
00112
00113 // SIGNAL 0
00114 void ScanHistory::historyChanged()
00115 {
00116     QMetaObject::activate(this, &staticMetaObject, 0, nullptr);
00117 }
00118 QT_WARNING_POP
```

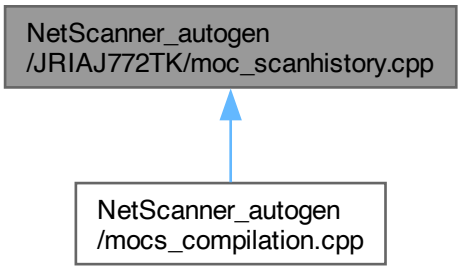
9.41 NetScanner_autogen/JRIAJ772TK/moc_scanhistory.cpp 文件参考

```
#include "../../scanhistory.h"
#include <QtCore/qmetatype.h>
#include <QtCore/qtmochelpers.h>
#include <memory>
#include <QtCore/qxptype_traits.h>
```

moc_scanhistory.cpp 的引用(Include)关系图:



此图展示该文件被哪些文件直接或间接地引用了:



类

- struct QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN11ScanHistoryE_t

命名空间

- namespace QT_WARNING_DISABLE_DEPRECATED

宏定义

- #define Q_CONSTINIT

9.41.1 宏定义说明

9.41.1.1 Q_CONSTINIT

```
#define Q_CONSTINIT
```

9.42 moc_scanhistory.cpp

浏览该文件的文档.

```
00001 /****************************************************************************
00002 ** Meta object code from reading C++ file 'scanhistory.h'
00003 **
00004 ** Created by: The Qt Meta Object Compiler version 69 (Qt 6.9.0)
00005 **
00006 ** WARNING! All changes made in this file will be lost!
00007 *****************************************************************************/
00008
00009 #include "../../scanhistory.h"
00010 #include <QtCore/qmetatype.h>
00011
00012 #include <QtCore/qtmochelpers.h>
00013
00014 #include <memory>
00015
00016
00017 #include <QtCore/qxptype_traits.h>
00018 #if !defined(Q_MOC_OUTPUT_REVISION)
00019 #error "The header file 'scanhistory.h' doesn't include <QObject>."
00020 #elif Q_MOC_OUTPUT_REVISION != 69
00021 #error "This file was generated using the moc from 6.9.0. It"
00022 #error "cannot be used with the include files from this version of Qt."
00023 #error "(The moc has changed too much.)"
00024 #endif
00025
00026 #ifndef Q_CONSTINIT
00027 #define Q_CONSTINIT
00028 #endif
00029
00030 QT_WARNING_PUSH
00031 QT_WARNING_DISABLE_DEPRECATED
00032 QT_WARNING_DISABLE_GCC("-Wuseless-cast")
00033 namespace {
00034 struct qt_meta_tag_ZN11ScanHistoryE_t {};
00035 } // unnamed namespace
00036
00037 template <> constexpr inline auto
      ScanHistory::qt_create_metaobjectdata<qt_meta_tag_ZN11ScanHistoryE_t>()
00038 {
00039     namespace QMC = QtMocConstants;
00040     QtMocHelpers::StringRefStorage qt_stringData {
00041         "ScanHistory",
00042         "historyChanged",
00043         ""
00044     };
00045
00046     QtMocHelpers::UintData qt_methods {
00047         // Signal 'historyChanged'
00048         QtMocHelpers::SignalData<void()>(1, 2, QMC::AccessPublic, QMetaType::Void),
00049     };
00050     QtMocHelpers::UintData qt_properties {
00051     };
00052     QtMocHelpers::UintData qt_enums {
00053     };
00054     return QtMocHelpers::metaObjectData<ScanHistory,
      qt_meta_tag_ZN11ScanHistoryE_t>(QMC::MetaObjectFlag{}, qt_stringData,
00055             qt_methods, qt_properties, qt_enums);
00056 }
00057 Q_CONSTINIT const QMetaObject ScanHistory::staticMetaObject = { {
00058     QMetaObject::SuperData::link<QObject::staticMetaObject>(),
00059     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN11ScanHistoryE_t>.stringdata,
00060     qt_staticMetaObjectStaticContent<qt_meta_tag_ZN11ScanHistoryE_t>.data,
00061     qt_static_metacall,
00062     nullptr,
00063     qt_staticMetaObjectRelocatingContent<qt_meta_tag_ZN11ScanHistoryE_t>.metaTypes,
00064     nullptr
00065 } };
00066
00067 void ScanHistory::qt_static_metacall(QObject *_o, QMetaObject::Call _c, int _id, void **_a)
00068 {
00069     auto *_t = static_cast<ScanHistory *>(_o);
00070     if (_c == QMetaObject::InvokeMetaMethod) {
00071         switch (_id) {
00072         case 0: _t->historyChanged(); break;
00073         default: ;
00074         }
00075     }
00076     if (_c == QMetaObject::IndexOfMethod) {
00077         if (QtMocHelpers::indexOfMethod<void (ScanHistory::*)()>(_a, &ScanHistory::historyChanged, 0))
00078             return;
00079     }
00080 }
```

```
00081
00082 const QMetaObject *ScanHistory::metaObject() const
00083 {
00084     return QObject::d_ptr->metaObject ? QObject::d_ptr->dynamicMetaObject() : &staticMetaObject;
00085 }
00086
00087 void *ScanHistory::qt_metacast(const char *_clname)
00088 {
00089     if (!_clname) return nullptr;
00090     if (!strcmp(_clname, qt_staticMetaObjectStaticContent<qt_meta_tag_ZN11ScanHistoryE_t>.strings))
00091         return static_cast<void*>(this);
00092     return QObject::qt_metacast(_clname);
00093 }
00094
00095 int ScanHistory::qt_metacall(QMetaObject::Call _c, int _id, void **_a)
00096 {
00097     _id = QObject::qt_metacall(_c, _id, _a);
00098     if (_id < 0)
00099         return _id;
00100     if (_c == QMetaObject::InvokeMetaMethod) {
00101         if (_id < 1)
00102             qt_static_metacall(this, _c, _id, _a);
00103         _id -= 1;
00104     }
00105     if (_c == QMetaObject::RegisterMethodArgumentMetaType) {
00106         if (_id < 1)
00107             *reinterpret_cast<QMetaType *>(_a[0]) = QMetaType();
00108         _id -= 1;
00109     }
00110     return _id;
00111 }
00112
00113 // SIGNAL 0
00114 void ScanHistory::historyChanged()
00115 {
00116     QMetaObject::activate(this, &staticMetaObject, 0, nullptr);
00117 }
00118 QT_WARNING_POP
```

9.43 build/NetScanner_autogen/EWIEGA46WW/moc_scanhistory.cpp.d 文件参考

9.44 NetScanner_autogen/JRIAJ772TK/moc_scanhistory.cpp.d 文件参考

9.45 build/NetScanner_autogen/moc_predefs.h 文件参考

宏定义

- #define QT_CHARTS_LIB 1
- #define QT_CHARTS_USE_NAMESPACE 1
- #define QT_CONCURRENT_LIB 1
- #define QT_CORE_LIB 1
- #define QT_GUI_LIB 1
- #define QT_NETWORK_LIB 1
- #define QT_NO_DEBUG 1
- #define QT_OPENGLWIDGETS_LIB 1
- #define QT_OPENGL_LIB 1
- #define QT_WIDGETS_LIB 1
- #define SIZEOF_DPTR (sizeof(void*))
- #define TARGET_IPHONE_SIMULATOR 0
- #define TARGET_OS_ARROW 1
- #define TARGET_OS_BRIDGE 0
- #define TARGET_OS_DRIVERKIT 0
- #define TARGET_OS_EMBEDDED 0

- #define TARGET_OS_IOS 0
- #define TARGET_OS_IOSMAC 0
- #define TARGET_OS_IPHONE 0
- #define TARGET_OS_LINUX 0
- #define TARGET_OS_MAC 1
- #define TARGET_OS_MACCATALYST 0
- #define TARGET_OS_NANO 0
- #define TARGET_OS_OSX 1
- #define TARGET_OS_SIMULATOR 0
- #define TARGET_OS_TV 0
- #define TARGET_OS_UIKITFORMAC 0
- #define TARGET_OS_UNIX 0
- #define TARGET_OS_VISION 0
- #define TARGET_OS_WATCH 0
- #define TARGET_OS_WIN32 0
- #define TARGET_OS_WINDOWS 0
- #define TARGET_OS_XR 0
- #define _LP64 1
- #define __AARCH64EL__ 1
- #define __AARCH64_CMODEL_SMALL__ 1
- #define __AARCH64_SIMD__ 1
- #define __APPLE_CC__ 6000
- #define __APPLE__ 1
- #define __ARM64_ARCH_8__ 1
- #define __ARM_64BIT_STATE 1
- #define __ARM_ACLE 200
- #define __ARM_ALIGN_MAX_STACK_PWR 4
- #define __ARM_ARCH 8
- #define __ARM_ARCH_8_3__ 1
- #define __ARM_ARCH_8_4__ 1
- #define __ARM_ARCH_8_5__ 1
- #define __ARM_ARCH_ISA_A64 1
- #define __ARM_ARCH_PROFILE 'A'
- #define __ARM_FEATURE_AES 1
- #define __ARM_FEATURE_ATOMICS 1
- #define __ARM_FEATURE_BTI 1
- #define __ARM_FEATURE_CLZ 1
- #define __ARM_FEATURE_COMPLEX 1
- #define __ARM_FEATURE_CRC32 1
- #define __ARM_FEATURE_CRYPTO 1
- #define __ARM_FEATURE_DIRECTED_ROUNDING 1
- #define __ARM_FEATURE_DIV 1
- #define __ARM_FEATURE_DOTPROD 1
- #define __ARM_FEATURE_FMA 1
- #define __ARM_FEATURE_FP16_FML 1
- #define __ARM_FEATURE_FP16_SCALAR_ARITHMETIC 1
- #define __ARM_FEATURE_FP16_VECTOR_ARITHMETIC 1
- #define __ARM_FEATURE_FRINT 1
- #define __ARM_FEATURE_IDIV 1
- #define __ARM_FEATURE_JCVT 1
- #define __ARM_FEATURE_LDREX 0xF
- #define __ARM_FEATURE_NUMERIC_MAXMIN 1
- #define __ARM_FEATURE_PAUTH 1
- #define __ARM_FEATURE_QRDMX 1
- #define __ARM_FEATURE_RCPC 1

- #define __ARM_FEATURE_SHA2 1
- #define __ARM_FEATURE_SHA3 1
- #define __ARM_FEATURE_SHA512 1
- #define __ARM_FEATURE_UNALIGNED 1
- #define __ARM_FP 0xE
- #define __ARM_FP16_ARGS 1
- #define __ARM_FP16_FORMAT_IEEE 1
- #define __ARM_NEON 1
- #define __ARM_NEON_FP 0xE
- #define __ARM_NEON__ 1
- #define __ARM_PCS_AAPCS64 1
- #define __ARM_SIZEOF_MINIMAL_ENUM 4
- #define __ARM_SIZEOF_WCHAR_T 4
- #define __ARM_STATE_ZA 1
- #define __ARM_STATE_ZT0 1
- #define __ATOMIC_ACQUIRE 2
- #define __ATOMIC_ACQ_REL 4
- #define __ATOMIC_CONSUME 1
- #define __ATOMIC_RELAXED 0
- #define __ATOMIC_RELEASE 3
- #define __ATOMIC_SEQ_CST 5
- #define __BIGGEST_ALIGNMENT__ 8
- #define __BITINT_MAXWIDTH__ 128
- #define __BLOCKS__ 1
- #define __BOOL_WIDTH__ 8
- #define __BYTE_ORDER__ __ORDER_LITTLE_ENDIAN__
- #define __CHAR16_TYPE__ unsigned short
- #define __CHAR32_TYPE__ unsigned int
- #define __CHAR_BIT__ 8
- #define __CLANG_ATOMIC_BOOL_LOCK_FREE 2
- #define __CLANG_ATOMIC_CHAR16_T_LOCK_FREE 2
- #define __CLANG_ATOMIC_CHAR32_T_LOCK_FREE 2
- #define __CLANG_ATOMIC_CHAR_LOCK_FREE 2
- #define __CLANG_ATOMIC_INT_LOCK_FREE 2
- #define __CLANG_ATOMIC_LLONG_LOCK_FREE 2
- #define __CLANG_ATOMIC_LONG_LOCK_FREE 2
- #define __CLANG_ATOMIC_POINTER_LOCK_FREE 2
- #define __CLANG_ATOMIC_SHORT_LOCK_FREE 2
- #define __CLANG_ATOMIC_WCHAR_T_LOCK_FREE 2
- #define __CONSTANT_CFSTRINGS__ 1
- #define __DBL_DECIMAL_DIG__ 17
- #define __DBL_DENORM_MIN__ 4.9406564584124654e-324
- #define __DBL_DIG__ 15
- #define __DBL_EPSILON__ 2.2204460492503131e-16
- #define __DBL_HAS_DENORM__ 1
- #define __DBL_HAS_INFINITY__ 1
- #define __DBL_HAS_QUIET_NAN__ 1
- #define __DBL_MANT_DIG__ 53
- #define __DBL_MAX_10_EXP__ 308
- #define __DBL_MAX_EXP__ 1024
- #define __DBL_MAX__ 1.7976931348623157e+308
- #define __DBL_MIN_10_EXP__ (-307)
- #define __DBL_MIN_EXP__ (-1021)
- #define __DBL_MIN__ 2.2250738585072014e-308
- #define __DBL_NORM_MAX__ 1.7976931348623157e+308

- #define __DECIMAL_DIG__ __LDBL_DECIMAL_DIG__
- #define __DEPRECATED 1
- #define __DYNAMIC__ 1
- #define __ENVIRONMENT_MAC_OS_X_VERSION_MIN_REQUIRED__ 150000
- #define __ENVIRONMENT_OS_VERSION_MIN_REQUIRED__ 150000
- #define __EXCEPTIONS 1
- #define __FINITE_MATH_ONLY__ 0
- #define __FLT16_DECIMAL_DIG__ 5
- #define __FLT16_DENORM_MIN__ 5.9604644775390625e-8F16
- #define __FLT16_DIG__ 3
- #define __FLT16_EPSILON__ 9.765625e-4F16
- #define __FLT16_HAS_DENORM__ 1
- #define __FLT16_HAS_INFINITY__ 1
- #define __FLT16_HAS_QUIET_NAN__ 1
- #define __FLT16_MANT_DIG__ 11
- #define __FLT16_MAX_10_EXP__ 4
- #define __FLT16_MAX_EXP__ 16
- #define __FLT16_MAX__ 6.5504e+4F16
- #define __FLT16_MIN_10_EXP__ (-4)
- #define __FLT16_MIN_EXP__ (-13)
- #define __FLT16_MIN__ 6.103515625e-5F16
- #define __FLT16_NORM_MAX__ 6.5504e+4F16
- #define __FLT_DECIMAL_DIG__ 9
- #define __FLT_DENORM_MIN__ 1.40129846e-45F
- #define __FLT_DIG__ 6
- #define __FLT_EPSILON__ 1.19209290e-7F
- #define __FLT_HAS_DENORM__ 1
- #define __FLT_HAS_INFINITY__ 1
- #define __FLT_HAS_QUIET_NAN__ 1
- #define __FLT_MANT_DIG__ 24
- #define __FLT_MAX_10_EXP__ 38
- #define __FLT_MAX_EXP__ 128
- #define __FLT_MAX__ 3.40282347e+38F
- #define __FLT_MIN_10_EXP__ (-37)
- #define __FLT_MIN_EXP__ (-125)
- #define __FLT_MIN__ 1.17549435e-38F
- #define __FLT_NORM_MAX__ 3.40282347e+38F
- #define __FLT_RADIX__ 2
- #define __FPCLASS_NEGINF 0x0004
- #define __FPCLASS_NEGNORMAL 0x0008
- #define __FPCLASS_NEGSUBNORMAL 0x0010
- #define __FPCLASS_NEGZERO 0x0020
- #define __FPCLASS_POSINF 0x0200
- #define __FPCLASS_POSNORMAL 0x0100
- #define __FPCLASS_POSSUBNORMAL 0x0080
- #define __FPCLASS_POSZERO 0x0040
- #define __FPCLASS_QNAN 0x0002
- #define __FPCLASS_SNAN 0x0001
- #define __FP_FAST_FMA 1
- #define __FP_FAST_FMAF 1
- #define __GCC_ASM_FLAG_OUTPUTS__ 1
- #define __GCC_ATOMIC_BOOL_LOCK_FREE 2
- #define __GCC_ATOMIC_CHAR16_T_LOCK_FREE 2
- #define __GCC_ATOMIC_CHAR32_T_LOCK_FREE 2
- #define __GCC_ATOMIC_CHAR_LOCK_FREE 2

- #define __GCC_ATOMIC_INT_LOCK_FREE 2
- #define __GCC_ATOMIC_LLONG_LOCK_FREE 2
- #define __GCC_ATOMIC_LONG_LOCK_FREE 2
- #define __GCC_ATOMIC_POINTER_LOCK_FREE 2
- #define __GCC_ATOMIC_SHORT_LOCK_FREE 2
- #define __GCC_ATOMIC_TEST_AND_SET_TRUEVAL 1
- #define __GCC_ATOMIC_WCHAR_T_LOCK_FREE 2
- #define __GCC_CONSTRUCTIVE_SIZE 64
- #define __GCC_DESTRUCTIVE_SIZE 64
- #define __GCC_HAVE_DWARF2_CFI_ASM 1
- #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_1 1
- #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_16 1
- #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_2 1
- #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_4 1
- #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_8 1
- #define __GLIBCXX_BITSIZE_INT_N_0 128
- #define __GLIBCXX_TYPE_INT_N_0 __int128
- #define __GNUC_GNU_INLINE__ 1
- #define __GNUC_MINOR__ 2
- #define __GNUC_PATCHLEVEL__ 1
- #define __GNUC__ 4
- #define __GNUG__ 4
- #define __GXX_ABI_VERSION 1002
- #define __GXX_EXPERIMENTAL_CXX0X__ 1
- #define __GXX_RTTI 1
- #define __GXX_WEAK__ 1
- #define __HAVE_FUNCTION_MULTI_VERSIONING 1
- #define __INT16_C_SUFFIX__
- #define __INT16_FMTd__ "hd"
- #define __INT16_FMTi__ "hi"
- #define __INT16_MAX__ 32767
- #define __INT16_TYPE__ short
- #define __INT32_C_SUFFIX__
- #define __INT32_FMTd__ "d"
- #define __INT32_FMTi__ "i"
- #define __INT32_MAX__ 2147483647
- #define __INT32_TYPE__ int
- #define __INT64_C_SUFFIX__ LL
- #define __INT64_FMTd__ "lld"
- #define __INT64_FMTi__ "lli"
- #define __INT64_MAX__ 9223372036854775807LL
- #define __INT64_TYPE__ long long int
- #define __INT8_C_SUFFIX__
- #define __INT8_FMTd__ "hhd"
- #define __INT8_FMTi__ "hhi"
- #define __INT8_MAX__ 127
- #define __INT8_TYPE__ signed char
- #define __INTMAX_C_SUFFIX__ L
- #define __INTMAX_FMTd__ "ld"
- #define __INTMAX_FMTi__ "li"
- #define __INTMAX_MAX__ 9223372036854775807L
- #define __INTMAX_TYPE__ long int
- #define __INTMAX_WIDTH__ 64
- #define __INTPTR_FMTd__ "ld"
- #define __INTPTR_FMTi__ "li"

- #define __INTPTR_MAX__ 9223372036854775807L
- #define __INTPTR_TYPE__ long int
- #define __INTPTR_WIDTH__ 64
- #define __INT_FAST16_FMTd__ "hd"
- #define __INT_FAST16_FMTi__ "hi"
- #define __INT_FAST16_MAX__ 32767
- #define __INT_FAST16_TYPE__ short
- #define __INT_FAST16_WIDTH__ 16
- #define __INT_FAST32_FMTd__ "d"
- #define __INT_FAST32_FMTi__ "i"
- #define __INT_FAST32_MAX__ 2147483647
- #define __INT_FAST32_TYPE__ int
- #define __INT_FAST32_WIDTH__ 32
- #define __INT_FAST64_FMTd__ "lld"
- #define __INT_FAST64_FMTi__ "lli"
- #define __INT_FAST64_MAX__ 9223372036854775807LL
- #define __INT_FAST64_TYPE__ long long int
- #define __INT_FAST64_WIDTH__ 64
- #define __INT_FAST8_FMTd__ "hhd"
- #define __INT_FAST8_FMTi__ "hhi"
- #define __INT_FAST8_MAX__ 127
- #define __INT_FAST8_TYPE__ signed char
- #define __INT_FAST8_WIDTH__ 8
- #define __INT_LEAST16_FMTd__ "hd"
- #define __INT_LEAST16_FMTi__ "hi"
- #define __INT_LEAST16_MAX__ 32767
- #define __INT_LEAST16_TYPE__ short
- #define __INT_LEAST16_WIDTH__ 16
- #define __INT_LEAST32_FMTd__ "d"
- #define __INT_LEAST32_FMTi__ "i"
- #define __INT_LEAST32_MAX__ 2147483647
- #define __INT_LEAST32_TYPE__ int
- #define __INT_LEAST32_WIDTH__ 32
- #define __INT_LEAST64_FMTd__ "lld"
- #define __INT_LEAST64_FMTi__ "lli"
- #define __INT_LEAST64_MAX__ 9223372036854775807LL
- #define __INT_LEAST64_TYPE__ long long int
- #define __INT_LEAST64_WIDTH__ 64
- #define __INT_LEAST8_FMTd__ "hhd"
- #define __INT_LEAST8_FMTi__ "hhi"
- #define __INT_LEAST8_MAX__ 127
- #define __INT_LEAST8_TYPE__ signed char
- #define __INT_LEAST8_WIDTH__ 8
- #define __INT_MAX__ 2147483647
- #define __INT_WIDTH__ 32
- #define __LDBL_DECIMAL_DIG__ 17
- #define __LDBL_DENORM_MIN__ 4.9406564584124654e-324L
- #define __LDBL_DIG__ 15
- #define __LDBL_EPSILON__ 2.2204460492503131e-16L
- #define __LDBL_HAS_DENORM__ 1
- #define __LDBL_HAS_INFINITY__ 1
- #define __LDBL_HAS_QUIET_NAN__ 1
- #define __LDBL_MANT_DIG__ 53
- #define __LDBL_MAX_10_EXP__ 308
- #define __LDBL_MAX_EXP__ 1024

- #define __LDBL_MAX__ 1.7976931348623157e+308L
- #define __LDBL_MIN_10_EXP__ (-307)
- #define __LDBL_MIN_EXP__ (-1021)
- #define __LDBL_MIN__ 2.2250738585072014e-308L
- #define __LDBL_NORM_MAX__ 1.7976931348623157e+308L
- #define __LITTLE_ENDIAN__ 1
- #define __LLONG_WIDTH__ 64
- #define __LONG_LONG_MAX__ 9223372036854775807LL
- #define __LONG_MAX__ 9223372036854775807L
- #define __LONG_WIDTH__ 64
- #define __LP64__ 1
- #define __MACH__ 1
- #define __MEMORY_SCOPE_DEVICE 1
- #define __MEMORY_SCOPE_SINGLE 4
- #define __MEMORY_SCOPE_SYSTEM 0
- #define __MEMORY_SCOPE_WRKGRP 2
- #define __MEMORY_SCOPE_WVFRNT 3
- #define __NO_INLINE__ 1
- #define __NO_MATH_ERRNO__ 1
- #define __OBJC_BOOL_IS_BOOL 1
- #define __OPENCL_MEMORY_SCOPE_ALL_SVM_DEVICES 3
- #define __OPENCL_MEMORY_SCOPE_DEVICE 2
- #define __OPENCL_MEMORY_SCOPE_SUB_GROUP 4
- #define __OPENCL_MEMORY_SCOPE_WORK_GROUP 1
- #define __OPENCL_MEMORY_SCOPE_WORK_ITEM 0
- #define __ORDER_BIG_ENDIAN__ 4321
- #define __ORDER_LITTLE_ENDIAN__ 1234
- #define __ORDER_PDP_ENDIAN__ 3412
- #define __PIC__ 2
- #define __POINTER_WIDTH__ 64
- #define __PRAGMA_REDEFINE_EXTNAME 1
- #define __PTRDIFF_FMTd__ "ld"
- #define __PTRDIFF_FMTi__ "li"
- #define __PTRDIFF_MAX__ 9223372036854775807L
- #define __PTRDIFF_TYPE__ long int
- #define __PTRDIFF_WIDTH__ 64
- #define __REGISTER_PREFIX__
- #define __SCHAR_MAX__ 127
- #define __SHRT_MAX__ 32767
- #define __SHRT_WIDTH__ 16
- #define __SIG_ATOMIC_MAX__ 2147483647
- #define __SIG_ATOMIC_WIDTH__ 32
- #define __SIZEOF_DOUBLE__ 8
- #define __SIZEOF_FLOAT__ 4
- #define __SIZEOF_INT128__ 16
- #define __SIZEOF_INT__ 4
- #define __SIZEOF_LONG_DOUBLE__ 8
- #define __SIZEOF_LONG_LONG__ 8
- #define __SIZEOF_LONG__ 8
- #define __SIZEOF_POINTER__ 8
- #define __SIZEOF_PTRDIFF_T__ 8
- #define __SIZEOF_SHORT__ 2
- #define __SIZEOF_SIZE_T__ 8
- #define __SIZEOF_WCHAR_T__ 4
- #define __SIZEOF_WINT_T__ 4

- #define __SIZE_FMTX__ "lX"
- #define __SIZE_FMTo__ "lo"
- #define __SIZE_FMTu__ "lu"
- #define __SIZE_FMTx__ "lx"
- #define __SIZE_MAX__ 18446744073709551615UL
- #define __SIZE_TYPE__ long unsigned int
- #define __SIZE_WIDTH__ 64
- #define __SSP__ 1
- #define __STDCPP_DEFAULT_NEW_ALIGNMENT__ 16UL
- #define __STDCPP_THREADS__ 1
- #define __STDC_EMBED_EMPTY__ 2
- #define __STDC_EMBED_FOUND__ 1
- #define __STDC_EMBED_NOT_FOUND__ 0
- #define __STDC_HOSTED__ 1
- #define __STDC_NO_THREADS__ 1
- #define __STDC_UTF_16__ 1
- #define __STDC_UTF_32__ 1
- #define __STDC__ 1
- #define __UINT16_C_SUFFIX__
- #define __UINT16_FMTX__ "hX"
- #define __UINT16_FMTo__ "ho"
- #define __UINT16_FMTu__ "hu"
- #define __UINT16_FMTx__ "hx"
- #define __UINT16_MAX__ 65535
- #define __UINT16_TYPE__ unsigned short
- #define __UINT32_C_SUFFIX__ U
- #define __UINT32_FMTX__ "X"
- #define __UINT32_FMTo__ "o"
- #define __UINT32_FMTu__ "u"
- #define __UINT32_FMTx__ "x"
- #define __UINT32_MAX__ 4294967295U
- #define __UINT32_TYPE__ unsigned int
- #define __UINT64_C_SUFFIX__ ULL
- #define __UINT64_FMTX__ "llX"
- #define __UINT64_FMTo__ "llo"
- #define __UINT64_FMTu__ "llu"
- #define __UINT64_FMTx__ "llx"
- #define __UINT64_MAX__ 18446744073709551615ULL
- #define __UINT64_TYPE__ long long unsigned int
- #define __UINT8_C_SUFFIX__
- #define __UINT8_FMTX__ "hhX"
- #define __UINT8_FMTo__ "hho"
- #define __UINT8_FMTu__ "hhu"
- #define __UINT8_FMTx__ "hhx"
- #define __UINT8_MAX__ 255
- #define __UINT8_TYPE__ unsigned char
- #define __UINTMAX_C_SUFFIX__ UL
- #define __UINTMAX_FMTX__ "lX"
- #define __UINTMAX_FMTo__ "lo"
- #define __UINTMAX_FMTu__ "lu"
- #define __UINTMAX_FMTx__ "lx"
- #define __UINTMAX_MAX__ 18446744073709551615UL
- #define __UINTMAX_TYPE__ long unsigned int
- #define __UINTMAX_WIDTH__ 64
- #define __UINTPTR_FMTX__ "lX"

- #define __UINTPTR_FMTo__ "lo"
- #define __UINTPTR_FMTu__ "lu"
- #define __UINTPTR_FMTx__ "lx"
- #define __UINTPTR_MAX__ 18446744073709551615UL
- #define __UINTPTR_TYPE__ long unsigned int
- #define __UINTPTR_WIDTH__ 64
- #define __UINT_FAST16_FMTX__ "hX"
- #define __UINT_FAST16_FMTo__ "ho"
- #define __UINT_FAST16_FMTu__ "hu"
- #define __UINT_FAST16_FMTx__ "hx"
- #define __UINT_FAST16_MAX__ 65535
- #define __UINT_FAST16_TYPE__ unsigned short
- #define __UINT_FAST32_FMTX__ "X"
- #define __UINT_FAST32_FMTo__ "o"
- #define __UINT_FAST32_FMTu__ "u"
- #define __UINT_FAST32_FMTx__ "x"
- #define __UINT_FAST32_MAX__ 4294967295U
- #define __UINT_FAST32_TYPE__ unsigned int
- #define __UINT_FAST64_FMTX__ "llX"
- #define __UINT_FAST64_FMTo__ "llo"
- #define __UINT_FAST64_FMTu__ "llu"
- #define __UINT_FAST64_FMTx__ "llx"
- #define __UINT_FAST64_MAX__ 18446744073709551615ULL
- #define __UINT_FAST64_TYPE__ long long unsigned int
- #define __UINT_FAST8_FMTX__ "hhX"
- #define __UINT_FAST8_FMTo__ "hho"
- #define __UINT_FAST8_FMTu__ "hhu"
- #define __UINT_FAST8_FMTx__ "hhx"
- #define __UINT_FAST8_MAX__ 255
- #define __UINT_FAST8_TYPE__ unsigned char
- #define __UINT_LEAST16_FMTX__ "hX"
- #define __UINT_LEAST16_FMTo__ "ho"
- #define __UINT_LEAST16_FMTu__ "hu"
- #define __UINT_LEAST16_FMTx__ "hx"
- #define __UINT_LEAST16_MAX__ 65535
- #define __UINT_LEAST16_TYPE__ unsigned short
- #define __UINT_LEAST32_FMTX__ "X"
- #define __UINT_LEAST32_FMTo__ "o"
- #define __UINT_LEAST32_FMTu__ "u"
- #define __UINT_LEAST32_FMTx__ "x"
- #define __UINT_LEAST32_MAX__ 4294967295U
- #define __UINT_LEAST32_TYPE__ unsigned int
- #define __UINT_LEAST64_FMTX__ "llX"
- #define __UINT_LEAST64_FMTo__ "llo"
- #define __UINT_LEAST64_FMTu__ "llu"
- #define __UINT_LEAST64_FMTx__ "llx"
- #define __UINT_LEAST64_MAX__ 18446744073709551615ULL
- #define __UINT_LEAST64_TYPE__ long long unsigned int
- #define __UINT_LEAST8_FMTX__ "hhX"
- #define __UINT_LEAST8_FMTo__ "hho"
- #define __UINT_LEAST8_FMTu__ "hhu"
- #define __UINT_LEAST8_FMTx__ "hhx"
- #define __UINT_LEAST8_MAX__ 255
- #define __UINT_LEAST8_TYPE__ unsigned char
- #define __USER_LABEL_PREFIX__ _

- #define __VERSION__ "Apple LLVM 17.0.0 (clang-1700.0.13.3)"
- #define __WCHAR_MAX__ 2147483647
- #define __WCHAR_TYPE__ int
- #define __WCHAR_WIDTH__ 32
- #define __WINT_MAX__ 2147483647
- #define __WINT_TYPE__ int
- #define __WINT_WIDTH__ 32
- #define __aarch64__ 1
- #define __apple_build_version__ 17000013
- #define __arm64 1
- #define __arm64__ 1
- #define __block __attribute__((__blocks__(byref)))
- #define __clang__ 1
- #define __clang_literal_encoding__ "UTF-8"
- #define __clang_major__ 17
- #define __clang_minor__ 0
- #define __clang_patchlevel__ 0
- #define __clang_version__ "17.0.0 (clang-1700.0.13.3)"
- #define __clang_wide_literal_encoding__ "UTF-32"
- #define __cplusplus 201703L
- #define __cpp_aggregate_bases 201603L
- #define __cpp_aggregate_nsdmi 201304L
- #define __cpp_alias_templates 200704L
- #define __cpp_aligned_new 201606L
- #define __cpp_attributes 200809L
- #define __cpp_binary_literals 201304L
- #define __cpp_capture_star_this 201603L
- #define __cpp_constexpr 201603L
- #define __cpp_constexpr_in_decltype 201711L
- #define __cpp_decltype 200707L
- #define __cpp_decltype_auto 201304L
- #define __cpp_deduction_guides 201703L
- #define __cpp_delegating_constructors 200604L
- #define __cpp_deleted_function 202403L
- #define __cpp_digit_separators 201309L
- #define __cpp_enumerator_attributes 201411L
- #define __cpp_exceptions 199711L
- #define __cpp_fold_expressions 201603L
- #define __cpp_generic_lambdas 201304L
- #define __cpp_guaranteed_copy_elision 201606L
- #define __cpp_hex_float 201603L
- #define __cpp_if_constexpr 201606L
- #define __cpp_impl_destroying_delete 201806L
- #define __cpp_inheriting_constructors 201511L
- #define __cpp_init_captures 201304L
- #define __cpp_initializer_lists 200806L
- #define __cpp_inline_variables 201606L
- #define __cpp_lambdas 200907L
- #define __cpp_named_character_escapes 202207L
- #define __cpp_namespace_attributes 201411L
- #define __cpp_nested_namespace_definitions 201411L
- #define __cpp_noexcept_function_type 201510L
- #define __cpp_nontype_template_args 201411L
- #define __cpp_nontype_template_parameter_auto 201606L
- #define __cpp_nsdmi 200809L

- #define __cpp_pack_indexing 202311L
- #define __cpp_placeholder_variables 202306L
- #define __cpp_range_based_for 201603L
- #define __cpp_raw_strings 200710L
- #define __cpp_ref_qualifiers 200710L
- #define __cpp_return_type_deduction 201304L
- #define __cpp_rtti 199711L
- #define __cpp_rvalue_references 200610L
- #define __cpp_static_assert 201411L
- #define __cpp_static_call_operator 202207L
- #define __cpp_structured_bindings 202403L
- #define __cpp_template_auto 201606L
- #define __cpp_template_template_args 201611L
- #define __cpp_threadsafe_static_init 200806L
- #define __cpp_unicode_characters 200704L
- #define __cpp_unicode_literals 200710L
- #define __cpp_user_defined_literals 200809L
- #define __cpp_variable_templates 201304L
- #define __cpp_variadic_templates 200704L
- #define __cpp_variadic_using 201611L
- #define __llvm__ 1
- #define __nonnull _Nonnull
- #define __null_unspecified _Null_unspecified
- #define __nullable _Nullable
- #define __pic__ 2
- #define __private_extern__ extern
- #define __strong
- #define __unsafe_unretained
- #define __weak __attribute__((objc_gc(weak)))

9.45.1 宏定义说明

9.45.1.1 __aarch64__

```
#define __aarch64__ 1
```

9.45.1.2 __AARCH64_CMODEL_SMALL__

```
#define __AARCH64_CMODEL_SMALL__ 1
```

9.45.1.3 __AARCH64_SIMD__

```
#define __AARCH64_SIMD__ 1
```

9.45.1.4 __AARCH64EL__

```
#define __AARCH64EL__ 1
```

9.45.1.5 __APPLE__

```
#define __APPLE__ 1
```

9.45.1.6 __apple_build_version__

```
#define __apple_build_version__ 17000013
```

9.45.1.7 __APPLE_CC__

```
#define __APPLE_CC__ 6000
```

9.45.1.8 __arm64

```
#define __arm64 1
```

9.45.1.9 __arm64__

```
#define __arm64__ 1
```

9.45.1.10 __ARM64_ARCH_8__

```
#define __ARM64_ARCH_8__ 1
```

9.45.1.11 __ARM_64BIT_STATE

```
#define __ARM_64BIT_STATE 1
```

9.45.1.12 __ARM_ACLE

```
#define __ARM_ACLE 200
```

9.45.1.13 __ARM_ALIGN_MAX_STACK_PWR

```
#define __ARM_ALIGN_MAX_STACK_PWR 4
```

9.45.1.14 __ARM_ARCH

```
#define __ARM_ARCH 8
```

9.45.1.15 __ARM_ARCH_8_3__

```
#define __ARM_ARCH_8_3__ 1
```

9.45.1.16 __ARM_ARCH_8_4__

```
#define __ARM_ARCH_8_4__ 1
```

9.45.1.17 __ARM_ARCH_8_5__

```
#define __ARM_ARCH_8_5__ 1
```

9.45.1.18 __ARM_ARCH_ISA_A64

```
#define __ARM_ARCH_ISA_A64 1
```

9.45.1.19 __ARM_ARCH_PROFILE

```
#define __ARM_ARCH_PROFILE 'A'
```

9.45.1.20 __ARM_FEATURE_AES

```
#define __ARM_FEATURE_AES 1
```

9.45.1.21 __ARM_FEATURE_ATOMICS

```
#define __ARM_FEATURE_ATOMICS 1
```

9.45.1.22 __ARM_FEATURE_BTI

```
#define __ARM_FEATURE_BTI 1
```

9.45.1.23 __ARM_FEATURE_CLZ

```
#define __ARM_FEATURE_CLZ 1
```

9.45.1.24 __ARM_FEATURE_COMPLEX

```
#define __ARM_FEATURE_COMPLEX 1
```

9.45.1.25 __ARM_FEATURE_CRC32

```
#define __ARM_FEATURE_CRC32 1
```

9.45.1.26 __ARM_FEATURE_CRYPTO

```
#define __ARM_FEATURE_CRYPTO 1
```

9.45.1.27 __ARM_FEATURE_DIRECTED_ROUNDING

```
#define __ARM_FEATURE_DIRECTED_ROUNDING 1
```

9.45.1.28 __ARM_FEATURE_DIV

```
#define __ARM_FEATURE_DIV 1
```

9.45.1.29 __ARM_FEATURE_DOTPROD

```
#define __ARM_FEATURE_DOTPROD 1
```

9.45.1.30 __ARM_FEATURE_FMA

```
#define __ARM_FEATURE_FMA 1
```

9.45.1.31 __ARM_FEATURE_FP16_FML

```
#define __ARM_FEATURE_FP16_FML 1
```

9.45.1.32 __ARM_FEATURE_FP16_SCALAR_ARITHMETIC

```
#define __ARM_FEATURE_FP16_SCALAR_ARITHMETIC 1
```

9.45.1.33 __ARM_FEATURE_FP16_VECTOR_ARITHMETIC

```
#define __ARM_FEATURE_FP16_VECTOR_ARITHMETIC 1
```

9.45.1.34 __ARM_FEATURE_FRINT

```
#define __ARM_FEATURE_FRINT 1
```

9.45.1.35 __ARM_FEATURE_IDIV

```
#define __ARM_FEATURE_IDIV 1
```

9.45.1.36 __ARM_FEATURE_JCVT

```
#define __ARM_FEATURE_JCVT 1
```

9.45.1.37 __ARM_FEATURE_LDREX

```
#define __ARM_FEATURE_LDREX 0xF
```

9.45.1.38 __ARM_FEATURE_NUMERIC_MAXMIN

```
#define __ARM_FEATURE_NUMERIC_MAXMIN 1
```

9.45.1.39 __ARM_FEATURE_PAUTH

```
#define __ARM_FEATURE_PAUTH 1
```

9.45.1.40 __ARM_FEATURE_QRDMX

```
#define __ARM_FEATURE_QRDMX 1
```

9.45.1.41 __ARM_FEATURE_RCPC

```
#define __ARM_FEATURE_RCPC 1
```

9.45.1.42 __ARM_FEATURE_SHA2

```
#define __ARM_FEATURE_SHA2 1
```

9.45.1.43 __ARM_FEATURE_SHA3

```
#define __ARM_FEATURE_SHA3 1
```

9.45.1.44 __ARM_FEATURE_SHA512

```
#define __ARM_FEATURE_SHA512 1
```

9.45.1.45 __ARM_FEATURE_UNALIGNED

```
#define __ARM_FEATURE_UNALIGNED 1
```

9.45.1.46 __ARM_FP

```
#define __ARM_FP 0xE
```

9.45.1.47 __ARM_FP16_ARGS

```
#define __ARM_FP16_ARGS 1
```

9.45.1.48 __ARM_FP16_FORMAT_IEEE

```
#define __ARM_FP16_FORMAT_IEEE 1
```

9.45.1.49 __ARM_NEON

```
#define __ARM_NEON 1
```

9.45.1.50 __ARM_NEON__

```
#define __ARM_NEON__ 1
```

9.45.1.51 __ARM_NEON_FP

```
#define __ARM_NEON_FP 0xE
```

9.45.1.52 __ARM_PCS_AAPCS64

```
#define __ARM_PCS_AAPCS64 1
```

9.45.1.53 __ARM_SIZEOF_MINIMAL_ENUM

```
#define __ARM_SIZEOF_MINIMAL_ENUM 4
```

9.45.1.54 __ARM_SIZEOF_WCHAR_T

```
#define __ARM_SIZEOF_WCHAR_T 4
```

9.45.1.55 __ARM_STATE_ZA

```
#define __ARM_STATE_ZA 1
```

9.45.1.56 __ARM_STATE_ZT0

```
#define __ARM_STATE_ZT0 1
```

9.45.1.57 __ATOMIC_ACQ_REL

```
#define __ATOMIC_ACQ_REL 4
```

9.45.1.58 __ATOMIC_ACQUIRE

```
#define __ATOMIC_ACQUIRE 2
```

9.45.1.59 __ATOMIC_CONSUME

```
#define __ATOMIC_CONSUME 1
```

9.45.1.60 __ATOMIC_RELAXED

```
#define __ATOMIC_RELAXED 0
```

9.45.1.61 __ATOMIC_RELEASE

```
#define __ATOMIC_RELEASE 3
```

9.45.1.62 __ATOMIC_SEQ_CST

```
#define __ATOMIC_SEQ_CST 5
```

9.45.1.63 __BIGGEST_ALIGNMENT__

```
#define __BIGGEST_ALIGNMENT__ 8
```

9.45.1.64 __BITINT_MAXWIDTH__

```
#define __BITINT_MAXWIDTH__ 128
```

9.45.1.65 __block

```
#define __block __attribute__((__blocks__(byref)))
```

9.45.1.66 __BLOCKS__

```
#define __BLOCKS__ 1
```

9.45.1.67 __BOOL_WIDTH__

```
#define __BOOL_WIDTH__ 8
```

9.45.1.68 __BYTE_ORDER__

```
#define __BYTE_ORDER__ __ORDER_LITTLE_ENDIAN__
```

9.45.1.69 __CHAR16_TYPE__

```
#define __CHAR16_TYPE__ unsigned short
```

9.45.1.70 __CHAR32_TYPE__

```
#define __CHAR32_TYPE__ unsigned int
```

9.45.1.71 __CHAR_BIT__

```
#define __CHAR_BIT__ 8
```

9.45.1.72 __clang__

```
#define __clang__ 1
```

9.45.1.73 __CLANG_ATOMIC_BOOL_LOCK_FREE

```
#define __CLANG_ATOMIC_BOOL_LOCK_FREE 2
```

9.45.1.74 __CLANG_ATOMIC_CHAR16_T_LOCK_FREE

```
#define __CLANG_ATOMIC_CHAR16_T_LOCK_FREE 2
```

9.45.1.75 __CLANG_ATOMIC_CHAR32_T_LOCK_FREE

```
#define __CLANG_ATOMIC_CHAR32_T_LOCK_FREE 2
```

9.45.1.76 __CLANG_ATOMIC_CHAR_LOCK_FREE

```
#define __CLANG_ATOMIC_CHAR_LOCK_FREE 2
```

9.45.1.77 __CLANG_ATOMIC_INT_LOCK_FREE

```
#define __CLANG_ATOMIC_INT_LOCK_FREE 2
```

9.45.1.78 __CLANG_ATOMIC_LLONG_LOCK_FREE

```
#define __CLANG_ATOMIC_LLONG_LOCK_FREE 2
```

9.45.1.79 __CLANG_ATOMIC_LONG_LOCK_FREE

```
#define __CLANG_ATOMIC_LONG_LOCK_FREE 2
```

9.45.1.80 __CLANG_ATOMIC_POINTER_LOCK_FREE

```
#define __CLANG_ATOMIC_POINTER_LOCK_FREE 2
```

9.45.1.81 __CLANG_ATOMIC_SHORT_LOCK_FREE

```
#define __CLANG_ATOMIC_SHORT_LOCK_FREE 2
```

9.45.1.82 __CLANG_ATOMIC_WCHAR_T_LOCK_FREE

```
#define __CLANG_ATOMIC_WCHAR_T_LOCK_FREE 2
```

9.45.1.83 __clang_literal_encoding__

```
#define __clang_literal_encoding__ "UTF-8"
```

9.45.1.84 __clang_major__

```
#define __clang_major__ 17
```

9.45.1.85 __clang_minor__

```
#define __clang_minor__ 0
```

9.45.1.86 __clang_patchlevel__

```
#define __clang_patchlevel__ 0
```

9.45.1.87 __clang_version__

```
#define __clang_version__ "17.0.0 (clang-1700.0.13.3)"
```

9.45.1.88 __clang_wide_literal_encoding__

```
#define __clang_wide_literal_encoding__ "UTF-32"
```

9.45.1.89 __CONSTANT_CFSTRINGS__

```
#define __CONSTANT_CFSTRINGS__ 1
```

9.45.1.90 __cplusplus

```
#define __cplusplus 201703L
```

9.45.1.91 __cpp_aggregate_bases

```
#define __cpp_aggregate_bases 201603L
```

9.45.1.92 __cpp_aggregate_nsdmi

```
#define __cpp_aggregate_nsdmi 201304L
```

9.45.1.93 __cpp_alias_templates

```
#define __cpp_alias_templates 200704L
```

9.45.1.94 __cpp_aligned_new

```
#define __cpp_aligned_new 201606L
```

9.45.1.95 __cpp_attributes

```
#define __cpp_attributes 200809L
```

9.45.1.96 __cpp_binary_literals

```
#define __cpp_binary_literals 201304L
```

9.45.1.97 __cpp_capture_star_this

```
#define __cpp_capture_star_this 201603L
```

9.45.1.98 __cpp_constexpr

```
#define __cpp_constexpr 201603L
```

9.45.1.99 __cpp_constexpr_in_decltype

```
#define __cpp_constexpr_in_decltype 201711L
```

9.45.1.100 __cpp_decltype

```
#define __cpp_decltype 200707L
```

9.45.1.101 __cpp_decltype_auto

```
#define __cpp_decltype_auto 201304L
```

9.45.1.102 __cpp_deduction_guides

```
#define __cpp_deduction_guides 201703L
```

9.45.1.103 __cpp_delegating_constructors

```
#define __cpp_delegating_constructors 200604L
```

9.45.1.104 __cpp_deleted_function

```
#define __cpp_deleted_function 202403L
```

9.45.1.105 __cpp_digit_separators

```
#define __cpp_digit_separators 201309L
```

9.45.1.106 __cpp_enumerator_attributes

```
#define __cpp_enumerator_attributes 201411L
```

9.45.1.107 __cpp_exceptions

```
#define __cpp_exceptions 199711L
```

9.45.1.108 __cpp_fold_expressions

```
#define __cpp_fold_expressions 201603L
```

9.45.1.109 __cpp_generic_lambdas

```
#define __cpp_generic_lambdas 201304L
```

9.45.1.110 __cpp_guaranteed_copy_elision

```
#define __cpp_guaranteed_copy_elision 201606L
```

9.45.1.111 __cpp_hex_float

```
#define __cpp_hex_float 201603L
```

9.45.1.112 __cpp_if_constexpr

```
#define __cpp_if_constexpr 201606L
```

9.45.1.113 __cpp_impl_destroying_delete

```
#define __cpp_impl_destroying_delete 201806L
```

9.45.1.114 __cpp_inheriting_constructors

```
#define __cpp_inheriting_constructors 201511L
```

9.45.1.115 __cpp_init_captures

```
#define __cpp_init_captures 201304L
```

9.45.1.116 __cpp_initializer_lists

```
#define __cpp_initializer_lists 200806L
```

9.45.1.117 __cpp_inline_variables

```
#define __cpp_inline_variables 201606L
```

9.45.1.118 __cpp_lambdas

```
#define __cpp_lambdas 200907L
```

9.45.1.119 __cpp_named_character_escapes

```
#define __cpp_named_character_escapes 202207L
```

9.45.1.120 __cpp_namespace_attributes

```
#define __cpp_namespace_attributes 201411L
```

9.45.1.121 __cpp_nested_namespace_definitions

```
#define __cpp_nested_namespace_definitions 201411L
```

9.45.1.122 __cpp_noexcept_function_type

```
#define __cpp_noexcept_function_type 201510L
```

9.45.1.123 __cpp_nontype_template_args

```
#define __cpp_nontype_template_args 201411L
```

9.45.1.124 __cpp_nontype_template_parameter_auto

```
#define __cpp_nontype_template_parameter_auto 201606L
```

9.45.1.125 __cpp_nsdmi

```
#define __cpp_nsdmi 200809L
```

9.45.1.126 __cpp_pack_indexing

```
#define __cpp_pack_indexing 202311L
```

9.45.1.127 __cpp_placeholder_variables

```
#define __cpp_placeholder_variables 202306L
```

9.45.1.128 __cpp_range_based_for

```
#define __cpp_range_based_for 201603L
```

9.45.1.129 __cpp_raw_strings

```
#define __cpp_raw_strings 200710L
```

9.45.1.130 __cpp_ref_qualifiers

```
#define __cpp_ref_qualifiers 200710L
```

9.45.1.131 __cpp_return_type_deduction

```
#define __cpp_return_type_deduction 201304L
```

9.45.1.132 __cpp_rtti

```
#define __cpp_rtti 199711L
```

9.45.1.133 __cpp_rvalue_references

```
#define __cpp_rvalue_references 200610L
```

9.45.1.134 __cpp_static_assert

```
#define __cpp_static_assert 201411L
```

9.45.1.135 __cpp_static_call_operator

```
#define __cpp_static_call_operator 202207L
```

9.45.1.136 __cpp_structured_bindings

```
#define __cpp_structured_bindings 202403L
```

9.45.1.137 __cpp_template_auto

```
#define __cpp_template_auto 201606L
```

9.45.1.138 __cpp_template_template_args

```
#define __cpp_template_template_args 201611L
```

9.45.1.139 __cpp_threadsafe_static_init

```
#define __cpp_threadsafe_static_init 200806L
```

9.45.1.140 __cpp_unicode_characters

```
#define __cpp_unicode_characters 200704L
```

9.45.1.141 __cpp_unicode_literals

```
#define __cpp_unicode_literals 200710L
```

9.45.1.142 __cpp_user_defined_literals

```
#define __cpp_user_defined_literals 200809L
```

9.45.1.143 __cpp_variable_templates

```
#define __cpp_variable_templates 201304L
```

9.45.1.144 __cpp_variadic_templates

```
#define __cpp_variadic_templates 200704L
```

9.45.1.145 __cpp_variadic_using

```
#define __cpp_variadic_using 201611L
```

9.45.1.146 __DBL_DECIMAL_DIG__

```
#define __DBL_DECIMAL_DIG__ 17
```

9.45.1.147 __DBL_DENORM_MIN__

```
#define __DBL_DENORM_MIN__ 4.9406564584124654e-324
```

9.45.1.148 __DBL_DIG__

```
#define __DBL_DIG__ 15
```

9.45.1.149 __DBL_EPSILON__

```
#define __DBL_EPSILON__ 2.2204460492503131e-16
```

9.45.1.150 __DBL_HAS_DENORM__

```
#define __DBL_HAS_DENORM__ 1
```

9.45.1.151 __DBL_HAS_INFINITY__

```
#define __DBL_HAS_INFINITY__ 1
```

9.45.1.152 __DBL_HAS_QUIET_NAN__

```
#define __DBL_HAS_QUIET_NAN__ 1
```

9.45.1.153 __DBL_MANT_DIG__

```
#define __DBL_MANT_DIG__ 53
```

9.45.1.154 __DBL_MAX_10_EXP__

```
#define __DBL_MAX_10_EXP__ 308
```

9.45.1.155 __DBL_MAX__

```
#define __DBL_MAX__ 1.7976931348623157e+308
```

9.45.1.156 __DBL_MAX_EXP__

```
#define __DBL_MAX_EXP__ 1024
```

9.45.1.157 __DBL_MIN_10_EXP__

```
#define __DBL_MIN_10_EXP__ (-307)
```

9.45.1.158 __DBL_MIN__

```
#define __DBL_MIN__ 2.2250738585072014e-308
```

9.45.1.159 __DBL_MIN_EXP__

```
#define __DBL_MIN_EXP__ (-1021)
```

9.45.1.160 __DBL_NORM_MAX__

```
#define __DBL_NORM_MAX__ 1.7976931348623157e+308
```

9.45.1.161 __DECIMAL_DIG__

```
#define __DECIMAL_DIG__ __LDBL_DECIMAL_DIG__
```

9.45.1.162 __DEPRECATED

```
#define __DEPRECATED 1
```

9.45.1.163 __DYNAMIC__

```
#define __DYNAMIC__ 1
```

9.45.1.164 __ENVIRONMENT_MAC_OS_X_VERSION_MIN_REQUIRED__

```
#define __ENVIRONMENT_MAC_OS_X_VERSION_MIN_REQUIRED__ 150000
```

9.45.1.165 __ENVIRONMENT_OS_VERSION_MIN_REQUIRED__

```
#define __ENVIRONMENT_OS_VERSION_MIN_REQUIRED__ 150000
```

9.45.1.166 __EXCEPTIONS

```
#define __EXCEPTIONS 1
```

9.45.1.167 __FINITE_MATH_ONLY__

```
#define __FINITE_MATH_ONLY__ 0
```

9.45.1.168 __FLT16_DECIMAL_DIG__

```
#define __FLT16_DECIMAL_DIG__ 5
```

9.45.1.169 __FLT16_DENORM_MIN__

```
#define __FLT16_DENORM_MIN__ 5.9604644775390625e-8F16
```

9.45.1.170 __FLT16_DIG__

```
#define __FLT16_DIG__ 3
```

9.45.1.171 __FLT16_EPSILON__

```
#define __FLT16_EPSILON__ 9.765625e-4F16
```

9.45.1.172 __FLT16_HAS_DENORM__

```
#define __FLT16_HAS_DENORM__ 1
```

9.45.1.173 __FLT16_HAS_INFINITY__

```
#define __FLT16_HAS_INFINITY__ 1
```

9.45.1.174 __FLT16_HAS_QUIET_NAN__

```
#define __FLT16_HAS_QUIET_NAN__ 1
```

9.45.1.175 __FLT16_MANT_DIG__

```
#define __FLT16_MANT_DIG__ 11
```

9.45.1.176 __FLT16_MAX_10_EXP__

```
#define __FLT16_MAX_10_EXP__ 4
```

9.45.1.177 __FLT16_MAX__

```
#define __FLT16_MAX__ 6.5504e+4F16
```

9.45.1.178 __FLT16_MAX_EXP__

```
#define __FLT16_MAX_EXP__ 16
```

9.45.1.179 __FLT16_MIN_10_EXP__

```
#define __FLT16_MIN_10_EXP__ (-4)
```

9.45.1.180 __FLT16_MIN__

```
#define __FLT16_MIN__ 6.103515625e-5F16
```

9.45.1.181 __FLT16_MIN_EXP__

```
#define __FLT16_MIN_EXP__ (-13)
```

9.45.1.182 __FLT16_NORM_MAX__

```
#define __FLT16_NORM_MAX__ 6.5504e+4F16
```

9.45.1.183 __FLT_DECIMAL_DIG__

```
#define __FLT_DECIMAL_DIG__ 9
```

9.45.1.184 __FLT_DENORM_MIN__

```
#define __FLT_DENORM_MIN__ 1.40129846e-45F
```

9.45.1.185 __FLT_DIG__

```
#define __FLT_DIG__ 6
```

9.45.1.186 __FLT_EPSILON__

```
#define __FLT_EPSILON__ 1.19209290e-7F
```

9.45.1.187 __FLT_HAS_DENORM__

```
#define __FLT_HAS_DENORM__ 1
```

9.45.1.188 __FLT_HAS_INFINITY__

```
#define __FLT_HAS_INFINITY__ 1
```

9.45.1.189 __FLT_HAS_QUIET_NAN__

```
#define __FLT_HAS_QUIET_NAN__ 1
```

9.45.1.190 __FLT_MANT_DIG__

```
#define __FLT_MANT_DIG__ 24
```

9.45.1.191 __FLT_MAX_10_EXP__

```
#define __FLT_MAX_10_EXP__ 38
```

9.45.1.192 __FLT_MAX__

```
#define __FLT_MAX__ 3.40282347e+38F
```

9.45.1.193 __FLT_MAX_EXP__

```
#define __FLT_MAX_EXP__ 128
```

9.45.1.194 __FLT_MIN_10_EXP__

```
#define __FLT_MIN_10_EXP__ (-37)
```

9.45.1.195 __FLT_MIN__

```
#define __FLT_MIN__ 1.17549435e-38F
```

9.45.1.196 __FLT_MIN_EXP__

```
#define __FLT_MIN_EXP__ (-125)
```

9.45.1.197 __FLT_NORM_MAX__

```
#define __FLT_NORM_MAX__ 3.40282347e+38F
```

9.45.1.198 __FLT_RADIX__

```
#define __FLT_RADIX__ 2
```

9.45.1.199 __FP_FAST_FMA

```
#define __FP_FAST_FMA 1
```

9.45.1.200 __FP_FAST_FMAF

```
#define __FP_FAST_FMAF 1
```

9.45.1.201 __FPCLASS_NEGINF

```
#define __FPCLASS_NEGINF 0x0004
```

9.45.1.202 __FPCLASS_NEGNORMAL

```
#define __FPCLASS_NEGNORMAL 0x0008
```

9.45.1.203 __FPCLASS_NEGSUBNORMAL

```
#define __FPCLASS_NEGSUBNORMAL 0x0010
```

9.45.1.204 __FPCLASS_NEGZERO

```
#define __FPCLASS_NEGZERO 0x0020
```

9.45.1.205 __FPCLASS_POSINF

```
#define __FPCLASS_POSINF 0x0200
```

9.45.1.206 __FPCLASS_POSNORMAL

```
#define __FPCLASS_POSNORMAL 0x0100
```

9.45.1.207 __FPCLASS_POSSUBNORMAL

```
#define __FPCLASS_POSSUBNORMAL 0x0080
```

9.45.1.208 __FPCLASS_POSZERO

```
#define __FPCLASS_POSZERO 0x0040
```

9.45.1.209 __FPCLASS_QNAN

```
#define __FPCLASS_QNAN 0x0002
```

9.45.1.210 __FPCLASS_SNAN

```
#define __FPCLASS_SNAN 0x0001
```

9.45.1.211 __GCC_ASM_FLAG_OUTPUTS__

```
#define __GCC_ASM_FLAG_OUTPUTS__ 1
```

9.45.1.212 __GCC_ATOMIC_BOOL_LOCK_FREE

```
#define __GCC_ATOMIC_BOOL_LOCK_FREE 2
```

9.45.1.213 __GCC_ATOMIC_CHAR16_T_LOCK_FREE

```
#define __GCC_ATOMIC_CHAR16_T_LOCK_FREE 2
```

9.45.1.214 __GCC_ATOMIC_CHAR32_T_LOCK_FREE

```
#define __GCC_ATOMIC_CHAR32_T_LOCK_FREE 2
```

9.45.1.215 __GCC_ATOMIC_CHAR_LOCK_FREE

```
#define __GCC_ATOMIC_CHAR_LOCK_FREE 2
```

9.45.1.216 __GCC_ATOMIC_INT_LOCK_FREE

```
#define __GCC_ATOMIC_INT_LOCK_FREE 2
```

9.45.1.217 __GCC_ATOMIC_LLONG_LOCK_FREE

```
#define __GCC_ATOMIC_LLONG_LOCK_FREE 2
```

9.45.1.218 __GCC_ATOMIC_LONG_LOCK_FREE

```
#define __GCC_ATOMIC_LONG_LOCK_FREE 2
```

9.45.1.219 __GCC_ATOMIC_POINTER_LOCK_FREE

```
#define __GCC_ATOMIC_POINTER_LOCK_FREE 2
```

9.45.1.220 __GCC_ATOMIC_SHORT_LOCK_FREE

```
#define __GCC_ATOMIC_SHORT_LOCK_FREE 2
```

9.45.1.221 __GCC_ATOMIC_TEST_AND_SET_TRUEVAL

```
#define __GCC_ATOMIC_TEST_AND_SET_TRUEVAL 1
```

9.45.1.222 __GCC_ATOMIC_WCHAR_T_LOCK_FREE

```
#define __GCC_ATOMIC_WCHAR_T_LOCK_FREE 2
```

9.45.1.223 __GCC_CONSTRUCTIVE_SIZE

```
#define __GCC_CONSTRUCTIVE_SIZE 64
```

9.45.1.224 __GCC_DESTRUCTIVE_SIZE

```
#define __GCC_DESTRUCTIVE_SIZE 64
```

9.45.1.225 __GCC_HAVE_DWARF2_CFI_ASM

```
#define __GCC_HAVE_DWARF2_CFI_ASM 1
```

9.45.1.226 __GCC_HAVE_SYNC_COMPARE_AND_SWAP_1

```
#define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_1 1
```

9.45.1.227 __GCC_HAVE_SYNC_COMPARE_AND_SWAP_16

```
#define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_16 1
```

9.45.1.228 __GCC_HAVE_SYNC_COMPARE_AND_SWAP_2

```
#define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_2 1
```

9.45.1.229 __GCC_HAVE_SYNC_COMPARE_AND_SWAP_4

```
#define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_4 1
```

9.45.1.230 __GCC_HAVE_SYNC_COMPARE_AND_SWAP_8

```
#define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_8 1
```

9.45.1.231 __GLIBCXX_BITSIZE_INT_N_0

```
#define __GLIBCXX_BITSIZE_INT_N_0 128
```

9.45.1.232 __GLIBCXX_TYPE_INT_N_0

```
#define __GLIBCXX_TYPE_INT_N_0 __int128
```

9.45.1.233 __GNUC__

```
#define __GNUC__ 4
```

9.45.1.234 __GNUC_GNU_INLINE__

```
#define __GNUC_GNU_INLINE__ 1
```

9.45.1.235 __GNUC_MINOR__

```
#define __GNUC_MINOR__ 2
```

9.45.1.236 __GNUC_PATCHLEVEL__

```
#define __GNUC_PATCHLEVEL__ 1
```

9.45.1.237 __GNUG__

```
#define __GNUG__ 4
```

9.45.1.238 __GXX_ABI_VERSION

```
#define __GXX_ABI_VERSION 1002
```

9.45.1.239 __GXX_EXPERIMENTAL_CXX0X__

```
#define __GXX_EXPERIMENTAL_CXX0X__ 1
```

9.45.1.240 __GXX_RTTI

```
#define __GXX_RTTI 1
```

9.45.1.241 __GXX_WEAK__

```
#define __GXX_WEAK__ 1
```

9.45.1.242 __HAVE_FUNCTION_MULTI_VERSIONING

```
#define __HAVE_FUNCTION_MULTI_VERSIONING 1
```

9.45.1.243 __INT16_C_SUFFIX__

```
#define __INT16_C_SUFFIX__
```

9.45.1.244 __INT16_FMTd__

```
#define __INT16_FMTd__ "hd"
```

9.45.1.245 __INT16_FMTi__

```
#define __INT16_FMTi__ "hi"
```

9.45.1.246 __INT16_MAX__

```
#define __INT16_MAX__ 32767
```

9.45.1.247 __INT16_TYPE__

```
#define __INT16_TYPE__ short
```

9.45.1.248 __INT32_C_SUFFIX__

```
#define __INT32_C_SUFFIX__
```

9.45.1.249 __INT32_FMTd__

```
#define __INT32_FMTd__ "d"
```

9.45.1.250 __INT32_FMTi__

```
#define __INT32_FMTi__ "i"
```

9.45.1.251 __INT32_MAX__

```
#define __INT32_MAX__ 2147483647
```

9.45.1.252 __INT32_TYPE__

```
#define __INT32_TYPE__ int
```

9.45.1.253 __INT64_C_SUFFIX__

```
#define __INT64_C_SUFFIX__ LL
```

9.45.1.254 __INT64_FMTd__

```
#define __INT64_FMTd__ "lld"
```

9.45.1.255 __INT64_FMTi__

```
#define __INT64_FMTi__ "lli"
```

9.45.1.256 __INT64_MAX__

```
#define __INT64_MAX__ 9223372036854775807LL
```

9.45.1.257 __INT64_TYPE__

```
#define __INT64_TYPE__ long long int
```

9.45.1.258 __INT8_C_SUFFIX__

```
#define __INT8_C_SUFFIX__
```

9.45.1.259 __INT8_FMTd__

```
#define __INT8_FMTd__ "hhd"
```

9.45.1.260 __INT8_FMTi__

```
#define __INT8_FMTi__ "hhi"
```

9.45.1.261 __INT8_MAX__

```
#define __INT8_MAX__ 127
```

9.45.1.262 __INT8_TYPE__

```
#define __INT8_TYPE__ signed char
```

9.45.1.263 __INT_FAST16_FMTd__

```
#define __INT_FAST16_FMTd__ "hd"
```

9.45.1.264 __INT_FAST16_FMTi__

```
#define __INT_FAST16_FMTi__ "hi"
```

9.45.1.265 __INT_FAST16_MAX__

```
#define __INT_FAST16_MAX__ 32767
```

9.45.1.266 __INT_FAST16_TYPE__

```
#define __INT_FAST16_TYPE__ short
```

9.45.1.267 __INT_FAST16_WIDTH__

```
#define __INT_FAST16_WIDTH__ 16
```

9.45.1.268 __INT_FAST32_FMTd__

```
#define __INT_FAST32_FMTd__ "d"
```

9.45.1.269 __INT_FAST32_FMTi__

```
#define __INT_FAST32_FMTi__ "i"
```

9.45.1.270 __INT_FAST32_MAX__

```
#define __INT_FAST32_MAX__ 2147483647
```

9.45.1.271 __INT_FAST32_TYPE__

```
#define __INT_FAST32_TYPE__ int
```

9.45.1.272 __INT_FAST32_WIDTH__

```
#define __INT_FAST32_WIDTH__ 32
```

9.45.1.273 __INT_FAST64_FMTd__

```
#define __INT_FAST64_FMTd__ "lld"
```

9.45.1.274 __INT_FAST64_FMTi__

```
#define __INT_FAST64_FMTi__ "lli"
```

9.45.1.275 __INT_FAST64_MAX__

```
#define __INT_FAST64_MAX__ 9223372036854775807LL
```

9.45.1.276 __INT_FAST64_TYPE__

```
#define __INT_FAST64_TYPE__ long long int
```

9.45.1.277 __INT_FAST64_WIDTH__

```
#define __INT_FAST64_WIDTH__ 64
```

9.45.1.278 __INT_FAST8_FMTd__

```
#define __INT_FAST8_FMTd__ "hhd"
```

9.45.1.279 __INT_FAST8_FMTi__

```
#define __INT_FAST8_FMTi__ "hhi"
```

9.45.1.280 __INT_FAST8_MAX__

```
#define __INT_FAST8_MAX__ 127
```

9.45.1.281 __INT_FAST8_TYPE__

```
#define __INT_FAST8_TYPE__ signed char
```

9.45.1.282 __INT_FAST8_WIDTH__

```
#define __INT_FAST8_WIDTH__ 8
```

9.45.1.283 __INT_LEAST16_FMTd__

```
#define __INT_LEAST16_FMTd__ "hd"
```

9.45.1.284 __INT_LEAST16_FMTi__

```
#define __INT_LEAST16_FMTi__ "hi"
```

9.45.1.285 __INT_LEAST16_MAX__

```
#define __INT_LEAST16_MAX__ 32767
```

9.45.1.286 __INT_LEAST16_TYPE__

```
#define __INT_LEAST16_TYPE__ short
```

9.45.1.287 __INT_LEAST16_WIDTH__

```
#define __INT_LEAST16_WIDTH__ 16
```

9.45.1.288 __INT_LEAST32_FMTd__

```
#define __INT_LEAST32_FMTd__ "d"
```

9.45.1.289 __INT_LEAST32_FMTi__

```
#define __INT_LEAST32_FMTi__ "i"
```

9.45.1.290 __INT_LEAST32_MAX__

```
#define __INT_LEAST32_MAX__ 2147483647
```

9.45.1.291 __INT_LEAST32_TYPE__

```
#define __INT_LEAST32_TYPE__ int
```

9.45.1.292 __INT_LEAST32_WIDTH__

```
#define __INT_LEAST32_WIDTH__ 32
```

9.45.1.293 __INT_LEAST64_FMTd__

```
#define __INT_LEAST64_FMTd__ "lld"
```

9.45.1.294 __INT_LEAST64_FMTi__

```
#define __INT_LEAST64_FMTi__ "lli"
```

9.45.1.295 __INT_LEAST64_MAX__

```
#define __INT_LEAST64_MAX__ 9223372036854775807LL
```

9.45.1.296 __INT_LEAST64_TYPE__

```
#define __INT_LEAST64_TYPE__ long long int
```

9.45.1.297 __INT_LEAST64_WIDTH__

```
#define __INT_LEAST64_WIDTH__ 64
```

9.45.1.298 __INT_LEAST8_FMTd__

```
#define __INT_LEAST8_FMTd__ "hhd"
```

9.45.1.299 __INT_LEAST8_FMTi__

```
#define __INT_LEAST8_FMTi__ "hhi"
```

9.45.1.300 __INT_LEAST8_MAX__

```
#define __INT_LEAST8_MAX__ 127
```

9.45.1.301 __INT_LEAST8_TYPE__

```
#define __INT_LEAST8_TYPE__ signed char
```

9.45.1.302 __INT_LEAST8_WIDTH__

```
#define __INT_LEAST8_WIDTH__ 8
```

9.45.1.303 __INT_MAX__

```
#define __INT_MAX__ 2147483647
```

9.45.1.304 __INT_WIDTH__

```
#define __INT_WIDTH__ 32
```

9.45.1.305 __INTMAX_C_SUFFIX__

```
#define __INTMAX_C_SUFFIX__ L
```

9.45.1.306 __INTMAX_FMTd__

```
#define __INTMAX_FMTd__ "ld"
```

9.45.1.307 __INTMAX_FMTi__

```
#define __INTMAX_FMTi__ "li"
```

9.45.1.308 __INTMAX_MAX__

```
#define __INTMAX_MAX__ 9223372036854775807L
```

9.45.1.309 __INTMAX_TYPE__

```
#define __INTMAX_TYPE__ long int
```

9.45.1.310 __INTMAX_WIDTH__

```
#define __INTMAX_WIDTH__ 64
```

9.45.1.311 __INTPTR_FMTd__

```
#define __INTPTR_FMTd__ "ld"
```

9.45.1.312 __INTPTR_FMTi__

```
#define __INTPTR_FMTi__ "li"
```

9.45.1.313 __INTPTR_MAX__

```
#define __INTPTR_MAX__ 9223372036854775807L
```

9.45.1.314 __INTPTR_TYPE__

```
#define __INTPTR_TYPE__ long int
```

9.45.1.315 __INTPTR_WIDTH__

```
#define __INTPTR_WIDTH__ 64
```

9.45.1.316 __LDBL_DECIMAL_DIG__

```
#define __LDBL_DECIMAL_DIG__ 17
```

9.45.1.317 __LDBL_DENORM_MIN__

```
#define __LDBL_DENORM_MIN__ 4.9406564584124654e-324L
```

9.45.1.318 __LDBL_DIG__

```
#define __LDBL_DIG__ 15
```

9.45.1.319 __LDBL_EPSILON__

```
#define __LDBL_EPSILON__ 2.2204460492503131e-16L
```

9.45.1.320 __LDBL_HAS_DENORM__

```
#define __LDBL_HAS_DENORM__ 1
```

9.45.1.321 __LDBL_HAS_INFINITY__

```
#define __LDBL_HAS_INFINITY__ 1
```

9.45.1.322 __LDBL_HAS_QUIET_NAN__

```
#define __LDBL_HAS_QUIET_NAN__ 1
```

9.45.1.323 __LDBL_MANT_DIG__

```
#define __LDBL_MANT_DIG__ 53
```

9.45.1.324 __LDBL_MAX_10_EXP__

```
#define __LDBL_MAX_10_EXP__ 308
```

9.45.1.325 __LDBL_MAX__

```
#define __LDBL_MAX__ 1.7976931348623157e+308L
```

9.45.1.326 __LDBL_MAX_EXP__

```
#define __LDBL_MAX_EXP__ 1024
```

9.45.1.327 __LDBL_MIN_10_EXP__

```
#define __LDBL_MIN_10_EXP__ (-307)
```

9.45.1.328 __LDBL_MIN__

```
#define __LDBL_MIN__ 2.2250738585072014e-308L
```

9.45.1.329 __LDBL_MIN_EXP__

```
#define __LDBL_MIN_EXP__ (-1021)
```

9.45.1.330 __LDBL_NORM_MAX__

```
#define __LDBL_NORM_MAX__ 1.7976931348623157e+308L
```

9.45.1.331 __LITTLE_ENDIAN__

```
#define __LITTLE_ENDIAN__ 1
```

9.45.1.332 __LLONG_WIDTH__

```
#define __LLONG_WIDTH__ 64
```

9.45.1.333 __llvm__

```
#define __llvm__ 1
```

9.45.1.334 __LONG_LONG_MAX__

```
#define __LONG_LONG_MAX__ 9223372036854775807LL
```

9.45.1.335 __LONG_MAX__

```
#define __LONG_MAX__ 9223372036854775807L
```

9.45.1.336 __LONG_WIDTH__

```
#define __LONG_WIDTH__ 64
```

9.45.1.337 __LP64__

```
#define __LP64__ 1
```

9.45.1.338 __MACH__

```
#define __MACH__ 1
```

9.45.1.339 __MEMORY_SCOPE_DEVICE

```
#define __MEMORY_SCOPE_DEVICE 1
```

9.45.1.340 __MEMORY_SCOPE_SINGLE

```
#define __MEMORY_SCOPE_SINGLE 4
```

9.45.1.341 __MEMORY_SCOPE_SYSTEM

```
#define __MEMORY_SCOPE_SYSTEM 0
```

9.45.1.342 __MEMORY_SCOPE_WRKGRP

```
#define __MEMORY_SCOPE_WRKGRP 2
```

9.45.1.343 __MEMORY_SCOPE_WVFRNT

```
#define __MEMORY_SCOPE_WVFRNT 3
```

9.45.1.344 __NO_INLINE__

```
#define __NO_INLINE__ 1
```

9.45.1.345 __NO_MATH_ERRNO__

```
#define __NO_MATH_ERRNO__ 1
```

9.45.1.346 __nonnull

```
#define __nonnull _Nonnull
```

9.45.1.347 __null_unspecified

```
#define __null_unspecified _Null_unspecified
```

9.45.1.348 __nullable

```
#define __nullable _Nullable
```

9.45.1.349 __OBJC_BOOL_IS_BOOL

```
#define __OBJC_BOOL_IS_BOOL 1
```

9.45.1.350 __OPENCL_MEMORY_SCOPE_ALL_SVM_DEVICES

```
#define __OPENCL_MEMORY_SCOPE_ALL_SVM_DEVICES 3
```

9.45.1.351 __OPENCL_MEMORY_SCOPE_DEVICE

```
#define __OPENCL_MEMORY_SCOPE_DEVICE 2
```

9.45.1.352 __OPENCL_MEMORY_SCOPE_SUB_GROUP

```
#define __OPENCL_MEMORY_SCOPE_SUB_GROUP 4
```

9.45.1.353 __OPENCL_MEMORY_SCOPE_WORK_GROUP

```
#define __OPENCL_MEMORY_SCOPE_WORK_GROUP 1
```

9.45.1.354 __OPENCL_MEMORY_SCOPE_WORK_ITEM

```
#define __OPENCL_MEMORY_SCOPE_WORK_ITEM 0
```

9.45.1.355 __ORDER_BIG_ENDIAN__

```
#define __ORDER_BIG_ENDIAN__ 4321
```

9.45.1.356 __ORDER_LITTLE_ENDIAN__

```
#define __ORDER_LITTLE_ENDIAN__ 1234
```

9.45.1.357 __ORDER_PDP_ENDIAN__

```
#define __ORDER_PDP_ENDIAN__ 3412
```

9.45.1.358 __PIC__

```
#define __PIC__ 2
```

9.45.1.359 __pic__

```
#define __pic__ 2
```

9.45.1.360 __POINTER_WIDTH__

```
#define __POINTER_WIDTH__ 64
```

9.45.1.361 __PRAGMA_REDEFINE_EXTNAME

```
#define __PRAGMA_REDEFINE_EXTNAME 1
```

9.45.1.362 __private_extern__

```
#define __private_extern__ extern
```

9.45.1.363 __PTRDIFF_FMTd__

```
#define __PTRDIFF_FMTd__ "ld"
```

9.45.1.364 __PTRDIFF_FMTi__

```
#define __PTRDIFF_FMTi__ "li"
```

9.45.1.365 __PTRDIFF_MAX__

```
#define __PTRDIFF_MAX__ 9223372036854775807L
```

9.45.1.366 __PTRDIFF_TYPE__

```
#define __PTRDIFF_TYPE__ long int
```

9.45.1.367 __PTRDIFF_WIDTH__

```
#define __PTRDIFF_WIDTH__ 64
```

9.45.1.368 __REGISTER_PREFIX__

```
#define __REGISTER_PREFIX__
```

9.45.1.369 __SCHAR_MAX__

```
#define __SCHAR_MAX__ 127
```

9.45.1.370 __SHRT_MAX__

```
#define __SHRT_MAX__ 32767
```

9.45.1.371 __SHRT_WIDTH__

```
#define __SHRT_WIDTH__ 16
```

9.45.1.372 __SIG_ATOMIC_MAX__

```
#define __SIG_ATOMIC_MAX__ 2147483647
```

9.45.1.373 __SIG_ATOMIC_WIDTH__

```
#define __SIG_ATOMIC_WIDTH__ 32
```

9.45.1.374 __SIZE_FMTo__

```
#define __SIZE_FMTo__ "lo"
```

9.45.1.375 __SIZE_FMTu__

```
#define __SIZE_FMTu__ "lu"
```

9.45.1.376 __SIZE_FMTX__

```
#define __SIZE_FMTX__ "lX"
```

9.45.1.377 __SIZE_FMTx__

```
#define __SIZE_FMTx__ "lx"
```

9.45.1.378 __SIZE_MAX__

```
#define __SIZE_MAX__ 18446744073709551615UL
```

9.45.1.379 __SIZE_TYPE__

```
#define __SIZE_TYPE__ long unsigned int
```

9.45.1.380 __SIZE_WIDTH__

```
#define __SIZE_WIDTH__ 64
```

9.45.1.381 __SIZEOF_DOUBLE__

```
#define __SIZEOF_DOUBLE__ 8
```

9.45.1.382 __SIZEOF_FLOAT__

```
#define __SIZEOF_FLOAT__ 4
```

9.45.1.383 __SIZEOF_INT128__

```
#define __SIZEOF_INT128__ 16
```

9.45.1.384 __SIZEOF_INT__

```
#define __SIZEOF_INT__ 4
```

9.45.1.385 __SIZEOF_LONG__

```
#define __SIZEOF_LONG__ 8
```

9.45.1.386 __SIZEOF_LONG_DOUBLE__

```
#define __SIZEOF_LONG_DOUBLE__ 8
```

9.45.1.387 __SIZEOF_LONG_LONG__

```
#define __SIZEOF_LONG_LONG__ 8
```

9.45.1.388 __SIZEOF_POINTER__

```
#define __SIZEOF_POINTER__ 8
```

9.45.1.389 __SIZEOF_PTRDIFF_T__

```
#define __SIZEOF_PTRDIFF_T__ 8
```

9.45.1.390 __SIZEOF_SHORT__

```
#define __SIZEOF_SHORT__ 2
```

9.45.1.391 __SIZEOF_SIZE_T__

```
#define __SIZEOF_SIZE_T__ 8
```

9.45.1.392 __SIZEOF_WCHAR_T__

```
#define __SIZEOF_WCHAR_T__ 4
```

9.45.1.393 __SIZEOF_WINT_T__

```
#define __SIZEOF_WINT_T__ 4
```

9.45.1.394 __SSP__

```
#define __SSP__ 1
```

9.45.1.395 __STDC__

```
#define __STDC__ 1
```

9.45.1.396 __STDC_EMBED_EMPTY__

```
#define __STDC_EMBED_EMPTY__ 2
```

9.45.1.397 __STDC_EMBED_FOUND__

```
#define __STDC_EMBED_FOUND__ 1
```

9.45.1.398 __STDC_EMBED_NOT_FOUND__

```
#define __STDC_EMBED_NOT_FOUND__ 0
```

9.45.1.399 __STDC_HOSTED__

```
#define __STDC_HOSTED__ 1
```

9.45.1.400 __STDC_NO_THREADS__

```
#define __STDC_NO_THREADS__ 1
```

9.45.1.401 __STDC_UTF_16__

```
#define __STDC_UTF_16__ 1
```

9.45.1.402 __STDC_UTF_32__

```
#define __STDC_UTF_32__ 1
```

9.45.1.403 __STDCPP_DEFAULT_NEW_ALIGNMENT__

```
#define __STDCPP_DEFAULT_NEW_ALIGNMENT__ 16UL
```

9.45.1.404 __STDCPP_THREADS__

```
#define __STDCPP_THREADS__ 1
```

9.45.1.405 __strong

```
#define __strong
```

9.45.1.406 __UINT16_C_SUFFIX__

```
#define __UINT16_C_SUFFIX__
```

9.45.1.407 __UINT16_FMTo__

```
#define __UINT16_FMTo__ "ho"
```

9.45.1.408 __UINT16_FMTu__

```
#define __UINT16_FMTu__ "hu"
```

9.45.1.409 __UINT16_FMTX__

```
#define __UINT16_FMTX__ "hX"
```

9.45.1.410 __UINT16_FMTx__

```
#define __UINT16_FMTx__ "hx"
```

9.45.1.411 __UINT16_MAX__

```
#define __UINT16_MAX__ 65535
```

9.45.1.412 __UINT16_TYPE__

```
#define __UINT16_TYPE__ unsigned short
```

9.45.1.413 __UINT32_C_SUFFIX__

```
#define __UINT32_C_SUFFIX__ U
```

9.45.1.414 __UINT32_FMTo__

```
#define __UINT32_FMTo__ "o"
```

9.45.1.415 __UINT32_FMTu__

```
#define __UINT32_FMTu__ "u"
```

9.45.1.416 __UINT32_FMTX__

```
#define __UINT32_FMTX__ "X"
```

9.45.1.417 __UINT32_FMTx__

```
#define __UINT32_FMTx__ "x"
```

9.45.1.418 __UINT32_MAX__

```
#define __UINT32_MAX__ 4294967295U
```

9.45.1.419 __UINT32_TYPE__

```
#define __UINT32_TYPE__ unsigned int
```

9.45.1.420 __UINT64_C_SUFFIX__

```
#define __UINT64_C_SUFFIX__ ULL
```

9.45.1.421 __UINT64_FMTo__

```
#define __UINT64_FMTo__ "llo"
```

9.45.1.422 __UINT64_FMTu__

```
#define __UINT64_FMTu__ "llu"
```

9.45.1.423 __UINT64_FMTX__

```
#define __UINT64_FMTX__ "llX"
```

9.45.1.424 __UINT64_FMTx__

```
#define __UINT64_FMTx__ "llx"
```

9.45.1.425 __UINT64_MAX__

```
#define __UINT64_MAX__ 18446744073709551615ULL
```

9.45.1.426 __UINT64_TYPE__

```
#define __UINT64_TYPE__ long long unsigned int
```

9.45.1.427 __UINT8_C_SUFFIX__

```
#define __UINT8_C_SUFFIX__
```

9.45.1.428 __UINT8_FMTo__

```
#define __UINT8_FMTo__ "hho"
```

9.45.1.429 __UINT8_FMTu__

```
#define __UINT8_FMTu__ "hhu"
```

9.45.1.430 __UINT8_FMTX__

```
#define __UINT8_FMTX__ "hhX"
```

9.45.1.431 __UINT8_FMTx__

```
#define __UINT8_FMTx__ "hhx"
```

9.45.1.432 __UINT8_MAX__

```
#define __UINT8_MAX__ 255
```

9.45.1.433 __UINT8_TYPE__

```
#define __UINT8_TYPE__ unsigned char
```

9.45.1.434 __UINT_FAST16_FMTo__

```
#define __UINT_FAST16_FMTo__ "ho"
```

9.45.1.435 __UINT_FAST16_FMTu__

```
#define __UINT_FAST16_FMTu__ "hu"
```

9.45.1.436 __UINT_FAST16_FMTX__

```
#define __UINT_FAST16_FMTX__ "hX"
```

9.45.1.437 __UINT_FAST16_FMTx__

```
#define __UINT_FAST16_FMTx__ "hx"
```

9.45.1.438 __UINT_FAST16_MAX__

```
#define __UINT_FAST16_MAX__ 65535
```

9.45.1.439 __UINT_FAST16_TYPE__

```
#define __UINT_FAST16_TYPE__ unsigned short
```

9.45.1.440 __UINT_FAST32_FMTo__

```
#define __UINT_FAST32_FMTo__ "o"
```

9.45.1.441 __UINT_FAST32_FMTu__

```
#define __UINT_FAST32_FMTu__ "u"
```

9.45.1.442 __UINT_FAST32_FMTX__

```
#define __UINT_FAST32_FMTX__ "X"
```

9.45.1.443 __UINT_FAST32_FMTx__

```
#define __UINT_FAST32_FMTx__ "x"
```

9.45.1.444 __UINT_FAST32_MAX__

```
#define __UINT_FAST32_MAX__ 4294967295U
```

9.45.1.445 __UINT_FAST32_TYPE__

```
#define __UINT_FAST32_TYPE__ unsigned int
```

9.45.1.446 __UINT_FAST64_FMTo__

```
#define __UINT_FAST64_FMTo__ "llo"
```

9.45.1.447 __UINT_FAST64_FMTu__

```
#define __UINT_FAST64_FMTu__ "llu"
```

9.45.1.448 __UINT_FAST64_FMTX__

```
#define __UINT_FAST64_FMTX__ "llX"
```

9.45.1.449 __UINT_FAST64_FMTx__

```
#define __UINT_FAST64_FMTx__ "llx"
```

9.45.1.450 __UINT_FAST64_MAX__

```
#define __UINT_FAST64_MAX__ 18446744073709551615ULL
```

9.45.1.451 __UINT_FAST64_TYPE__

```
#define __UINT_FAST64_TYPE__ long long unsigned int
```

9.45.1.452 __UINT_FAST8_FMTo__

```
#define __UINT_FAST8_FMTo__ "hho"
```

9.45.1.453 __UINT_FAST8_FMTu__

```
#define __UINT_FAST8_FMTu__ "hhu"
```

9.45.1.454 __UINT_FAST8_FMTX__

```
#define __UINT_FAST8_FMTX__ "hhX"
```

9.45.1.455 __UINT_FAST8_FMTx__

```
#define __UINT_FAST8_FMTx__ "hhx"
```

9.45.1.456 __UINT_FAST8_MAX__

```
#define __UINT_FAST8_MAX__ 255
```

9.45.1.457 __UINT_FAST8_TYPE__

```
#define __UINT_FAST8_TYPE__ unsigned char
```

9.45.1.458 __UINT_LEAST16_FMTo__

```
#define __UINT_LEAST16_FMTo__ "ho"
```

9.45.1.459 __UINT_LEAST16_FMTu__

```
#define __UINT_LEAST16_FMTu__ "hu"
```

9.45.1.460 __UINT_LEAST16_FMTX__

```
#define __UINT_LEAST16_FMTX__ "hX"
```

9.45.1.461 __UINT_LEAST16_FMTx__

```
#define __UINT_LEAST16_FMTx__ "hx"
```

9.45.1.462 __UINT_LEAST16_MAX__

```
#define __UINT_LEAST16_MAX__ 65535
```

9.45.1.463 __UINT_LEAST16_TYPE__

```
#define __UINT_LEAST16_TYPE__ unsigned short
```

9.45.1.464 __UINT_LEAST32_FMTo__

```
#define __UINT_LEAST32_FMTo__ "o"
```

9.45.1.465 __UINT_LEAST32_FMTu__

```
#define __UINT_LEAST32_FMTu__ "u"
```

9.45.1.466 __UINT_LEAST32_FMTX__

```
#define __UINT_LEAST32_FMTX__ "X"
```

9.45.1.467 __UINT_LEAST32_FMTx__

```
#define __UINT_LEAST32_FMTx__ "x"
```

9.45.1.468 __UINT_LEAST32_MAX__

```
#define __UINT_LEAST32_MAX__ 4294967295U
```

9.45.1.469 __UINT_LEAST32_TYPE__

```
#define __UINT_LEAST32_TYPE__ unsigned int
```

9.45.1.470 __UINT_LEAST64_FMTo__

```
#define __UINT_LEAST64_FMTo__ "llo"
```

9.45.1.471 __UINT_LEAST64_FMTu__

```
#define __UINT_LEAST64_FMTu__ "llu"
```

9.45.1.472 __UINT_LEAST64_FMTX__

```
#define __UINT_LEAST64_FMTX__ "llX"
```

9.45.1.473 __UINT_LEAST64_FMTx__

```
#define __UINT_LEAST64_FMTx__ "llx"
```

9.45.1.474 __UINT_LEAST64_MAX__

```
#define __UINT_LEAST64_MAX__ 18446744073709551615ULL
```

9.45.1.475 __UINT_LEAST64_TYPE__

```
#define __UINT_LEAST64_TYPE__ long long unsigned int
```

9.45.1.476 __UINT_LEAST8_FMTo__

```
#define __UINT_LEAST8_FMTo__ "hho"
```

9.45.1.477 __UINT_LEAST8_FMTu__

```
#define __UINT_LEAST8_FMTu__ "hhu"
```

9.45.1.478 __UINT_LEAST8_FMTX__

```
#define __UINT_LEAST8_FMTX__ "hhX"
```

9.45.1.479 __UINT_LEAST8_FMTx__

```
#define __UINT_LEAST8_FMTx__ "hhx"
```

9.45.1.480 __UINT_LEAST8_MAX__

```
#define __UINT_LEAST8_MAX__ 255
```

9.45.1.481 __UINT_LEAST8_TYPE__

```
#define __UINT_LEAST8_TYPE__ unsigned char
```

9.45.1.482 __UINTMAX_C_SUFFIX__

```
#define __UINTMAX_C_SUFFIX__ UL
```

9.45.1.483 __UINTMAX_FMTo__

```
#define __UINTMAX_FMTo__ "lo"
```

9.45.1.484 __UINTMAX_FMTu__

```
#define __UINTMAX_FMTu__ "lu"
```

9.45.1.485 __UINTMAX_FMTX__

```
#define __UINTMAX_FMTX__ "lX"
```

9.45.1.486 __UINTMAX_FMTx__

```
#define __UINTMAX_FMTx__ "lx"
```

9.45.1.487 __UINTMAX_MAX__

```
#define __UINTMAX_MAX__ 18446744073709551615UL
```

9.45.1.488 __UINTMAX_TYPE__

```
#define __UINTMAX_TYPE__ long unsigned int
```

9.45.1.489 __UINTMAX_WIDTH__

```
#define __UINTMAX_WIDTH__ 64
```

9.45.1.490 __UINTPTR_FMTo__

```
#define __UINTPTR_FMTo__ "lo"
```

9.45.1.491 __UINTPTR_FMTu__

```
#define __UINTPTR_FMTu__ "lu"
```

9.45.1.492 __UINTPTR_FMTX__

```
#define __UINTPTR_FMTX__ "lX"
```

9.45.1.493 __UINTPTR_FMTx__

```
#define __UINTPTR_FMTx__ "lx"
```

9.45.1.494 __UINTPTR_MAX__

```
#define __UINTPTR_MAX__ 18446744073709551615UL
```

9.45.1.495 __UINTPTR_TYPE__

```
#define __UINTPTR_TYPE__ long unsigned int
```

9.45.1.496 __UINTPTR_WIDTH__

```
#define __UINTPTR_WIDTH__ 64
```

9.45.1.497 __unsafe_unretained

```
#define __unsafe_unretained
```

9.45.1.498 __USER_LABEL_PREFIX__

```
#define __USER_LABEL_PREFIX__ _
```

9.45.1.499 __VERSION__

```
#define __VERSION__ "Apple LLVM 17.0.0 (clang-1700.0.13.3)"
```

9.45.1.500 __WCHAR_MAX__

```
#define __WCHAR_MAX__ 2147483647
```

9.45.1.501 __WCHAR_TYPE__

```
#define __WCHAR_TYPE__ int
```

9.45.1.502 __WCHAR_WIDTH__

```
#define __WCHAR_WIDTH__ 32
```

9.45.1.503 __weak

```
#define __weak __attribute__((objc_gc(weak)))
```

9.45.1.504 __WINT_MAX__

```
#define __WINT_MAX__ 2147483647
```

9.45.1.505 __WINT_TYPE__

```
#define __WINT_TYPE__ int
```

9.45.1.506 __WINT_WIDTH__

```
#define __WINT_WIDTH__ 32
```

9.45.1.507 _LP64

```
#define _LP64 1
```

9.45.1.508 QT_CHARTS_LIB

```
#define QT_CHARTS_LIB 1
```

9.45.1.509 QT_CHARTS_USE_NAMESPACE

```
#define QT_CHARTS_USE_NAMESPACE 1
```

9.45.1.510 QT_CONCURRENT_LIB

```
#define QT_CONCURRENT_LIB 1
```

9.45.1.511 QT_CORE_LIB

```
#define QT_CORE_LIB 1
```

9.45.1.512 QT_GUI_LIB

```
#define QT_GUI_LIB 1
```

9.45.1.513 QT_NETWORK_LIB

```
#define QT_NETWORK_LIB 1
```

9.45.1.514 QT_NO_DEBUG

```
#define QT_NO_DEBUG 1
```

9.45.1.515 QT_OPENGL_LIB

```
#define QT_OPENGL_LIB 1
```

9.45.1.516 QT_OPENGLWIDGETS_LIB

```
#define QT_OPENGLWIDGETS_LIB 1
```

9.45.1.517 QT_WIDGETS_LIB

```
#define QT_WIDGETS_LIB 1
```

9.45.1.518 SIZEOF_DPTR

```
#define SIZEOF_DPTR (sizeof(void*))
```

9.45.1.519 TARGET_IPHONE_SIMULATOR

```
#define TARGET_IPHONE_SIMULATOR 0
```

9.45.1.520 TARGET_OS_ARROW

```
#define TARGET_OS_ARROW 1
```

9.45.1.521 TARGET_OS_BRIDGE

```
#define TARGET_OS_BRIDGE 0
```

9.45.1.522 TARGET_OS_DRIVERKIT

```
#define TARGET_OS_DRIVERKIT 0
```

9.45.1.523 TARGET_OS_EMBEDDED

```
#define TARGET_OS_EMBEDDED 0
```

9.45.1.524 TARGET_OS_IOS

```
#define TARGET_OS_IOS 0
```

9.45.1.525 TARGET_OS_IOSMAC

```
#define TARGET_OS_IOSMAC 0
```

9.45.1.526 TARGET_OS_IPHONE

```
#define TARGET_OS_IPHONE 0
```

9.45.1.527 TARGET_OS_LINUX

```
#define TARGET_OS_LINUX 0
```

9.45.1.528 TARGET_OS_MAC

```
#define TARGET_OS_MAC 1
```

9.45.1.529 TARGET_OS_MACCATALYST

```
#define TARGET_OS_MACCATALYST 0
```

9.45.1.530 TARGET_OS_NANO

```
#define TARGET_OS_NANO 0
```

9.45.1.531 TARGET_OS_OSX

```
#define TARGET_OS_OSX 1
```

9.45.1.532 TARGET_OS_SIMULATOR

```
#define TARGET_OS_SIMULATOR 0
```

9.45.1.533 TARGET_OS_TV

```
#define TARGET_OS_TV 0
```

9.45.1.534 TARGET_OS_UIKITFORMAC

```
#define TARGET_OS_UIKITFORMAC 0
```

9.45.1.535 TARGET_OS_UNIX

```
#define TARGET_OS_UNIX 0
```

9.45.1.536 TARGET_OS_VISION

```
#define TARGET_OS_VISION 0
```

9.45.1.537 TARGET_OS_WATCH

```
#define TARGET_OS_WATCH 0
```

9.45.1.538 TARGET_OS_WIN32

```
#define TARGET_OS_WIN32 0
```

9.45.1.539 TARGET_OS_WINDOWS

```
#define TARGET_OS_WINDOWS 0
```

9.45.1.540 TARGET_OS_XR

```
#define TARGET_OS_XR 0
```

9.46 moc_predefs.h

浏览该文件的文档.

```
00001 #define QT_CHARTS_LIB 1
00002 #define QT_CHARTS_USE_NAMESPACE 1
00003 #define QT_CONCURRENT_LIB 1
00004 #define QT_CORE_LIB 1
00005 #define QT_GUI_LIB 1
00006 #define QT_NETWORK_LIB 1
00007 #define QT_NO_DEBUG 1
00008 #define QT_OPENGLWIDGETS_LIB 1
00009 #define QT_OPENGL_LIB 1
00010 #define QT_WIDGETS_LIB 1
00011 #define SIZEOF_DPTR (sizeof(void*))
00012 #define TARGET_IPHONE_SIMULATOR 0
00013 #define TARGET_OS_ARROW 1
00014 #define TARGET_OS_BRIDGE 0
00015 #define TARGET_OS_DRIVERKIT 0
00016 #define TARGET_OS_EMBEDDED 0
00017 #define TARGET_OS_IOS 0
00018 #define TARGET_OS_IOSMAC 0
00019 #define TARGET_OS_IPHONE 0
00020 #define TARGET_OS_LINUX 0
00021 #define TARGET_OS_MAC 1
00022 #define TARGET_OS_MACCATALYST 0
00023 #define TARGET_OS_NANO 0
00024 #define TARGET_OS_OSX 1
00025 #define TARGET_OS_SIMULATOR 0
00026 #define TARGET_OS_TV 0
00027 #define TARGET_OS_UIKITFORMAC 0
00028 #define TARGET_OS_UNIX 0
00029 #define TARGET_OS_VISION 0
```

```
00030 #define TARGET_OS_WATCH 0
00031 #define TARGET_OS_WIN32 0
00032 #define TARGET_OS_WINDOWS 0
00033 #define TARGET_OS_XR 0
00034 #define _LP64 1
00035 #define __AARCH64EL__ 1
00036 #define __AARCH64_CMODEL_SMALL__ 1
00037 #define __AARCH64_SIMD__ 1
00038 #define __APPLE_CC__ 6000
00039 #define __APPLE__ 1
00040 #define __ARM64_ARCH_8__ 1
00041 #define __ARM_64BIT_STATE 1
00042 #define __ARM_ACLE 200
00043 #define __ARM_ALIGN_MAX_STACK_PWR 4
00044 #define __ARM_ARCH 8
00045 #define __ARM_ARCH_8_3__ 1
00046 #define __ARM_ARCH_8_4__ 1
00047 #define __ARM_ARCH_8_5__ 1
00048 #define __ARM_ARCH_ISA_A64 1
00049 #define __ARM_ARCH_PROFILE 'A'
00050 #define __ARM_FEATURE_AES 1
00051 #define __ARM_FEATURE_ATOMICS 1
00052 #define __ARM_FEATURE_BTI 1
00053 #define __ARM_FEATURE_CLZ 1
00054 #define __ARM_FEATURE_COMPLEX 1
00055 #define __ARM_FEATURE_CRC32 1
00056 #define __ARM_FEATURE_CRYPTO 1
00057 #define __ARM_FEATURE_DIRECTED_ROUNDING 1
00058 #define __ARM_FEATURE_DIV 1
00059 #define __ARM_FEATURE_DOTPROD 1
00060 #define __ARM_FEATURE_FMA 1
00061 #define __ARM_FEATURE_FP16_FML 1
00062 #define __ARM_FEATURE_FP16_SCALAR_ARITHMETIC 1
00063 #define __ARM_FEATURE_FP16_VECTOR_ARITHMETIC 1
00064 #define __ARM_FEATURE_FRINT 1
00065 #define __ARM_FEATURE_IDIV 1
00066 #define __ARM_FEATURE_JCVT 1
00067 #define __ARM_FEATURE_LDREX 0xF
00068 #define __ARM_FEATURE_NUMERIC_MAXMIN 1
00069 #define __ARM_FEATURE_PAUTH 1
00070 #define __ARM_FEATURE_QRDMX 1
00071 #define __ARM_FEATURE_RCPC 1
00072 #define __ARM_FEATURE_SHA2 1
00073 #define __ARM_FEATURE_SHA3 1
00074 #define __ARM_FEATURE_SHA512 1
00075 #define __ARM_FEATURE_UNALIGNED 1
00076 #define __ARM_FP 0xE
00077 #define __ARM_FP16_ARGS 1
00078 #define __ARM_FP16_FORMAT_IEEE 1
00079 #define __ARM_NEON 1
00080 #define __ARM_NEON_FP 0xE
00081 #define __ARM_NEON__ 1
00082 #define __ARM_PCS_AAPCS64 1
00083 #define __ARM_SIZEOF_MINIMAL_ENUM 4
00084 #define __ARM_SIZEOF_WCHAR_T 4
00085 #define __ARM_STATE_ZA 1
00086 #define __ARM_STATE_ZT0 1
00087 #define __ATOMIC_ACQUIRE 2
00088 #define __ATOMIC_ACQ_REL 4
00089 #define __ATOMIC_CONSUME 1
00090 #define __ATOMIC_RELAXED 0
00091 #define __ATOMIC_RELEASE 3
00092 #define __ATOMIC_SEQ_CST 5
00093 #define __BIGGEST_ALIGNMENT__ 8
00094 #define __BITINT_MAXWIDTH__ 128
00095 #define __BLOCKS__ 1
00096 #define __BOOL_WIDTH__ 8
00097 #define __BYTE_ORDER__ __ORDER_LITTLE_ENDIAN__
00098 #define __CHAR16_TYPE__ unsigned short
00099 #define __CHAR32_TYPE__ unsigned int
00100 #define __CHAR_BIT__ 8
00101 #define __CLANG_ATOMIC_BOOL_LOCK_FREE 2
00102 #define __CLANG_ATOMIC_CHAR16_T_LOCK_FREE 2
00103 #define __CLANG_ATOMIC_CHAR32_T_LOCK_FREE 2
00104 #define __CLANG_ATOMIC_CHAR_LOCK_FREE 2
00105 #define __CLANG_ATOMIC_INT_LOCK_FREE 2
00106 #define __CLANG_ATOMIC_LLONG_LOCK_FREE 2
00107 #define __CLANG_ATOMIC_LONG_LOCK_FREE 2
00108 #define __CLANG_ATOMIC_POINTER_LOCK_FREE 2
00109 #define __CLANG_ATOMIC_SHORT_LOCK_FREE 2
00110 #define __CLANG_ATOMIC_WCHAR_T_LOCK_FREE 2
00111 #define __CONSTANT_CFSTRINGS__ 1
00112 #define __DBL_DECIMAL_DIG__ 17
00113 #define __DBL_DENORM_MIN__ 4.9406564584124654e-324
00114 #define __DBL_DIG__ 15
00115 #define __DBL_EPSILON__ 2.2204460492503131e-16
00116 #define __DBL_HAS_DENORM__ 1
```

```
00117 #define __DBL_HAS_INFINITY__ 1
00118 #define __DBL_HAS_QUIET_NAN__ 1
00119 #define __DBL_MANT_DIG__ 53
00120 #define __DBL_MAX_10_EXP__ 308
00121 #define __DBL_MAX_EXP__ 1024
00122 #define __DBL_MAX__ 1.7976931348623157e+308
00123 #define __DBL_MIN_10_EXP__ (-307)
00124 #define __DBL_MIN_EXP__ (-1021)
00125 #define __DBL_MIN__ 2.2250738585072014e-308
00126 #define __DBL_NORM_MAX__ 1.7976931348623157e+308
00127 #define __DECIMAL_DIG__ __LDBL_DECIMAL_DIG__
00128 #define __DEPRECATED 1
00129 #define __DYNAMIC__ 1
00130 #define __ENVIRONMENT_MAC_OS_X_VERSION_MIN_REQUIRED__ 150000
00131 #define __ENVIRONMENT_OS_VERSION_MIN_REQUIRED__ 150000
00132 #define __EXCEPTIONS 1
00133 #define __FINITE_MATH_ONLY__ 0
00134 #define __FLT16_DECIMAL_DIG__ 5
00135 #define __FLT16_DENORM_MIN__ 5.9604644775390625e-8F16
00136 #define __FLT16_DIG__ 3
00137 #define __FLT16_EPSILON__ 9.765625e-4F16
00138 #define __FLT16_HAS_DENORM__ 1
00139 #define __FLT16_HAS_INFINITY__ 1
00140 #define __FLT16_HAS_QUIET_NAN__ 1
00141 #define __FLT16_MANT_DIG__ 11
00142 #define __FLT16_MAX_10_EXP__ 4
00143 #define __FLT16_MAX_EXP__ 16
00144 #define __FLT16_MAX__ 6.5504e+4F16
00145 #define __FLT16_MIN_10_EXP__ (-4)
00146 #define __FLT16_MIN_EXP__ (-13)
00147 #define __FLT16_MIN__ 6.103515625e-5F16
00148 #define __FLT16_NORM_MAX__ 6.5504e+4F16
00149 #define __FLT_DECIMAL_DIG__ 9
00150 #define __FLT_DENORM_MIN__ 1.40129846e-45F
00151 #define __FLT_DIG__ 6
00152 #define __FLT_EPSILON__ 1.19209290e-7F
00153 #define __FLT_HAS_DENORM__ 1
00154 #define __FLT_HAS_INFINITY__ 1
00155 #define __FLT_HAS_QUIET_NAN__ 1
00156 #define __FLT_MANT_DIG__ 24
00157 #define __FLT_MAX_10_EXP__ 38
00158 #define __FLT_MAX_EXP__ 128
00159 #define __FLT_MAX__ 3.40282347e+38F
00160 #define __FLT_MIN_10_EXP__ (-37)
00161 #define __FLT_MIN_EXP__ (-125)
00162 #define __FLT_MIN__ 1.17549435e-38F
00163 #define __FLT_NORM_MAX__ 3.40282347e+38F
00164 #define __FLT_RADIX__ 2
00165 #define __FPCLASS_NEGINF 0x0004
00166 #define __FPCLASS_NEGNORMAL 0x0008
00167 #define __FPCLASS_NEGSUBNORMAL 0x0010
00168 #define __FPCLASS_NEGZERO 0x0020
00169 #define __FPCLASS_POSINF 0x0200
00170 #define __FPCLASS_POSNORMAL 0x0100
00171 #define __FPCLASS_POSSUBNORMAL 0x0080
00172 #define __FPCLASS_POSZERO 0x0040
00173 #define __FPCLASS_QNAN 0x0002
00174 #define __FPCLASS_SNAN 0x0001
00175 #define __FP_FAST_FMA 1
00176 #define __FP_FAST_FMAF 1
00177 #define __GCC_ASM_FLAG_OUTPUTS__ 1
00178 #define __GCC_ATOMIC_BOOL_LOCK_FREE 2
00179 #define __GCC_ATOMIC_CHAR16_T_LOCK_FREE 2
00180 #define __GCC_ATOMIC_CHAR32_T_LOCK_FREE 2
00181 #define __GCC_ATOMIC_CHAR_LOCK_FREE 2
00182 #define __GCC_ATOMIC_INT_LOCK_FREE 2
00183 #define __GCC_ATOMIC_LLONG_LOCK_FREE 2
00184 #define __GCC_ATOMIC_LONG_LOCK_FREE 2
00185 #define __GCC_ATOMIC_POINTER_LOCK_FREE 2
00186 #define __GCC_ATOMIC_SHORT_LOCK_FREE 2
00187 #define __GCC_ATOMIC_TEST_AND_SET_TRUEVAL 1
00188 #define __GCC_ATOMIC_WCHAR_T_LOCK_FREE 2
00189 #define __GCC_CONSTRUCTIVE_SIZE 64
00190 #define __GCC_DESTRUCTIVE_SIZE 64
00191 #define __GCC_HAVE_DWARF2_CFI_ASM 1
00192 #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_1 1
00193 #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_16 1
00194 #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_2 1
00195 #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_4 1
00196 #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_8 1
00197 #define __GLIBCXX_BITSIZE_INT_N_0 128
00198 #define __GLIBCXX_TYPE_INT_N_0 __int128
00199 #define __GNUC_GNU_INLINE__ 1
00200 #define __GNUC_MINOR__ 2
00201 #define __GNUC_PATCHLEVEL__ 1
00202 #define __GNUC__ 4
00203 #define __GNUG__ 4
```

```
00204 #define __GXX_ABI_VERSION 1002
00205 #define __GXX_EXPERIMENTAL_CXX0X__ 1
00206 #define __GXX_RTTI 1
00207 #define __GXX_WEAK__ 1
00208 #define __HAVE_FUNCTION_MULTI_VERSIONING 1
00209 #define __INT16_C_SUFFIX__
00210 #define __INT16_FMTd__ "hd"
00211 #define __INT16_FMTi__ "hi"
00212 #define __INT16_MAX__ 32767
00213 #define __INT16_TYPE__ short
00214 #define __INT32_C_SUFFIX__
00215 #define __INT32_FMTd__ "d"
00216 #define __INT32_FMTi__ "i"
00217 #define __INT32_MAX__ 2147483647
00218 #define __INT32_TYPE__ int
00219 #define __INT64_C_SUFFIX__ LL
00220 #define __INT64_FMTd__ "lld"
00221 #define __INT64_FMTi__ "lli"
00222 #define __INT64_MAX__ 9223372036854775807LL
00223 #define __INT64_TYPE__ long long int
00224 #define __INT8_C_SUFFIX__
00225 #define __INT8_FMTd__ "hhd"
00226 #define __INT8_FMTi__ "hhi"
00227 #define __INT8_MAX__ 127
00228 #define __INT8_TYPE__ signed char
00229 #define __INTMAX_C_SUFFIX__ L
00230 #define __INTMAX_FMTd__ "ld"
00231 #define __INTMAX_FMTi__ "li"
00232 #define __INTMAX_MAX__ 9223372036854775807L
00233 #define __INTMAX_TYPE__ long int
00234 #define __INTMAX_WIDTH__ 64
00235 #define __INTPTR_FMTd__ "ld"
00236 #define __INTPTR_FMTi__ "li"
00237 #define __INTPTR_MAX__ 9223372036854775807L
00238 #define __INTPTR_TYPE__ long int
00239 #define __INTPTR_WIDTH__ 64
00240 #define __INT_FAST16_FMTd__ "hd"
00241 #define __INT_FAST16_FMTi__ "hi"
00242 #define __INT_FAST16_MAX__ 32767
00243 #define __INT_FAST16_TYPE__ short
00244 #define __INT_FAST16_WIDTH__ 16
00245 #define __INT_FAST32_FMTd__ "d"
00246 #define __INT_FAST32_FMTi__ "i"
00247 #define __INT_FAST32_MAX__ 2147483647
00248 #define __INT_FAST32_TYPE__ int
00249 #define __INT_FAST32_WIDTH__ 32
00250 #define __INT_FAST64_FMTd__ "lld"
00251 #define __INT_FAST64_FMTi__ "lli"
00252 #define __INT_FAST64_MAX__ 9223372036854775807LL
00253 #define __INT_FAST64_TYPE__ long long int
00254 #define __INT_FAST64_WIDTH__ 64
00255 #define __INT_FAST8_FMTd__ "hhd"
00256 #define __INT_FAST8_FMTi__ "hhi"
00257 #define __INT_FAST8_MAX__ 127
00258 #define __INT_FAST8_TYPE__ signed char
00259 #define __INT_FAST8_WIDTH__ 8
00260 #define __INT_LEAST16_FMTd__ "hd"
00261 #define __INT_LEAST16_FMTi__ "hi"
00262 #define __INT_LEAST16_MAX__ 32767
00263 #define __INT_LEAST16_TYPE__ short
00264 #define __INT_LEAST16_WIDTH__ 16
00265 #define __INT_LEAST32_FMTd__ "d"
00266 #define __INT_LEAST32_FMTi__ "i"
00267 #define __INT_LEAST32_MAX__ 2147483647
00268 #define __INT_LEAST32_TYPE__ int
00269 #define __INT_LEAST32_WIDTH__ 32
00270 #define __INT_LEAST64_FMTd__ "lld"
00271 #define __INT_LEAST64_FMTi__ "lli"
00272 #define __INT_LEAST64_MAX__ 9223372036854775807LL
00273 #define __INT_LEAST64_TYPE__ long long int
00274 #define __INT_LEAST64_WIDTH__ 64
00275 #define __INT_LEAST8_FMTd__ "hhd"
00276 #define __INT_LEAST8_FMTi__ "hhi"
00277 #define __INT_LEAST8_MAX__ 127
00278 #define __INT_LEAST8_TYPE__ signed char
00279 #define __INT_LEAST8_WIDTH__ 8
00280 #define __INT_MAX__ 2147483647
00281 #define __INT_WIDTH__ 32
00282 #define __LDBL_DECIMAL_DIG__ 17
00283 #define __LDBL_DENORM_MIN__ 4.9406564584124654e-324L
00284 #define __LDBL_DIG__ 15
00285 #define __LDBL_EPSILON__ 2.2204460492503131e-16L
00286 #define __LDBL_HAS_DENORM__ 1
00287 #define __LDBL_HAS_INFINITY__ 1
00288 #define __LDBL_HAS_QUIET_NAN__ 1
00289 #define __LDBL_MANT_DIG__ 53
00290 #define __LDBL_MAX_10_EXP__ 308
```

```
00291 #define __LDBL_MAX_EXP__ 1024
00292 #define __LDBL_MAX__ 1.7976931348623157e+308L
00293 #define __LDBL_MIN_10_EXP__ (-307)
00294 #define __LDBL_MIN_EXP__ (-1021)
00295 #define __LDBL_MIN__ 2.2250738585072014e-308L
00296 #define __LDBL_NORM_MAX__ 1.7976931348623157e+308L
00297 #define __LITTLE_ENDIAN__ 1
00298 #define __LLONG_WIDTH__ 64
00299 #define __LONG_LONG_MAX__ 9223372036854775807LL
00300 #define __LONG_MAX__ 9223372036854775807L
00301 #define __LONG_WIDTH__ 64
00302 #define __LP64__ 1
00303 #define __MACH__ 1
00304 #define __MEMORY_SCOPE_DEVICE 1
00305 #define __MEMORY_SCOPE_SINGLE 4
00306 #define __MEMORY_SCOPE_SYSTEM 0
00307 #define __MEMORY_SCOPE_WRKGRP 2
00308 #define __MEMORY_SCOPE_WVFRNT 3
00309 #define __NO_INLINE__ 1
00310 #define __NO_MATH_ERRNO__ 1
00311 #define __OBJC_BOOL_IS_BOOL 1
00312 #define __OPENCL_MEMORY_SCOPE_ALL_SVM_DEVICES 3
00313 #define __OPENCL_MEMORY_SCOPE_DEVICE 2
00314 #define __OPENCL_MEMORY_SCOPE_SUB_GROUP 4
00315 #define __OPENCL_MEMORY_SCOPE_WORK_GROUP 1
00316 #define __OPENCL_MEMORY_SCOPE_WORK_ITEM 0
00317 #define __ORDER_BIG_ENDIAN__ 4321
00318 #define __ORDER_LITTLE_ENDIAN__ 1234
00319 #define __ORDER_PDP_ENDIAN__ 3412
00320 #define __PIC__ 2
00321 #define __POINTER_WIDTH__ 64
00322 #define __PRAGMA_REDEFINE_EXTNAME 1
00323 #define __PTRDIFF_FMTd__ "ld"
00324 #define __PTRDIFF_FMTi__ "li"
00325 #define __PTRDIFF_MAX__ 9223372036854775807L
00326 #define __PTRDIFF_TYPE__ long int
00327 #define __PTRDIFF_WIDTH__ 64
00328 #define __REGISTER_PREFIX__
00329 #define __SCHAR_MAX__ 127
00330 #define __SHRT_MAX__ 32767
00331 #define __SHRT_WIDTH__ 16
00332 #define __SIG_ATOMIC_MAX__ 2147483647
00333 #define __SIG_ATOMIC_WIDTH__ 32
00334 #define __SIZEOF_DOUBLE__ 8
00335 #define __SIZEOF_FLOAT__ 4
00336 #define __SIZEOF_INT128__ 16
00337 #define __SIZEOF_INT__ 4
00338 #define __SIZEOF_LONG_DOUBLE__ 8
00339 #define __SIZEOF_LONG_LONG__ 8
00340 #define __SIZEOF_LONG__ 8
00341 #define __SIZEOF_POINTER__ 8
00342 #define __SIZEOF_PTRDIFF_T__ 8
00343 #define __SIZEOF_SHORT__ 2
00344 #define __SIZEOF_SIZE_T__ 8
00345 #define __SIZEOF_WCHAR_T__ 4
00346 #define __SIZEOF_WINT_T__ 4
00347 #define __SIZE_FMTX__ "lX"
00348 #define __SIZE_FMTo__ "lo"
00349 #define __SIZE_FMTu__ "lu"
00350 #define __SIZE_FMTx__ "lx"
00351 #define __SIZE_MAX__ 18446744073709551615UL
00352 #define __SIZE_TYPE__ long unsigned int
00353 #define __SIZE_WIDTH__ 64
00354 #define __SSP__ 1
00355 #define __STDCPP_DEFAULT_NEW_ALIGNMENT__ 16UL
00356 #define __STDCPP_THREADS__ 1
00357 #define __STDC_EMBED_EMPTY__ 2
00358 #define __STDC_EMBED_FOUND__ 1
00359 #define __STDC_EMBED_NOT_FOUND__ 0
00360 #define __STDC_HOSTED__ 1
00361 #define __STDC_NO_THREADS__ 1
00362 #define __STDC_UTF_16__ 1
00363 #define __STDC_UTF_32__ 1
00364 #define __STDC__ 1
00365 #define __UINT16_C_SUFFIX__
00366 #define __UINT16_FMTX__ "hX"
00367 #define __UINT16_FMTo__ "ho"
00368 #define __UINT16_FMTu__ "hu"
00369 #define __UINT16_FMTx__ "hx"
00370 #define __UINT16_MAX__ 65535
00371 #define __UINT16_TYPE__ unsigned short
00372 #define __UINT32_C_SUFFIX__ U
00373 #define __UINT32_FMTX__ "X"
00374 #define __UINT32_FMTo__ "o"
00375 #define __UINT32_FMTu__ "u"
00376 #define __UINT32_FMTx__ "x"
00377 #define __UINT32_MAX__ 4294967295U
```

```
00378 #define __UINT32_TYPE__ unsigned int
00379 #define __UINT64_C_SUFFIX__ ULL
00380 #define __UINT64_FMTX__ "llX"
00381 #define __UINT64_FMTo__ "llo"
00382 #define __UINT64_FMTu__ "llu"
00383 #define __UINT64_FMTx__ "llx"
00384 #define __UINT64_MAX__ 18446744073709551615ULL
00385 #define __UINT64_TYPE__ long long unsigned int
00386 #define __UINT8_C_SUFFIX__
00387 #define __UINT8_FMTX__ "hhX"
00388 #define __UINT8_FMTo__ "hho"
00389 #define __UINT8_FMTu__ "hhu"
00390 #define __UINT8_FMTx__ "hhx"
00391 #define __UINT8_MAX__ 255
00392 #define __UINT8_TYPE__ unsigned char
00393 #define __UINTMAX_C_SUFFIX__ UL
00394 #define __UINTMAX_FMTX__ "lX"
00395 #define __UINTMAX_FMTo__ "lo"
00396 #define __UINTMAX_FMTu__ "lu"
00397 #define __UINTMAX_FMTx__ "lx"
00398 #define __UINTMAX_MAX__ 18446744073709551615UL
00399 #define __UINTMAX_TYPE__ long unsigned int
00400 #define __UINTMAX_WIDTH__ 64
00401 #define __UINTPTR_FMTX__ "lX"
00402 #define __UINTPTR_FMTo__ "lo"
00403 #define __UINTPTR_FMTu__ "lu"
00404 #define __UINTPTR_FMTx__ "lx"
00405 #define __UINTPTR_MAX__ 18446744073709551615UL
00406 #define __UINTPTR_TYPE__ long unsigned int
00407 #define __UINTPTR_WIDTH__ 64
00408 #define __UINT_FAST16_FMTX__ "hX"
00409 #define __UINT_FAST16_FMTo__ "ho"
00410 #define __UINT_FAST16_FMTu__ "hu"
00411 #define __UINT_FAST16_FMTx__ "hx"
00412 #define __UINT_FAST16_MAX__ 65535
00413 #define __UINT_FAST16_TYPE__ unsigned short
00414 #define __UINT_FAST32_FMTX__ "X"
00415 #define __UINT_FAST32_FMTo__ "o"
00416 #define __UINT_FAST32_FMTu__ "u"
00417 #define __UINT_FAST32_FMTx__ "x"
00418 #define __UINT_FAST32_MAX__ 4294967295U
00419 #define __UINT_FAST32_TYPE__ unsigned int
00420 #define __UINT_FAST64_FMTX__ "llX"
00421 #define __UINT_FAST64_FMTo__ "llo"
00422 #define __UINT_FAST64_FMTu__ "llu"
00423 #define __UINT_FAST64_FMTx__ "llx"
00424 #define __UINT_FAST64_MAX__ 18446744073709551615ULL
00425 #define __UINT_FAST64_TYPE__ long long unsigned int
00426 #define __UINT_FAST8_FMTX__ "hhX"
00427 #define __UINT_FAST8_FMTo__ "hho"
00428 #define __UINT_FAST8_FMTu__ "hhu"
00429 #define __UINT_FAST8_FMTx__ "hhx"
00430 #define __UINT_FAST8_MAX__ 255
00431 #define __UINT_FAST8_TYPE__ unsigned char
00432 #define __UINT_LEAST16_FMTX__ "hX"
00433 #define __UINT_LEAST16_FMTo__ "ho"
00434 #define __UINT_LEAST16_FMTu__ "hu"
00435 #define __UINT_LEAST16_FMTx__ "hx"
00436 #define __UINT_LEAST16_MAX__ 65535
00437 #define __UINT_LEAST16_TYPE__ unsigned short
00438 #define __UINT_LEAST32_FMTX__ "X"
00439 #define __UINT_LEAST32_FMTo__ "o"
00440 #define __UINT_LEAST32_FMTu__ "u"
00441 #define __UINT_LEAST32_FMTx__ "x"
00442 #define __UINT_LEAST32_MAX__ 4294967295U
00443 #define __UINT_LEAST32_TYPE__ unsigned int
00444 #define __UINT_LEAST64_FMTX__ "llX"
00445 #define __UINT_LEAST64_FMTo__ "llo"
00446 #define __UINT_LEAST64_FMTu__ "llu"
00447 #define __UINT_LEAST64_FMTx__ "llx"
00448 #define __UINT_LEAST64_MAX__ 18446744073709551615ULL
00449 #define __UINT_LEAST64_TYPE__ long long unsigned int
00450 #define __UINT_LEAST8_FMTX__ "hhX"
00451 #define __UINT_LEAST8_FMTo__ "hho"
00452 #define __UINT_LEAST8_FMTu__ "hhu"
00453 #define __UINT_LEAST8_FMTx__ "hhx"
00454 #define __UINT_LEAST8_MAX__ 255
00455 #define __UINT_LEAST8_TYPE__ unsigned char
00456 #define __USER_LABEL_PREFIX__ _
00457 #define __VERSION__ "Apple LLVM 17.0.0 (clang-1700.0.13.3)"
00458 #define __WCHAR_MAX__ 2147483647
00459 #define __WCHAR_TYPE__ int
00460 #define __WCHAR_WIDTH__ 32
00461 #define __WINT_MAX__ 2147483647
00462 #define __WINT_TYPE__ int
00463 #define __WINT_WIDTH__ 32
00464 #define __aarch64__ 1
```

```
00465 #define __apple_build_version__ 17000013
00466 #define __arm64 1
00467 #define __arm64__ 1
00468 #define __block __attribute__((__blocks__(byref)))
00469 #define __clang__ 1
00470 #define __clang_literal_encoding__ "UTF-8"
00471 #define __clang_major__ 17
00472 #define __clang_minor__ 0
00473 #define __clang_patchlevel__ 0
00474 #define __clang_version__ "17.0.0 (clang-1700.0.13.3)"
00475 #define __clang_wide_literal_encoding__ "UTF-32"
00476 #define __cplusplus 201703L
00477 #define __cpp_aggregate_bases 201603L
00478 #define __cpp_aggregate_nsdmi 201304L
00479 #define __cpp_alias_templates 200704L
00480 #define __cpp_aligned_new 201606L
00481 #define __cpp_attributes 200809L
00482 #define __cpp_binary_literals 201304L
00483 #define __cpp_capture_star_this 201603L
00484 #define __cpp_constexpr 201603L
00485 #define __cpp_constexpr_in_decltype 201711L
00486 #define __cpp_decltype 200707L
00487 #define __cpp_decltype_auto 201304L
00488 #define __cpp_deduction_guides 201703L
00489 #define __cpp_delegating_constructors 200604L
00490 #define __cpp_deleted_function 202403L
00491 #define __cpp_digit_separators 201309L
00492 #define __cpp_enumerator_attributes 201411L
00493 #define __cpp_exceptions 199711L
00494 #define __cpp_fold_expressions 201603L
00495 #define __cpp_generic_lambdas 201304L
00496 #define __cpp_guaranteed_copy_elision 201606L
00497 #define __cpp_hex_float 201603L
00498 #define __cpp_if_constexpr 201606L
00499 #define __cpp_impl_destroying_delete 201806L
00500 #define __cpp_inheriting_constructors 201511L
00501 #define __cpp_init_captures 201304L
00502 #define __cpp_initializer_lists 200806L
00503 #define __cpp_inline_variables 201606L
00504 #define __cpp_lambdas 200907L
00505 #define __cpp_named_character_escapes 202207L
00506 #define __cpp_namespace_attributes 201411L
00507 #define __cpp_nested_namespace_definitions 201411L
00508 #define __cpp_noexcept_function_type 201510L
00509 #define __cpp_nontype_template_args 201411L
00510 #define __cpp_nontype_template_parameter_auto 201606L
00511 #define __cpp_nsdmi 200809L
00512 #define __cpp_pack_indexing 202311L
00513 #define __cpp_placeholder_variables 202306L
00514 #define __cpp_range_based_for 201603L
00515 #define __cpp_raw_strings 200710L
00516 #define __cpp_ref_qualifiers 200710L
00517 #define __cpp_return_type_deduction 201304L
00518 #define __cpp_rtti 199711L
00519 #define __cpp_rvalue_references 200610L
00520 #define __cpp_static_assert 201411L
00521 #define __cpp_static_call_operator 202207L
00522 #define __cpp_structured_bindings 202403L
00523 #define __cpp_template_auto 201606L
00524 #define __cpp_template_template_args 201611L
00525 #define __cpp_threadsafe_static_init 200806L
00526 #define __cpp_unicode_characters 200704L
00527 #define __cpp_unicode_literals 200710L
00528 #define __cpp_user_defined_literals 200809L
00529 #define __cpp_variable_templates 201304L
00530 #define __cpp_variadic_templates 200704L
00531 #define __cpp_variadic_using 201611L
00532 #define __llvm__ 1
00533 #define __nonnull _Nonnull
00534 #define __null_unspecified _Null_unspecified
00535 #define __nullable _Nullable
00536 #define __pic__ 2
00537 #define __private_extern__ extern
00538 #define __strong
00539 #define __unsafe_unretained
00540 #define __weak __attribute__((objc_gc(weak)))
```

9.47 NetScanner_autogen/moc_predefs.h 文件参考

宏定义

- #define QT_CHARTS_LIB 1

- #define QT_CORE_LIB 1
- #define QT_GUI_LIB 1
- #define QT_NETWORK_LIB 1
- #define QT_NO_DEBUG 1
- #define QT_OPENGLWIDGETS_LIB 1
- #define QT_OPENGL_LIB 1
- #define QT_WIDGETS_LIB 1
- #define SIZEOF_DPTR (sizeof(void*))
- #define TARGET_IPHONE_SIMULATOR 0
- #define TARGET_OS_ARROW 1
- #define TARGET_OS_BRIDGE 0
- #define TARGET_OS_DRIVERKIT 0
- #define TARGET_OS_EMBEDDED 0
- #define TARGET_OS_IOS 0
- #define TARGET_OS_IOSMAC 0
- #define TARGET_OS_IPHONE 0
- #define TARGET_OS_LINUX 0
- #define TARGET_OS_MAC 1
- #define TARGET_OS_MACCATALYST 0
- #define TARGET_OS_NANO 0
- #define TARGET_OS_OSX 1
- #define TARGET_OS_SIMULATOR 0
- #define TARGET_OS_TV 0
- #define TARGET_OS_UIKITFORMAC 0
- #define TARGET_OS_UNIX 0
- #define TARGET_OS_VISION 0
- #define TARGET_OS_WATCH 0
- #define TARGET_OS_WIN32 0
- #define TARGET_OS_WINDOWS 0
- #define TARGET_OS_XR 0
- #define __LP64__ 1
- #define __AARCH64EL__ 1
- #define __AARCH64_CMODEL_SMALL__ 1
- #define __AARCH64_SIMD__ 1
- #define __APPLE_CC__ 6000
- #define __APPLE__ 1
- #define __ARM64_ARCH_8__ 1
- #define __ARM_64BIT_STATE 1
- #define __ARM_ACLE 200
- #define __ARM_ALIGN_MAX_STACK_PWR 4
- #define __ARM_ARCH 8
- #define __ARM_ARCH_8_3__ 1
- #define __ARM_ARCH_8_4__ 1
- #define __ARM_ARCH_8_5__ 1
- #define __ARM_ARCH_ISA_A64 1
- #define __ARM_ARCH_PROFILE 'A'
- #define __ARM_FEATURE_AES 1
- #define __ARM_FEATURE_ATOMICS 1
- #define __ARM_FEATURE_BTI 1
- #define __ARM_FEATURE_CLZ 1
- #define __ARM_FEATURE_COMPLEX 1
- #define __ARM_FEATURE_CRC32 1
- #define __ARM_FEATURE_CRYPTO 1
- #define __ARM_FEATURE_DIRECTED_ROUNDING 1
- #define __ARM_FEATURE_DIV 1

- #define __ARM_FEATURE_DOTPROD 1
- #define __ARM_FEATURE_FMA 1
- #define __ARM_FEATURE_FP16_FML 1
- #define __ARM_FEATURE_FP16_SCALAR_ARITHMETIC 1
- #define __ARM_FEATURE_FP16_VECTOR_ARITHMETIC 1
- #define __ARM_FEATURE_FRINT 1
- #define __ARM_FEATURE_IDIV 1
- #define __ARM_FEATURE_JCVT 1
- #define __ARM_FEATURE_LDREX 0xF
- #define __ARM_FEATURE_NUMERIC_MAXMIN 1
- #define __ARM_FEATURE_PAUTH 1
- #define __ARM_FEATURE_QRDMX 1
- #define __ARM_FEATURE_RCPC 1
- #define __ARM_FEATURE_SHA2 1
- #define __ARM_FEATURE_SHA3 1
- #define __ARM_FEATURE_SHA512 1
- #define __ARM_FEATURE_UNALIGNED 1
- #define __ARM_FP 0xE
- #define __ARM_FP16_ARGS 1
- #define __ARM_FP16_FORMAT_IEEE 1
- #define __ARM_NEON 1
- #define __ARM_NEON_FP 0xE
- #define __ARM_NEON__ 1
- #define __ARM_PCS_AAPCS64 1
- #define __ARM_SIZEOF_MINIMAL_ENUM 4
- #define __ARM_SIZEOF_WCHAR_T 4
- #define __ARM_STATE_ZA 1
- #define __ARM_STATE_ZT0 1
- #define __ATOMIC_ACQUIRE 2
- #define __ATOMIC_ACQ_REL 4
- #define __ATOMIC_CONSUME 1
- #define __ATOMIC_RELAXED 0
- #define __ATOMIC_RELEASE 3
- #define __ATOMIC_SEQ_CST 5
- #define __BIGGEST_ALIGNMENT__ 8
- #define __BITINT_MAXWIDTH__ 128
- #define __BLOCKS__ 1
- #define __BOOL_WIDTH__ 8
- #define __BYTE_ORDER__ __ORDER_LITTLE_ENDIAN__
- #define __CHAR16_TYPE__ unsigned short
- #define __CHAR32_TYPE__ unsigned int
- #define __CHAR_BIT__ 8
- #define __CLANG_ATOMIC_BOOL_LOCK_FREE 2
- #define __CLANG_ATOMIC_CHAR16_T_LOCK_FREE 2
- #define __CLANG_ATOMIC_CHAR32_T_LOCK_FREE 2
- #define __CLANG_ATOMIC_CHAR_LOCK_FREE 2
- #define __CLANG_ATOMIC_INT_LOCK_FREE 2
- #define __CLANG_ATOMIC_LLONG_LOCK_FREE 2
- #define __CLANG_ATOMIC_LONG_LOCK_FREE 2
- #define __CLANG_ATOMIC_POINTER_LOCK_FREE 2
- #define __CLANG_ATOMIC_SHORT_LOCK_FREE 2
- #define __CLANG_ATOMIC_WCHAR_T_LOCK_FREE 2
- #define __CONSTANT_CFSTRINGS__ 1
- #define __DBL_DECIMAL_DIG__ 17
- #define __DBL_DENORM_MIN__ 4.9406564584124654e-324

- #define __DBL_DIG__ 15
- #define __DBL_EPSILON__ 2.2204460492503131e-16
- #define __DBL_HAS_DENORM__ 1
- #define __DBL_HAS_INFINITY__ 1
- #define __DBL_HAS_QUIET_NAN__ 1
- #define __DBL_MANT_DIG__ 53
- #define __DBL_MAX_10_EXP__ 308
- #define __DBL_MAX_EXP__ 1024
- #define __DBL_MAX__ 1.7976931348623157e+308
- #define __DBL_MIN_10_EXP__ (-307)
- #define __DBL_MIN_EXP__ (-1021)
- #define __DBL_MIN__ 2.2250738585072014e-308
- #define __DBL_NORM_MAX__ 1.7976931348623157e+308
- #define __DECIMAL_DIG__ __LDBL_DECIMAL_DIG__
- #define __DEPRECATED 1
- #define __DYNAMIC__ 1
- #define __ENVIRONMENT_MAC_OS_X_VERSION_MIN_REQUIRED__ 150000
- #define __ENVIRONMENT_OS_VERSION_MIN_REQUIRED__ 150000
- #define __EXCEPTIONS 1
- #define __FINITE_MATH_ONLY__ 0
- #define __FLT16_DECIMAL_DIG__ 5
- #define __FLT16_DENORM_MIN__ 5.9604644775390625e-8F16
- #define __FLT16_DIG__ 3
- #define __FLT16_EPSILON__ 9.765625e-4F16
- #define __FLT16_HAS_DENORM__ 1
- #define __FLT16_HAS_INFINITY__ 1
- #define __FLT16_HAS_QUIET_NAN__ 1
- #define __FLT16_MANT_DIG__ 11
- #define __FLT16_MAX_10_EXP__ 4
- #define __FLT16_MAX_EXP__ 16
- #define __FLT16_MAX__ 6.5504e+4F16
- #define __FLT16_MIN_10_EXP__ (-4)
- #define __FLT16_MIN_EXP__ (-13)
- #define __FLT16_MIN__ 6.103515625e-5F16
- #define __FLT16_NORM_MAX__ 6.5504e+4F16
- #define __FLT_DECIMAL_DIG__ 9
- #define __FLT_DENORM_MIN__ 1.40129846e-45F
- #define __FLT_DIG__ 6
- #define __FLT_EPSILON__ 1.19209290e-7F
- #define __FLT_HAS_DENORM__ 1
- #define __FLT_HAS_INFINITY__ 1
- #define __FLT_HAS_QUIET_NAN__ 1
- #define __FLT_MANT_DIG__ 24
- #define __FLT_MAX_10_EXP__ 38
- #define __FLT_MAX_EXP__ 128
- #define __FLT_MAX__ 3.40282347e+38F
- #define __FLT_MIN_10_EXP__ (-37)
- #define __FLT_MIN_EXP__ (-125)
- #define __FLT_MIN__ 1.17549435e-38F
- #define __FLT_NORM_MAX__ 3.40282347e+38F
- #define __FLT_RADIX__ 2
- #define __FPCLASS_NEGINF 0x0004
- #define __FPCLASS_NEGNORMAL 0x0008
- #define __FPCLASS_NEGSUBNORMAL 0x0010
- #define __FPCLASS_NEGZERO 0x0020

- #define __FPCLASS_POSINF 0x0200
- #define __FPCLASS_POSNORMAL 0x0100
- #define __FPCLASS_POSSUBNORMAL 0x0080
- #define __FPCLASS_POSZERO 0x0040
- #define __FPCLASS_QNAN 0x0002
- #define __FPCLASS_SNAN 0x0001
- #define __FP_FAST_FMA 1
- #define __FP_FAST_FMAF 1
- #define __GCC_ASM_FLAG_OUTPUTS__ 1
- #define __GCC_ATOMIC_BOOL_LOCK_FREE 2
- #define __GCC_ATOMIC_CHAR16_T_LOCK_FREE 2
- #define __GCC_ATOMIC_CHAR32_T_LOCK_FREE 2
- #define __GCC_ATOMIC_CHAR_LOCK_FREE 2
- #define __GCC_ATOMIC_INT_LOCK_FREE 2
- #define __GCC_ATOMIC_LLONG_LOCK_FREE 2
- #define __GCC_ATOMIC_LONG_LOCK_FREE 2
- #define __GCC_ATOMIC_POINTER_LOCK_FREE 2
- #define __GCC_ATOMIC_SHORT_LOCK_FREE 2
- #define __GCC_ATOMIC_TEST_AND_SET_TRUEVAL 1
- #define __GCC_ATOMIC_WCHAR_T_LOCK_FREE 2
- #define __GCC_CONSTRUCTIVE_SIZE 64
- #define __GCC_DESTRUCTIVE_SIZE 64
- #define __GCC_HAVE_DWARF2_CFI_ASM 1
- #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_1 1
- #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_16 1
- #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_2 1
- #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_4 1
- #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_8 1
- #define __GLIBCXX_BITSIZE_INT_N_0 128
- #define __GLIBCXX_TYPE_INT_N_0 __int128
- #define __GNUC_GNU_INLINE__ 1
- #define __GNUC_MINOR__ 2
- #define __GNUC_PATCHLEVEL__ 1
- #define __GNUC__ 4
- #define __GNUG__ 4
- #define __GXX_ABI_VERSION 1002
- #define __GXX_EXPERIMENTAL_CXX0X__ 1
- #define __GXX_RTTI 1
- #define __GXX_WEAK__ 1
- #define __HAVE_FUNCTION_MULTI_VERSIONING 1
- #define __INT16_C_SUFFIX__
- #define __INT16_FMTd__ "hd"
- #define __INT16_FMTi__ "hi"
- #define __INT16_MAX__ 32767
- #define __INT16_TYPE__ short
- #define __INT32_C_SUFFIX__
- #define __INT32_FMTd__ "d"
- #define __INT32_FMTi__ "i"
- #define __INT32_MAX__ 2147483647
- #define __INT32_TYPE__ int
- #define __INT64_C_SUFFIX__ LL
- #define __INT64_FMTd__ "lld"
- #define __INT64_FMTi__ "lli"
- #define __INT64_MAX__ 9223372036854775807LL
- #define __INT64_TYPE__ long long int

- #define __INT8_C_SUFFIX__
- #define __INT8_FMTd__ "hhd"
- #define __INT8_FMTi__ "hhi"
- #define __INT8_MAX__ 127
- #define __INT8_TYPE__ signed char
- #define __INTMAX_C_SUFFIX__ L
- #define __INTMAX_FMTd__ "ld"
- #define __INTMAX_FMTi__ "li"
- #define __INTMAX_MAX__ 9223372036854775807L
- #define __INTMAX_TYPE__ long int
- #define __INTMAX_WIDTH__ 64
- #define __INTPTR_FMTd__ "ld"
- #define __INTPTR_FMTi__ "li"
- #define __INTPTR_MAX__ 9223372036854775807L
- #define __INTPTR_TYPE__ long int
- #define __INTPTR_WIDTH__ 64
- #define __INT_FAST16_FMTd__ "hd"
- #define __INT_FAST16_FMTi__ "hi"
- #define __INT_FAST16_MAX__ 32767
- #define __INT_FAST16_TYPE__ short
- #define __INT_FAST16_WIDTH__ 16
- #define __INT_FAST32_FMTd__ "d"
- #define __INT_FAST32_FMTi__ "i"
- #define __INT_FAST32_MAX__ 2147483647
- #define __INT_FAST32_TYPE__ int
- #define __INT_FAST32_WIDTH__ 32
- #define __INT_FAST64_FMTd__ "lld"
- #define __INT_FAST64_FMTi__ "lli"
- #define __INT_FAST64_MAX__ 9223372036854775807LL
- #define __INT_FAST64_TYPE__ long long int
- #define __INT_FAST64_WIDTH__ 64
- #define __INT_FAST8_FMTd__ "hhd"
- #define __INT_FAST8_FMTi__ "hhi"
- #define __INT_FAST8_MAX__ 127
- #define __INT_FAST8_TYPE__ signed char
- #define __INT_FAST8_WIDTH__ 8
- #define __INT_LEAST16_FMTd__ "hd"
- #define __INT_LEAST16_FMTi__ "hi"
- #define __INT_LEAST16_MAX__ 32767
- #define __INT_LEAST16_TYPE__ short
- #define __INT_LEAST16_WIDTH__ 16
- #define __INT_LEAST32_FMTd__ "d"
- #define __INT_LEAST32_FMTi__ "i"
- #define __INT_LEAST32_MAX__ 2147483647
- #define __INT_LEAST32_TYPE__ int
- #define __INT_LEAST32_WIDTH__ 32
- #define __INT_LEAST64_FMTd__ "lld"
- #define __INT_LEAST64_FMTi__ "lli"
- #define __INT_LEAST64_MAX__ 9223372036854775807LL
- #define __INT_LEAST64_TYPE__ long long int
- #define __INT_LEAST64_WIDTH__ 64
- #define __INT_LEAST8_FMTd__ "hhd"
- #define __INT_LEAST8_FMTi__ "hhi"
- #define __INT_LEAST8_MAX__ 127
- #define __INT_LEAST8_TYPE__ signed char

- #define __INT_LEAST8_WIDTH__ 8
- #define __INT_MAX__ 2147483647
- #define __INT_WIDTH__ 32
- #define __LDBL_DECIMAL_DIG__ 17
- #define __LDBL_DENORM_MIN__ 4.9406564584124654e-324L
- #define __LDBL_DIG__ 15
- #define __LDBL_EPSILON__ 2.2204460492503131e-16L
- #define __LDBL_HAS_DENORM__ 1
- #define __LDBL_HAS_INFINITY__ 1
- #define __LDBL_HAS_QUIET_NAN__ 1
- #define __LDBL_MANT_DIG__ 53
- #define __LDBL_MAX_10_EXP__ 308
- #define __LDBL_MAX_EXP__ 1024
- #define __LDBL_MAX__ 1.7976931348623157e+308L
- #define __LDBL_MIN_10_EXP__ (-307)
- #define __LDBL_MIN_EXP__ (-1021)
- #define __LDBL_MIN__ 2.2250738585072014e-308L
- #define __LDBL_NORM_MAX__ 1.7976931348623157e+308L
- #define __LITTLE_ENDIAN__ 1
- #define __LLONG_WIDTH__ 64
- #define __LONG_LONG_MAX__ 9223372036854775807LL
- #define __LONG_MAX__ 9223372036854775807L
- #define __LONG_WIDTH__ 64
- #define __LP64__ 1
- #define __MACH__ 1
- #define __MEMORY_SCOPE_DEVICE 1
- #define __MEMORY_SCOPE_SINGLE 4
- #define __MEMORY_SCOPE_SYSTEM 0
- #define __MEMORY_SCOPE_WRKGRP 2
- #define __MEMORY_SCOPE_WVFRNT 3
- #define __NO_INLINE__ 1
- #define __NO_MATH_ERRNO__ 1
- #define __OBJC_BOOL_IS_BOOL 1
- #define __OPENCL_MEMORY_SCOPE_ALL_SVM_DEVICES 3
- #define __OPENCL_MEMORY_SCOPE_DEVICE 2
- #define __OPENCL_MEMORY_SCOPE_SUB_GROUP 4
- #define __OPENCL_MEMORY_SCOPE_WORK_GROUP 1
- #define __OPENCL_MEMORY_SCOPE_WORK_ITEM 0
- #define __ORDER_BIG_ENDIAN__ 4321
- #define __ORDER_LITTLE_ENDIAN__ 1234
- #define __ORDER_PDP_ENDIAN__ 3412
- #define __PIC__ 2
- #define __POINTER_WIDTH__ 64
- #define __PRAGMA_REDEFINE_EXTNAME 1
- #define __PTRDIFF_FMTd__ "ld"
- #define __PTRDIFF_FMTi__ "li"
- #define __PTRDIFF_MAX__ 9223372036854775807L
- #define __PTRDIFF_TYPE__ long int
- #define __PTRDIFF_WIDTH__ 64
- #define __REGISTER_PREFIX__
- #define __SCHAR_MAX__ 127
- #define __SHRT_MAX__ 32767
- #define __SHRT_WIDTH__ 16
- #define __SIG_ATOMIC_MAX__ 2147483647
- #define __SIG_ATOMIC_WIDTH__ 32

- #define __SIZEOF_DOUBLE__ 8
- #define __SIZEOF_FLOAT__ 4
- #define __SIZEOF_INT128__ 16
- #define __SIZEOF_INT__ 4
- #define __SIZEOF_LONG_DOUBLE__ 8
- #define __SIZEOF_LONG_LONG__ 8
- #define __SIZEOF_LONG__ 8
- #define __SIZEOF_POINTER__ 8
- #define __SIZEOF_PTRDIFF_T__ 8
- #define __SIZEOF_SHORT__ 2
- #define __SIZEOF_SIZE_T__ 8
- #define __SIZEOF_WCHAR_T__ 4
- #define __SIZEOF_WINT_T__ 4
- #define __SIZE_FMTX__ "lX"
- #define __SIZE_FMTo__ "lo"
- #define __SIZE_FMTu__ "lu"
- #define __SIZE_FMTx__ "lx"
- #define __SIZE_MAX__ 18446744073709551615UL
- #define __SIZE_TYPE__ long unsigned int
- #define __SIZE_WIDTH__ 64
- #define __SSP__ 1
- #define __STDCPP_DEFAULT_NEW_ALIGNMENT__ 16UL
- #define __STDCPP_THREADS__ 1
- #define __STDC_EMBED_EMPTY__ 2
- #define __STDC_EMBED_FOUND__ 1
- #define __STDC_EMBED_NOT_FOUND__ 0
- #define __STDC_HOSTED__ 1
- #define __STDC_NO_THREADS__ 1
- #define __STDC_UTF_16__ 1
- #define __STDC_UTF_32__ 1
- #define __STDC__ 1
- #define __UINT16_C_SUFFIX__
- #define __UINT16_FMTX__ "hX"
- #define __UINT16_FMTo__ "ho"
- #define __UINT16_FMTu__ "hu"
- #define __UINT16_FMTx__ "hx"
- #define __UINT16_MAX__ 65535
- #define __UINT16_TYPE__ unsigned short
- #define __UINT32_C_SUFFIX__ U
- #define __UINT32_FMTX__ "X"
- #define __UINT32_FMTo__ "o"
- #define __UINT32_FMTu__ "u"
- #define __UINT32_FMTx__ "x"
- #define __UINT32_MAX__ 4294967295U
- #define __UINT32_TYPE__ unsigned int
- #define __UINT64_C_SUFFIX__ ULL
- #define __UINT64_FMTX__ "llX"
- #define __UINT64_FMTo__ "llo"
- #define __UINT64_FMTu__ "llu"
- #define __UINT64_FMTx__ "llx"
- #define __UINT64_MAX__ 18446744073709551615ULL
- #define __UINT64_TYPE__ long long unsigned int
- #define __UINT8_C_SUFFIX__
- #define __UINT8_FMTX__ "hhX"
- #define __UINT8_FMTo__ "hho"

- #define __UINT8_FMTu__ "hhu"
- #define __UINT8_FMTx__ "hhx"
- #define __UINT8_MAX__ 255
- #define __UINT8_TYPE__ unsigned char
- #define __UINTMAX_C_SUFFIX__ UL
- #define __UINTMAX_FMTX__ "lX"
- #define __UINTMAX_FMTo__ "lo"
- #define __UINTMAX_FMTu__ "lu"
- #define __UINTMAX_FMTx__ "lx"
- #define __UINTMAX_MAX__ 18446744073709551615UL
- #define __UINTMAX_TYPE__ long unsigned int
- #define __UINTMAX_WIDTH__ 64
- #define __UINTPTR_FMTX__ "lX"
- #define __UINTPTR_FMTo__ "lo"
- #define __UINTPTR_FMTu__ "lu"
- #define __UINTPTR_FMTx__ "lx"
- #define __UINTPTR_MAX__ 18446744073709551615UL
- #define __UINTPTR_TYPE__ long unsigned int
- #define __UINTPTR_WIDTH__ 64
- #define __UINT_FAST16_FMTX__ "hX"
- #define __UINT_FAST16_FMTo__ "ho"
- #define __UINT_FAST16_FMTu__ "hu"
- #define __UINT_FAST16_FMTx__ "hx"
- #define __UINT_FAST16_MAX__ 65535
- #define __UINT_FAST16_TYPE__ unsigned short
- #define __UINT_FAST32_FMTX__ "X"
- #define __UINT_FAST32_FMTo__ "o"
- #define __UINT_FAST32_FMTu__ "u"
- #define __UINT_FAST32_FMTx__ "x"
- #define __UINT_FAST32_MAX__ 4294967295U
- #define __UINT_FAST32_TYPE__ unsigned int
- #define __UINT_FAST64_FMTX__ "llX"
- #define __UINT_FAST64_FMTo__ "llo"
- #define __UINT_FAST64_FMTu__ "llu"
- #define __UINT_FAST64_FMTx__ "llx"
- #define __UINT_FAST64_MAX__ 18446744073709551615ULL
- #define __UINT_FAST64_TYPE__ long long unsigned int
- #define __UINT_FAST8_FMTX__ "hhX"
- #define __UINT_FAST8_FMTo__ "hho"
- #define __UINT_FAST8_FMTu__ "hhu"
- #define __UINT_FAST8_FMTx__ "hhx"
- #define __UINT_FAST8_MAX__ 255
- #define __UINT_FAST8_TYPE__ unsigned char
- #define __UINT_LEAST16_FMTX__ "hX"
- #define __UINT_LEAST16_FMTo__ "ho"
- #define __UINT_LEAST16_FMTu__ "hu"
- #define __UINT_LEAST16_FMTx__ "hx"
- #define __UINT_LEAST16_MAX__ 65535
- #define __UINT_LEAST16_TYPE__ unsigned short
- #define __UINT_LEAST32_FMTX__ "X"
- #define __UINT_LEAST32_FMTo__ "o"
- #define __UINT_LEAST32_FMTu__ "u"
- #define __UINT_LEAST32_FMTx__ "x"
- #define __UINT_LEAST32_MAX__ 4294967295U
- #define __UINT_LEAST32_TYPE__ unsigned int

- #define __UINT_LEAST64_FMTX__ "llX"
- #define __UINT_LEAST64_FMTo__ "llo"
- #define __UINT_LEAST64_FMTu__ "llu"
- #define __UINT_LEAST64_FMTx__ "llx"
- #define __UINT_LEAST64_MAX__ 18446744073709551615ULL
- #define __UINT_LEAST64_TYPE__ long long unsigned int
- #define __UINT_LEAST8_FMTX__ "hhX"
- #define __UINT_LEAST8_FMTo__ "hho"
- #define __UINT_LEAST8_FMTu__ "hhu"
- #define __UINT_LEAST8_FMTx__ "hhx"
- #define __UINT_LEAST8_MAX__ 255
- #define __UINT_LEAST8_TYPE__ unsigned char
- #define __USER_LABEL_PREFIX__ _
- #define __VERSION__ "Apple LLVM 17.0.0 (clang-1700.0.13.3)"
- #define __WCHAR_MAX__ 2147483647
- #define __WCHAR_TYPE__ int
- #define __WCHAR_WIDTH__ 32
- #define __WINT_MAX__ 2147483647
- #define __WINT_TYPE__ int
- #define __WINT_WIDTH__ 32
- #define __aarch64__ 1
- #define __apple_build_version__ 17000013
- #define __arm64 1
- #define __arm64__ 1
- #define __block __attribute__((__blocks__(byref)))
- #define __clang__ 1
- #define __clang_literal_encoding__ "UTF-8"
- #define __clang_major__ 17
- #define __clang_minor__ 0
- #define __clang_patchlevel__ 0
- #define __clang_version__ "17.0.0 (clang-1700.0.13.3)"
- #define __clang_wide_literal_encoding__ "UTF-32"
- #define __cplusplus 201703L
- #define __cpp_aggregate_bases 201603L
- #define __cpp_aggregate_nsdmi 201304L
- #define __cpp_alias_templates 200704L
- #define __cpp_aligned_new 201606L
- #define __cpp_attributes 200809L
- #define __cpp_binary_literals 201304L
- #define __cpp_capture_star_this 201603L
- #define __cpp_constexpr 201603L
- #define __cpp_constexpr_in_decltype 201711L
- #define __cpp_decltype 200707L
- #define __cpp_decltype_auto 201304L
- #define __cpp_deduction_guides 201703L
- #define __cpp_delegating_constructors 200604L
- #define __cpp_deleted_function 202403L
- #define __cpp_digit_separators 201309L
- #define __cpp_enumerator_attributes 201411L
- #define __cpp_exceptions 199711L
- #define __cpp_fold_expressions 201603L
- #define __cpp_generic_lambdas 201304L
- #define __cpp_guaranteed_copy_elision 201606L
- #define __cpp_hex_float 201603L
- #define __cpp_if_constexpr 201606L

- #define __cpp_impl_destroying_delete 201806L
- #define __cpp_inheriting_constructors 201511L
- #define __cpp_init_captures 201304L
- #define __cpp_initializer_lists 200806L
- #define __cpp_inline_variables 201606L
- #define __cpp_lambdas 200907L
- #define __cpp_named_character_escapes 202207L
- #define __cpp_namespace_attributes 201411L
- #define __cpp_nested_namespace_definitions 201411L
- #define __cpp_noexcept_function_type 201510L
- #define __cpp_nontype_template_args 201411L
- #define __cpp_nontype_template_parameter_auto 201606L
- #define __cpp_nsdmi 200809L
- #define __cpp_pack_indexing 202311L
- #define __cpp_placeholder_variables 202306L
- #define __cpp_range_based_for 201603L
- #define __cpp_raw_strings 200710L
- #define __cpp_ref_qualifiers 200710L
- #define __cpp_return_type_deduction 201304L
- #define __cpp_rtti 199711L
- #define __cpp_rvalue_references 200610L
- #define __cpp_static_assert 201411L
- #define __cpp_static_call_operator 202207L
- #define __cpp_structured_bindings 202403L
- #define __cpp_template_auto 201606L
- #define __cpp_template_template_args 201611L
- #define __cpp_threadsafe_static_init 200806L
- #define __cpp_unicode_characters 200704L
- #define __cpp_unicode_literals 200710L
- #define __cpp_user_defined_literals 200809L
- #define __cpp_variable_templates 201304L
- #define __cpp_variadic_templates 200704L
- #define __cpp_variadic_using 201611L
- #define __llvm__ 1
- #define __nonnull _Nonnull
- #define __null_unspecified _Null_unspecified
- #define __nullable _Nullable
- #define __pic__ 2
- #define __private_extern__ extern
- #define __strong
- #define __unsafe_unretained
- #define __weak __attribute__((objc_gc(weak)))

9.47.1 宏定义说明

9.47.1.1 __aarch64__

```
#define __aarch64__ 1
```

9.47.1.2 __AARCH64_CMODEL_SMALL__

```
#define __AARCH64_CMODEL_SMALL__ 1
```

9.47.1.3 __AARCH64_SIMD__

```
#define __AARCH64_SIMD__ 1
```

9.47.1.4 __AARCH64EL__

```
#define __AARCH64EL__ 1
```

9.47.1.5 __APPLE__

```
#define __APPLE__ 1
```

9.47.1.6 __apple_build_version__

```
#define __apple_build_version__ 17000013
```

9.47.1.7 __APPLE_CC__

```
#define __APPLE_CC__ 6000
```

9.47.1.8 __arm64

```
#define __arm64 1
```

9.47.1.9 __arm64__

```
#define __arm64__ 1
```

9.47.1.10 __ARM64_ARCH_8__

```
#define __ARM64_ARCH_8__ 1
```

9.47.1.11 __ARM_64BIT_STATE

```
#define __ARM_64BIT_STATE 1
```

9.47.1.12 __ARM_ACLE

```
#define __ARM_ACLE 200
```

9.47.1.13 __ARM_ALIGN_MAX_STACK_PWR

```
#define __ARM_ALIGN_MAX_STACK_PWR 4
```

9.47.1.14 __ARM_ARCH

```
#define __ARM_ARCH 8
```

9.47.1.15 __ARM_ARCH_8_3__

```
#define __ARM_ARCH_8_3__ 1
```

9.47.1.16 __ARM_ARCH_8_4__

```
#define __ARM_ARCH_8_4__ 1
```

9.47.1.17 __ARM_ARCH_8_5__

```
#define __ARM_ARCH_8_5__ 1
```

9.47.1.18 __ARM_ARCH_ISA_A64

```
#define __ARM_ARCH_ISA_A64 1
```

9.47.1.19 __ARM_ARCH_PROFILE

```
#define __ARM_ARCH_PROFILE 'A'
```

9.47.1.20 __ARM_FEATURE_AES

```
#define __ARM_FEATURE_AES 1
```

9.47.1.21 __ARM_FEATURE_ATOMICS

```
#define __ARM_FEATURE_ATOMICS 1
```

9.47.1.22 __ARM_FEATURE_BTI

```
#define __ARM_FEATURE_BTI 1
```

9.47.1.23 __ARM_FEATURE_CLZ

```
#define __ARM_FEATURE_CLZ 1
```

9.47.1.24 __ARM_FEATURE_COMPLEX

```
#define __ARM_FEATURE_COMPLEX 1
```

9.47.1.25 __ARM_FEATURE_CRC32

```
#define __ARM_FEATURE_CRC32 1
```

9.47.1.26 __ARM_FEATURE_CRYPTO

```
#define __ARM_FEATURE_CRYPTO 1
```

9.47.1.27 __ARM_FEATURE_DIRECTED_ROUNDING

```
#define __ARM_FEATURE_DIRECTED_ROUNDING 1
```

9.47.1.28 __ARM_FEATURE_DIV

```
#define __ARM_FEATURE_DIV 1
```

9.47.1.29 __ARM_FEATURE_DOTPROD

```
#define __ARM_FEATURE_DOTPROD 1
```

9.47.1.30 __ARM_FEATURE_FMA

```
#define __ARM_FEATURE_FMA 1
```

9.47.1.31 __ARM_FEATURE_FP16_FML

```
#define __ARM_FEATURE_FP16_FML 1
```

9.47.1.32 __ARM_FEATURE_FP16_SCALAR_ARITHMETIC

```
#define __ARM_FEATURE_FP16_SCALAR_ARITHMETIC 1
```

9.47.1.33 __ARM_FEATURE_FP16_VECTOR_ARITHMETIC

```
#define __ARM_FEATURE_FP16_VECTOR_ARITHMETIC 1
```

9.47.1.34 __ARM_FEATURE_FRINT

```
#define __ARM_FEATURE_FRINT 1
```

9.47.1.35 __ARM_FEATURE_IDIV

```
#define __ARM_FEATURE_IDIV 1
```

9.47.1.36 __ARM_FEATURE_JCVT

```
#define __ARM_FEATURE_JCVT 1
```

9.47.1.37 __ARM_FEATURE_LDREX

```
#define __ARM_FEATURE_LDREX 0xF
```

9.47.1.38 __ARM_FEATURE_NUMERIC_MAXMIN

```
#define __ARM_FEATURE_NUMERIC_MAXMIN 1
```

9.47.1.39 __ARM_FEATURE_PAUTH

```
#define __ARM_FEATURE_PAUTH 1
```

9.47.1.40 __ARM_FEATURE_QRDMX

```
#define __ARM_FEATURE_QRDMX 1
```

9.47.1.41 __ARM_FEATURE_RCPC

```
#define __ARM_FEATURE_RCPC 1
```

9.47.1.42 __ARM_FEATURE_SHA2

```
#define __ARM_FEATURE_SHA2 1
```

9.47.1.43 __ARM_FEATURE_SHA3

```
#define __ARM_FEATURE_SHA3 1
```

9.47.1.44 __ARM_FEATURE_SHA512

```
#define __ARM_FEATURE_SHA512 1
```

9.47.1.45 __ARM_FEATURE_UNALIGNED

```
#define __ARM_FEATURE_UNALIGNED 1
```

9.47.1.46 __ARM_FP

```
#define __ARM_FP 0xE
```

9.47.1.47 __ARM_FP16_ARGS

```
#define __ARM_FP16_ARGS 1
```

9.47.1.48 __ARM_FP16_FORMAT_IEEE

```
#define __ARM_FP16_FORMAT_IEEE 1
```

9.47.1.49 __ARM_NEON

```
#define __ARM_NEON 1
```

9.47.1.50 __ARM_NEON__

```
#define __ARM_NEON__ 1
```

9.47.1.51 __ARM_NEON_FP

```
#define __ARM_NEON_FP 0xE
```

9.47.1.52 __ARM_PCS_AAPCS64

```
#define __ARM_PCS_AAPCS64 1
```

9.47.1.53 __ARM_SIZEOF_MINIMAL_ENUM

```
#define __ARM_SIZEOF_MINIMAL_ENUM 4
```

9.47.1.54 __ARM_SIZEOF_WCHAR_T

```
#define __ARM_SIZEOF_WCHAR_T 4
```

9.47.1.55 __ARM_STATE_ZA

```
#define __ARM_STATE_ZA 1
```

9.47.1.56 __ARM_STATE_ZT0

```
#define __ARM_STATE_ZT0 1
```

9.47.1.57 __ATOMIC_ACQ_REL

```
#define __ATOMIC_ACQ_REL 4
```

9.47.1.58 __ATOMIC_ACQUIRE

```
#define __ATOMIC_ACQUIRE 2
```

9.47.1.59 __ATOMIC_CONSUME

```
#define __ATOMIC_CONSUME 1
```

9.47.1.60 __ATOMIC_RELAXED

```
#define __ATOMIC_RELAXED 0
```

9.47.1.61 __ATOMIC_RELEASE

```
#define __ATOMIC_RELEASE 3
```

9.47.1.62 __ATOMIC_SEQ_CST

```
#define __ATOMIC_SEQ_CST 5
```

9.47.1.63 __BIGGEST_ALIGNMENT__

```
#define __BIGGEST_ALIGNMENT__ 8
```

9.47.1.64 __BITINT_MAXWIDTH__

```
#define __BITINT_MAXWIDTH__ 128
```

9.47.1.65 __block

```
#define __block __attribute__((__blocks__(byref)))
```

9.47.1.66 __BLOCKS__

```
#define __BLOCKS__ 1
```

9.47.1.67 __BOOL_WIDTH__

```
#define __BOOL_WIDTH__ 8
```

9.47.1.68 __BYTE_ORDER__

```
#define __BYTE_ORDER__ __ORDER_LITTLE_ENDIAN__
```

9.47.1.69 __CHAR16_TYPE__

```
#define __CHAR16_TYPE__ unsigned short
```

9.47.1.70 __CHAR32_TYPE__

```
#define __CHAR32_TYPE__ unsigned int
```

9.47.1.71 __CHAR_BIT__

```
#define __CHAR_BIT__ 8
```

9.47.1.72 __clang__

```
#define __clang__ 1
```

9.47.1.73 __CLANG_ATOMIC_BOOL_LOCK_FREE

```
#define __CLANG_ATOMIC_BOOL_LOCK_FREE 2
```

9.47.1.74 __CLANG_ATOMIC_CHAR16_T_LOCK_FREE

```
#define __CLANG_ATOMIC_CHAR16_T_LOCK_FREE 2
```

9.47.1.75 __CLANG_ATOMIC_CHAR32_T_LOCK_FREE

```
#define __CLANG_ATOMIC_CHAR32_T_LOCK_FREE 2
```

9.47.1.76 __CLANG_ATOMIC_CHAR_LOCK_FREE

```
#define __CLANG_ATOMIC_CHAR_LOCK_FREE 2
```

9.47.1.77 __CLANG_ATOMIC_INT_LOCK_FREE

```
#define __CLANG_ATOMIC_INT_LOCK_FREE 2
```

9.47.1.78 __CLANG_ATOMIC_LLONG_LOCK_FREE

```
#define __CLANG_ATOMIC_LLONG_LOCK_FREE 2
```

9.47.1.79 __CLANG_ATOMIC_LONG_LOCK_FREE

```
#define __CLANG_ATOMIC_LONG_LOCK_FREE 2
```

9.47.1.80 __CLANG_ATOMIC_POINTER_LOCK_FREE

```
#define __CLANG_ATOMIC_POINTER_LOCK_FREE 2
```

9.47.1.81 __CLANG_ATOMIC_SHORT_LOCK_FREE

```
#define __CLANG_ATOMIC_SHORT_LOCK_FREE 2
```

9.47.1.82 __CLANG_ATOMIC_WCHAR_T_LOCK_FREE

```
#define __CLANG_ATOMIC_WCHAR_T_LOCK_FREE 2
```

9.47.1.83 __clang_literal_encoding__

```
#define __clang_literal_encoding__ "UTF-8"
```

9.47.1.84 __clang_major__

```
#define __clang_major__ 17
```

9.47.1.85 __clang_minor__

```
#define __clang_minor__ 0
```

9.47.1.86 __clang_patchlevel__

```
#define __clang_patchlevel__ 0
```

9.47.1.87 __clang_version__

```
#define __clang_version__ "17.0.0 (clang-1700.0.13.3)"
```

9.47.1.88 __clang_wide_literal_encoding__

```
#define __clang_wide_literal_encoding__ "UTF-32"
```

9.47.1.89 __CONSTANT_CFSTRINGS__

```
#define __CONSTANT_CFSTRINGS__ 1
```

9.47.1.90 __cplusplus

```
#define __cplusplus 201703L
```

9.47.1.91 __cpp_aggregate_bases

```
#define __cpp_aggregate_bases 201603L
```

9.47.1.92 __cpp_aggregate_nsdmi

```
#define __cpp_aggregate_nsdmi 201304L
```

9.47.1.93 __cpp_alias_templates

```
#define __cpp_alias_templates 200704L
```

9.47.1.94 __cpp_aligned_new

```
#define __cpp_aligned_new 201606L
```

9.47.1.95 __cpp_attributes

```
#define __cpp_attributes 200809L
```

9.47.1.96 __cpp_binary_literals

```
#define __cpp_binary_literals 201304L
```

9.47.1.97 __cpp_capture_star_this

```
#define __cpp_capture_star_this 201603L
```

9.47.1.98 __cpp_constexpr

```
#define __cpp_constexpr 201603L
```

9.47.1.99 __cpp_constexpr_in_decltype

```
#define __cpp_constexpr_in_decltype 201711L
```

9.47.1.100 __cpp_decltype

```
#define __cpp_decltype 200707L
```

9.47.1.101 __cpp_decltype_auto

```
#define __cpp_decltype_auto 201304L
```

9.47.1.102 __cpp_deduction_guides

```
#define __cpp_deduction_guides 201703L
```

9.47.1.103 __cpp_delegating_constructors

```
#define __cpp_delegating_constructors 200604L
```

9.47.1.104 __cpp_deleted_function

```
#define __cpp_deleted_function 202403L
```

9.47.1.105 __cpp_digit_separators

```
#define __cpp_digit_separators 201309L
```

9.47.1.106 __cpp_enumerator_attributes

```
#define __cpp_enumerator_attributes 201411L
```

9.47.1.107 __cpp_exceptions

```
#define __cpp_exceptions 199711L
```

9.47.1.108 __cpp_fold_expressions

```
#define __cpp_fold_expressions 201603L
```

9.47.1.109 __cpp_generic_lambdas

```
#define __cpp_generic_lambdas 201304L
```

9.47.1.110 __cpp_guaranteed_copy_elision

```
#define __cpp_guaranteed_copy_elision 201606L
```

9.47.1.111 __cpp_hex_float

```
#define __cpp_hex_float 201603L
```

9.47.1.112 __cpp_if_constexpr

```
#define __cpp_if_constexpr 201606L
```

9.47.1.113 __cpp_impl_destroying_delete

```
#define __cpp_impl_destroying_delete 201806L
```

9.47.1.114 __cpp_inheriting_constructors

```
#define __cpp_inheriting_constructors 201511L
```

9.47.1.115 __cpp_init_captures

```
#define __cpp_init_captures 201304L
```

9.47.1.116 __cpp_initializer_lists

```
#define __cpp_initializer_lists 200806L
```

9.47.1.117 __cpp_inline_variables

```
#define __cpp_inline_variables 201606L
```

9.47.1.118 __cpp_lambdas

```
#define __cpp_lambdas 200907L
```

9.47.1.119 __cpp_named_character_escapes

```
#define __cpp_named_character_escapes 202207L
```

9.47.1.120 __cpp_namespace_attributes

```
#define __cpp_namespace_attributes 201411L
```

9.47.1.121 __cpp_nested_namespace_definitions

```
#define __cpp_nested_namespace_definitions 201411L
```

9.47.1.122 __cpp_noexcept_function_type

```
#define __cpp_noexcept_function_type 201510L
```

9.47.1.123 __cpp_nontype_template_args

```
#define __cpp_nontype_template_args 201411L
```

9.47.1.124 __cpp_nontype_template_parameter_auto

```
#define __cpp_nontype_template_parameter_auto 201606L
```

9.47.1.125 __cpp_nsdmi

```
#define __cpp_nsdmi 200809L
```

9.47.1.126 __cpp_pack_indexing

```
#define __cpp_pack_indexing 202311L
```

9.47.1.127 __cpp_placeholder_variables

```
#define __cpp_placeholder_variables 202306L
```

9.47.1.128 __cpp_range_based_for

```
#define __cpp_range_based_for 201603L
```

9.47.1.129 __cpp_raw_strings

```
#define __cpp_raw_strings 200710L
```

9.47.1.130 __cpp_ref_qualifiers

```
#define __cpp_ref_qualifiers 200710L
```

9.47.1.131 __cpp_return_type_deduction

```
#define __cpp_return_type_deduction 201304L
```

9.47.1.132 __cpp_rtti

```
#define __cpp_rtti 199711L
```

9.47.1.133 __cpp_rvalue_references

```
#define __cpp_rvalue_references 200610L
```

9.47.1.134 __cpp_static_assert

```
#define __cpp_static_assert 201411L
```

9.47.1.135 __cpp_static_call_operator

```
#define __cpp_static_call_operator 202207L
```

9.47.1.136 __cpp_structured_bindings

```
#define __cpp_structured_bindings 202403L
```

9.47.1.137 __cpp_template_auto

```
#define __cpp_template_auto 201606L
```

9.47.1.138 __cpp_template_template_args

```
#define __cpp_template_template_args 201611L
```

9.47.1.139 __cpp_threadsafe_static_init

```
#define __cpp_threadsafe_static_init 200806L
```

9.47.1.140 __cpp_unicode_characters

```
#define __cpp_unicode_characters 200704L
```

9.47.1.141 __cpp_unicode_literals

```
#define __cpp_unicode_literals 200710L
```

9.47.1.142 __cpp_user_defined_literals

```
#define __cpp_user_defined_literals 200809L
```

9.47.1.143 __cpp_variable_templates

```
#define __cpp_variable_templates 201304L
```

9.47.1.144 __cpp_variadic_templates

```
#define __cpp_variadic_templates 200704L
```

9.47.1.145 __cpp_variadic_using

```
#define __cpp_variadic_using 201611L
```

9.47.1.146 __DBL_DECIMAL_DIG__

```
#define __DBL_DECIMAL_DIG__ 17
```

9.47.1.147 __DBL_DENORM_MIN__

```
#define __DBL_DENORM_MIN__ 4.9406564584124654e-324
```

9.47.1.148 __DBL_DIG__

```
#define __DBL_DIG__ 15
```

9.47.1.149 __DBL_EPSILON__

```
#define __DBL_EPSILON__ 2.220446049250313e-16
```

9.47.1.150 __DBL_HAS_DENORM__

```
#define __DBL_HAS_DENORM__ 1
```

9.47.1.151 __DBL_HAS_INFINITY__

```
#define __DBL_HAS_INFINITY__ 1
```

9.47.1.152 __DBL_HAS_QUIET_NAN__

```
#define __DBL_HAS_QUIET_NAN__ 1
```

9.47.1.153 __DBL_MANT_DIG__

```
#define __DBL_MANT_DIG__ 53
```

9.47.1.154 __DBL_MAX_10_EXP__

```
#define __DBL_MAX_10_EXP__ 308
```

9.47.1.155 __DBL_MAX__

```
#define __DBL_MAX__ 1.7976931348623157e+308
```

9.47.1.156 __DBL_MAX_EXP__

```
#define __DBL_MAX_EXP__ 1024
```

9.47.1.157 __DBL_MIN_10_EXP__

```
#define __DBL_MIN_10_EXP__ (-307)
```

9.47.1.158 __DBL_MIN__

```
#define __DBL_MIN__ 2.2250738585072014e-308
```

9.47.1.159 __DBL_MIN_EXP__

```
#define __DBL_MIN_EXP__ (-1021)
```

9.47.1.160 __DBL_NORM_MAX__

```
#define __DBL_NORM_MAX__ 1.7976931348623157e+308
```

9.47.1.161 __DECIMAL_DIG__

```
#define __DECIMAL_DIG__ __LDBL_DECIMAL_DIG__
```

9.47.1.162 __DEPRECATED

```
#define __DEPRECATED 1
```

9.47.1.163 __DYNAMIC__

```
#define __DYNAMIC__ 1
```

9.47.1.164 __ENVIRONMENT_MAC_OS_X_VERSION_MIN_REQUIRED__

```
#define __ENVIRONMENT_MAC_OS_X_VERSION_MIN_REQUIRED__ 150000
```

9.47.1.165 __ENVIRONMENT_OS_VERSION_MIN_REQUIRED__

```
#define __ENVIRONMENT_OS_VERSION_MIN_REQUIRED__ 150000
```

9.47.1.166 __EXCEPTIONS

```
#define __EXCEPTIONS 1
```

9.47.1.167 __FINITE_MATH_ONLY__

```
#define __FINITE_MATH_ONLY__ 0
```

9.47.1.168 __FLT16_DECIMAL_DIG__

```
#define __FLT16_DECIMAL_DIG__ 5
```

9.47.1.169 __FLT16_DENORM_MIN__

```
#define __FLT16_DENORM_MIN__ 5.960464477539063e-8F16
```

9.47.1.170 __FLT16_DIG__

```
#define __FLT16_DIG__ 3
```

9.47.1.171 __FLT16_EPSILON__

```
#define __FLT16_EPSILON__ 9.765625e-4F16
```

9.47.1.172 __FLT16_HAS_DENORM__

```
#define __FLT16_HAS_DENORM__ 1
```

9.47.1.173 __FLT16_HAS_INFINITY__

```
#define __FLT16_HAS_INFINITY__ 1
```

9.47.1.174 __FLT16_HAS_QUIET_NAN__

```
#define __FLT16_HAS_QUIET_NAN__ 1
```

9.47.1.175 __FLT16_MANT_DIG__

```
#define __FLT16_MANT_DIG__ 11
```

9.47.1.176 __FLT16_MAX_10_EXP__

```
#define __FLT16_MAX_10_EXP__ 4
```

9.47.1.177 __FLT16_MAX__

```
#define __FLT16_MAX__ 6.5504e+4F16
```

9.47.1.178 __FLT16_MAX_EXP__

```
#define __FLT16_MAX_EXP__ 16
```

9.47.1.179 __FLT16_MIN_10_EXP__

```
#define __FLT16_MIN_10_EXP__ (-4)
```

9.47.1.180 __FLT16_MIN__

```
#define __FLT16_MIN__ 6.103515625e-5F16
```

9.47.1.181 __FLT16_MIN_EXP__

```
#define __FLT16_MIN_EXP__ (-13)
```

9.47.1.182 __FLT16_NORM_MAX__

```
#define __FLT16_NORM_MAX__ 6.5504e+4F16
```

9.47.1.183 __FLT_DECIMAL_DIG__

```
#define __FLT_DECIMAL_DIG__ 9
```

9.47.1.184 __FLT_DENORM_MIN__

```
#define __FLT_DENORM_MIN__ 1.40129846e-45F
```

9.47.1.185 __FLT_DIG__

```
#define __FLT_DIG__ 6
```

9.47.1.186 __FLT_EPSILON__

```
#define __FLT_EPSILON__ 1.19209290e-7F
```

9.47.1.187 __FLT_HAS_DENORM__

```
#define __FLT_HAS_DENORM__ 1
```

9.47.1.188 __FLT_HAS_INFINITY__

```
#define __FLT_HAS_INFINITY__ 1
```

9.47.1.189 __FLT_HAS_QUIET_NAN__

```
#define __FLT_HAS_QUIET_NAN__ 1
```

9.47.1.190 __FLT_MANT_DIG__

```
#define __FLT_MANT_DIG__ 24
```

9.47.1.191 __FLT_MAX_10_EXP__

```
#define __FLT_MAX_10_EXP__ 38
```

9.47.1.192 __FLT_MAX__

```
#define __FLT_MAX__ 3.40282347e+38F
```

9.47.1.193 __FLT_MAX_EXP__

```
#define __FLT_MAX_EXP__ 128
```

9.47.1.194 __FLT_MIN_10_EXP__

```
#define __FLT_MIN_10_EXP__ (-37)
```

9.47.1.195 __FLT_MIN__

```
#define __FLT_MIN__ 1.17549435e-38F
```

9.47.1.196 __FLT_MIN_EXP__

```
#define __FLT_MIN_EXP__ (-125)
```

9.47.1.197 __FLT_NORM_MAX__

```
#define __FLT_NORM_MAX__ 3.40282347e+38F
```

9.47.1.198 __FLT_RADIX__

```
#define __FLT_RADIX__ 2
```

9.47.1.199 __FP_FAST_FMA

```
#define __FP_FAST_FMA 1
```

9.47.1.200 __FP_FAST_FMAF

```
#define __FP_FAST_FMAF 1
```

9.47.1.201 __FPCLASS_NEGINF

```
#define __FPCLASS_NEGINF 0x0004
```

9.47.1.202 __FPCLASS_NEGNORMAL

```
#define __FPCLASS_NEGNORMAL 0x0008
```

9.47.1.203 __FPCLASS_NEGSUBNORMAL

```
#define __FPCLASS_NEGSUBNORMAL 0x0010
```

9.47.1.204 __FPCLASS_NEGZERO

```
#define __FPCLASS_NEGZERO 0x0020
```

9.47.1.205 __FPCLASS_POSINF

```
#define __FPCLASS_POSINF 0x0200
```

9.47.1.206 __FPCLASS_POSNORMAL

```
#define __FPCLASS_POSNORMAL 0x0100
```

9.47.1.207 __FPCLASS_POSSUBNORMAL

```
#define __FPCLASS_POSSUBNORMAL 0x0080
```

9.47.1.208 __FPCLASS_POSZERO

```
#define __FPCLASS_POSZERO 0x0040
```

9.47.1.209 __FPCLASS_QNAN

```
#define __FPCLASS_QNAN 0x0002
```

9.47.1.210 __FPCLASS_SNAN

```
#define __FPCLASS_SNAN 0x0001
```

9.47.1.211 __GCC_ASM_FLAG_OUTPUTS__

```
#define __GCC_ASM_FLAG_OUTPUTS__ 1
```

9.47.1.212 __GCC_ATOMIC_BOOL_LOCK_FREE

```
#define __GCC_ATOMIC_BOOL_LOCK_FREE 2
```

9.47.1.213 __GCC_ATOMIC_CHAR16_T_LOCK_FREE

```
#define __GCC_ATOMIC_CHAR16_T_LOCK_FREE 2
```

9.47.1.214 __GCC_ATOMIC_CHAR32_T_LOCK_FREE

```
#define __GCC_ATOMIC_CHAR32_T_LOCK_FREE 2
```

9.47.1.215 __GCC_ATOMIC_CHAR_LOCK_FREE

```
#define __GCC_ATOMIC_CHAR_LOCK_FREE 2
```

9.47.1.216 __GCC_ATOMIC_INT_LOCK_FREE

```
#define __GCC_ATOMIC_INT_LOCK_FREE 2
```

9.47.1.217 __GCC_ATOMIC_LLONG_LOCK_FREE

```
#define __GCC_ATOMIC_LLONG_LOCK_FREE 2
```

9.47.1.218 __GCC_ATOMIC_LONG_LOCK_FREE

```
#define __GCC_ATOMIC_LONG_LOCK_FREE 2
```

9.47.1.219 __GCC_ATOMIC_POINTER_LOCK_FREE

```
#define __GCC_ATOMIC_POINTER_LOCK_FREE 2
```

9.47.1.220 __GCC_ATOMIC_SHORT_LOCK_FREE

```
#define __GCC_ATOMIC_SHORT_LOCK_FREE 2
```

9.47.1.221 __GCC_ATOMIC_TEST_AND_SET_TRUEVAL

```
#define __GCC_ATOMIC_TEST_AND_SET_TRUEVAL 1
```

9.47.1.222 __GCC_ATOMIC_WCHAR_T_LOCK_FREE

```
#define __GCC_ATOMIC_WCHAR_T_LOCK_FREE 2
```

9.47.1.223 __GCC_CONSTRUCTIVE_SIZE

```
#define __GCC_CONSTRUCTIVE_SIZE 64
```

9.47.1.224 __GCC_DESTRUCTIVE_SIZE

```
#define __GCC_DESTRUCTIVE_SIZE 64
```

9.47.1.225 __GCC_HAVE_DWARF2_CFI_ASM

```
#define __GCC_HAVE_DWARF2_CFI_ASM 1
```

9.47.1.226 __GCC_HAVE_SYNC_COMPARE_AND_SWAP_1

```
#define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_1 1
```

9.47.1.227 __GCC_HAVE_SYNC_COMPARE_AND_SWAP_16

```
#define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_16 1
```

9.47.1.228 __GCC_HAVE_SYNC_COMPARE_AND_SWAP_2

```
#define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_2 1
```

9.47.1.229 __GCC_HAVE_SYNC_COMPARE_AND_SWAP_4

```
#define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_4 1
```

9.47.1.230 __GCC_HAVE_SYNC_COMPARE_AND_SWAP_8

```
#define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_8 1
```

9.47.1.231 __GLIBCXX_BITSIZE_INT_N_0

```
#define __GLIBCXX_BITSIZE_INT_N_0 128
```

9.47.1.232 __GLIBCXX_TYPE_INT_N_0

```
#define __GLIBCXX_TYPE_INT_N_0 __int128
```

9.47.1.233 __GNUC__

```
#define __GNUC__ 4
```

9.47.1.234 __GNUC_GNU_INLINE__

```
#define __GNUC_GNU_INLINE__ 1
```

9.47.1.235 __GNUC_MINOR__

```
#define __GNUC_MINOR__ 2
```

9.47.1.236 __GNUC_PATCHLEVEL__

```
#define __GNUC_PATCHLEVEL__ 1
```

9.47.1.237 __GNUG__

```
#define __GNUG__ 4
```

9.47.1.238 __GXX_ABI_VERSION

```
#define __GXX_ABI_VERSION 1002
```

9.47.1.239 __GXX_EXPERIMENTAL_CXX0X__

```
#define __GXX_EXPERIMENTAL_CXX0X__ 1
```

9.47.1.240 __GXX_RTTI

```
#define __GXX_RTTI 1
```

9.47.1.241 __GXX_WEAK__

```
#define __GXX_WEAK__ 1
```

9.47.1.242 __HAVE_FUNCTION_MULTI_VERSIONING

```
#define __HAVE_FUNCTION_MULTI_VERSIONING 1
```

9.47.1.243 __INT16_C_SUFFIX__

```
#define __INT16_C_SUFFIX__
```

9.47.1.244 __INT16_FMTd__

```
#define __INT16_FMTd__ "hd"
```

9.47.1.245 __INT16_FMTi__

```
#define __INT16_FMTi__ "hi"
```

9.47.1.246 __INT16_MAX__

```
#define __INT16_MAX__ 32767
```

9.47.1.247 __INT16_TYPE__

```
#define __INT16_TYPE__ short
```

9.47.1.248 __INT32_C_SUFFIX__

```
#define __INT32_C_SUFFIX__
```

9.47.1.249 __INT32_FMTd__

```
#define __INT32_FMTd__ "d"
```

9.47.1.250 __INT32_FMTi__

```
#define __INT32_FMTi__ "i"
```

9.47.1.251 __INT32_MAX__

```
#define __INT32_MAX__ 2147483647
```

9.47.1.252 __INT32_TYPE__

```
#define __INT32_TYPE__ int
```

9.47.1.253 __INT64_C_SUFFIX__

```
#define __INT64_C_SUFFIX__ LL
```

9.47.1.254 __INT64_FMTd__

```
#define __INT64_FMTd__ "lld"
```

9.47.1.255 __INT64_FMTi__

```
#define __INT64_FMTi__ "lli"
```

9.47.1.256 __INT64_MAX__

```
#define __INT64_MAX__ 9223372036854775807LL
```

9.47.1.257 __INT64_TYPE__

```
#define __INT64_TYPE__ long long int
```

9.47.1.258 __INT8_C_SUFFIX__

```
#define __INT8_C_SUFFIX__
```

9.47.1.259 __INT8_FMTd__

```
#define __INT8_FMTd__ "hhd"
```

9.47.1.260 __INT8_FMTi__

```
#define __INT8_FMTi__ "hhi"
```

9.47.1.261 __INT8_MAX__

```
#define __INT8_MAX__ 127
```

9.47.1.262 __INT8_TYPE__

```
#define __INT8_TYPE__ signed char
```

9.47.1.263 __INT_FAST16_FMTd__

```
#define __INT_FAST16_FMTd__ "hd"
```

9.47.1.264 __INT_FAST16_FMTi__

```
#define __INT_FAST16_FMTi__ "hi"
```

9.47.1.265 __INT_FAST16_MAX__

```
#define __INT_FAST16_MAX__ 32767
```

9.47.1.266 __INT_FAST16_TYPE__

```
#define __INT_FAST16_TYPE__ short
```

9.47.1.267 __INT_FAST16_WIDTH__

```
#define __INT_FAST16_WIDTH__ 16
```

9.47.1.268 __INT_FAST32_FMTd__

```
#define __INT_FAST32_FMTd__ "d"
```

9.47.1.269 __INT_FAST32_FMTi__

```
#define __INT_FAST32_FMTi__ "i"
```

9.47.1.270 __INT_FAST32_MAX__

```
#define __INT_FAST32_MAX__ 2147483647
```

9.47.1.271 __INT_FAST32_TYPE__

```
#define __INT_FAST32_TYPE__ int
```

9.47.1.272 __INT_FAST32_WIDTH__

```
#define __INT_FAST32_WIDTH__ 32
```

9.47.1.273 __INT_FAST64_FMTd__

```
#define __INT_FAST64_FMTd__ "lld"
```

9.47.1.274 __INT_FAST64_FMTi__

```
#define __INT_FAST64_FMTi__ "lli"
```

9.47.1.275 __INT_FAST64_MAX__

```
#define __INT_FAST64_MAX__ 9223372036854775807LL
```

9.47.1.276 __INT_FAST64_TYPE__

```
#define __INT_FAST64_TYPE__ long long int
```

9.47.1.277 __INT_FAST64_WIDTH__

```
#define __INT_FAST64_WIDTH__ 64
```

9.47.1.278 __INT_FAST8_FMTd__

```
#define __INT_FAST8_FMTd__ "hhd"
```

9.47.1.279 __INT_FAST8_FMTi__

```
#define __INT_FAST8_FMTi__ "hhi"
```

9.47.1.280 __INT_FAST8_MAX__

```
#define __INT_FAST8_MAX__ 127
```

9.47.1.281 __INT_FAST8_TYPE__

```
#define __INT_FAST8_TYPE__ signed char
```

9.47.1.282 __INT_FAST8_WIDTH__

```
#define __INT_FAST8_WIDTH__ 8
```

9.47.1.283 __INT_LEAST16_FMTd__

```
#define __INT_LEAST16_FMTd__ "hd"
```

9.47.1.284 __INT_LEAST16_FMTi__

```
#define __INT_LEAST16_FMTi__ "hi"
```

9.47.1.285 __INT_LEAST16_MAX__

```
#define __INT_LEAST16_MAX__ 32767
```

9.47.1.286 __INT_LEAST16_TYPE__

```
#define __INT_LEAST16_TYPE__ short
```

9.47.1.287 __INT_LEAST16_WIDTH__

```
#define __INT_LEAST16_WIDTH__ 16
```

9.47.1.288 __INT_LEAST32_FMTd__

```
#define __INT_LEAST32_FMTd__ "d"
```

9.47.1.289 __INT_LEAST32_FMTi__

```
#define __INT_LEAST32_FMTi__ "i"
```

9.47.1.290 __INT_LEAST32_MAX__

```
#define __INT_LEAST32_MAX__ 2147483647
```

9.47.1.291 __INT_LEAST32_TYPE__

```
#define __INT_LEAST32_TYPE__ int
```

9.47.1.292 __INT_LEAST32_WIDTH__

```
#define __INT_LEAST32_WIDTH__ 32
```

9.47.1.293 __INT_LEAST64_FMTd__

```
#define __INT_LEAST64_FMTd__ "lld"
```

9.47.1.294 __INT_LEAST64_FMTi__

```
#define __INT_LEAST64_FMTi__ "lli"
```

9.47.1.295 __INT_LEAST64_MAX__

```
#define __INT_LEAST64_MAX__ 9223372036854775807LL
```

9.47.1.296 __INT_LEAST64_TYPE__

```
#define __INT_LEAST64_TYPE__ long long int
```

9.47.1.297 __INT_LEAST64_WIDTH__

```
#define __INT_LEAST64_WIDTH__ 64
```

9.47.1.298 __INT_LEAST8_FMTd__

```
#define __INT_LEAST8_FMTd__ "hhd"
```

9.47.1.299 __INT_LEAST8_FMTi__

```
#define __INT_LEAST8_FMTi__ "hhi"
```

9.47.1.300 __INT_LEAST8_MAX__

```
#define __INT_LEAST8_MAX__ 127
```

9.47.1.301 __INT_LEAST8_TYPE__

```
#define __INT_LEAST8_TYPE__ signed char
```

9.47.1.302 __INT_LEAST8_WIDTH__

```
#define __INT_LEAST8_WIDTH__ 8
```

9.47.1.303 __INT_MAX__

```
#define __INT_MAX__ 2147483647
```

9.47.1.304 __INT_WIDTH__

```
#define __INT_WIDTH__ 32
```

9.47.1.305 __INTMAX_C_SUFFIX__

```
#define __INTMAX_C_SUFFIX__ L
```

9.47.1.306 __INTMAX_FMTd__

```
#define __INTMAX_FMTd__ "ld"
```

9.47.1.307 __INTMAX_FMTi__

```
#define __INTMAX_FMTi__ "li"
```

9.47.1.308 __INTMAX_MAX__

```
#define __INTMAX_MAX__ 9223372036854775807L
```

9.47.1.309 __INTMAX_TYPE__

```
#define __INTMAX_TYPE__ long int
```

9.47.1.310 __INTMAX_WIDTH__

```
#define __INTMAX_WIDTH__ 64
```

9.47.1.311 __INTPTR_FMTd__

```
#define __INTPTR_FMTd__ "ld"
```

9.47.1.312 __INTPTR_FMTi__

```
#define __INTPTR_FMTi__ "li"
```

9.47.1.313 __INTPTR_MAX__

```
#define __INTPTR_MAX__ 9223372036854775807L
```

9.47.1.314 __INTPTR_TYPE__

```
#define __INTPTR_TYPE__ long int
```

9.47.1.315 __INTPTR_WIDTH__

```
#define __INTPTR_WIDTH__ 64
```

9.47.1.316 __LDBL_DECIMAL_DIG__

```
#define __LDBL_DECIMAL_DIG__ 17
```

9.47.1.317 __LDBL_DENORM_MIN__

```
#define __LDBL_DENORM_MIN__ 4.9406564584124654e-324L
```

9.47.1.318 __LDBL_DIG__

```
#define __LDBL_DIG__ 15
```

9.47.1.319 __LDBL_EPSILON__

```
#define __LDBL_EPSILON__ 2.220446049250313le-16L
```

9.47.1.320 __LDBL_HAS_DENORM__

```
#define __LDBL_HAS_DENORM__ 1
```

9.47.1.321 __LDBL_HAS_INFINITY__

```
#define __LDBL_HAS_INFINITY__ 1
```

9.47.1.322 __LDBL_HAS_QUIET_NAN__

```
#define __LDBL_HAS_QUIET_NAN__ 1
```

9.47.1.323 __LDBL_MANT_DIG__

```
#define __LDBL_MANT_DIG__ 53
```

9.47.1.324 __LDBL_MAX_10_EXP__

```
#define __LDBL_MAX_10_EXP__ 308
```

9.47.1.325 __LDBL_MAX__

```
#define __LDBL_MAX__ 1.7976931348623157e+308L
```

9.47.1.326 __LDBL_MAX_EXP__

```
#define __LDBL_MAX_EXP__ 1024
```

9.47.1.327 __LDBL_MIN_10_EXP__

```
#define __LDBL_MIN_10_EXP__ (-307)
```

9.47.1.328 __LDBL_MIN__

```
#define __LDBL_MIN__ 2.2250738585072014e-308L
```

9.47.1.329 __LDBL_MIN_EXP__

```
#define __LDBL_MIN_EXP__ (-1021)
```

9.47.1.330 __LDBL_NORM_MAX__

```
#define __LDBL_NORM_MAX__ 1.7976931348623157e+308L
```

9.47.1.331 __LITTLE_ENDIAN__

```
#define __LITTLE_ENDIAN__ 1
```

9.47.1.332 __LLONG_WIDTH__

```
#define __LLONG_WIDTH__ 64
```

9.47.1.333 __llvm__

```
#define __llvm__ 1
```

9.47.1.334 __LONG_LONG_MAX__

```
#define __LONG_LONG_MAX__ 9223372036854775807LL
```

9.47.1.335 __LONG_MAX__

```
#define __LONG_MAX__ 9223372036854775807L
```

9.47.1.336 __LONG_WIDTH__

```
#define __LONG_WIDTH__ 64
```

9.47.1.337 __LP64__

```
#define __LP64__ 1
```

9.47.1.338 __MACH__

```
#define __MACH__ 1
```

9.47.1.339 __MEMORY_SCOPE_DEVICE

```
#define __MEMORY_SCOPE_DEVICE 1
```

9.47.1.340 __MEMORY_SCOPE_SINGLE

```
#define __MEMORY_SCOPE_SINGLE 4
```

9.47.1.341 __MEMORY_SCOPE_SYSTEM

```
#define __MEMORY_SCOPE_SYSTEM 0
```

9.47.1.342 __MEMORY_SCOPE_WRKGRP

```
#define __MEMORY_SCOPE_WRKGRP 2
```

9.47.1.343 __MEMORY_SCOPE_WVFRNT

```
#define __MEMORY_SCOPE_WVFRNT 3
```

9.47.1.344 __NO_INLINE__

```
#define __NO_INLINE__ 1
```

9.47.1.345 __NO_MATH_ERRNO__

```
#define __NO_MATH_ERRNO__ 1
```

9.47.1.346 __nonnull

```
#define __nonnull _Nonnull
```

9.47.1.347 __null_unspecified

```
#define __null_unspecified _Null_unspecified
```

9.47.1.348 __nullable

```
#define __nullable _Nullable
```

9.47.1.349 __OBJC_BOOL_IS_BOOL

```
#define __OBJC_BOOL_IS_BOOL 1
```

9.47.1.350 __OPENCL_MEMORY_SCOPE_ALL_SVM_DEVICES

```
#define __OPENCL_MEMORY_SCOPE_ALL_SVM_DEVICES 3
```

9.47.1.351 __OPENCL_MEMORY_SCOPE_DEVICE

```
#define __OPENCL_MEMORY_SCOPE_DEVICE 2
```

9.47.1.352 __OPENCL_MEMORY_SCOPE_SUB_GROUP

```
#define __OPENCL_MEMORY_SCOPE_SUB_GROUP 4
```

9.47.1.353 __OPENCL_MEMORY_SCOPE_WORK_GROUP

```
#define __OPENCL_MEMORY_SCOPE_WORK_GROUP 1
```

9.47.1.354 __OPENCL_MEMORY_SCOPE_WORK_ITEM

```
#define __OPENCL_MEMORY_SCOPE_WORK_ITEM 0
```

9.47.1.355 __ORDER_BIG_ENDIAN__

```
#define __ORDER_BIG_ENDIAN__ 4321
```

9.47.1.356 __ORDER_LITTLE_ENDIAN__

```
#define __ORDER_LITTLE_ENDIAN__ 1234
```

9.47.1.357 __ORDER_PDP_ENDIAN__

```
#define __ORDER_PDP_ENDIAN__ 3412
```

9.47.1.358 __PIC__

```
#define __PIC__ 2
```

9.47.1.359 __pic__

```
#define __pic__ 2
```

9.47.1.360 __POINTER_WIDTH__

```
#define __POINTER_WIDTH__ 64
```

9.47.1.361 __PRAGMA_REDEFINE_EXTNAME

```
#define __PRAGMA_REDEFINE_EXTNAME 1
```

9.47.1.362 __private_extern__

```
#define __private_extern__ extern
```

9.47.1.363 __PTRDIFF_FMTd__

```
#define __PTRDIFF_FMTd__ "ld"
```

9.47.1.364 __PTRDIFF_FMTi__

```
#define __PTRDIFF_FMTi__ "li"
```

9.47.1.365 __PTRDIFF_MAX__

```
#define __PTRDIFF_MAX__ 9223372036854775807L
```

9.47.1.366 __PTRDIFF_TYPE__

```
#define __PTRDIFF_TYPE__ long int
```

9.47.1.367 __PTRDIFF_WIDTH__

```
#define __PTRDIFF_WIDTH__ 64
```

9.47.1.368 __REGISTER_PREFIX__

```
#define __REGISTER_PREFIX__
```

9.47.1.369 __SCHAR_MAX__

```
#define __SCHAR_MAX__ 127
```

9.47.1.370 __SHRT_MAX__

```
#define __SHRT_MAX__ 32767
```

9.47.1.371 __SHRT_WIDTH__

```
#define __SHRT_WIDTH__ 16
```

9.47.1.372 __SIG_ATOMIC_MAX__

```
#define __SIG_ATOMIC_MAX__ 2147483647
```

9.47.1.373 __SIG_ATOMIC_WIDTH__

```
#define __SIG_ATOMIC_WIDTH__ 32
```

9.47.1.374 __SIZE_FMTo__

```
#define __SIZE_FMTo__ "lo"
```

9.47.1.375 __SIZE_FMTu__

```
#define __SIZE_FMTu__ "lu"
```

9.47.1.376 __SIZE_FMTX__

```
#define __SIZE_FMTX__ "lX"
```

9.47.1.377 __SIZE_FMTx__

```
#define __SIZE_FMTx__ "lx"
```

9.47.1.378 __SIZE_MAX__

```
#define __SIZE_MAX__ 18446744073709551615UL
```

9.47.1.379 __SIZE_TYPE__

```
#define __SIZE_TYPE__ long unsigned int
```

9.47.1.380 __SIZE_WIDTH__

```
#define __SIZE_WIDTH__ 64
```

9.47.1.381 __SIZEOF_DOUBLE__

```
#define __SIZEOF_DOUBLE__ 8
```

9.47.1.382 __SIZEOF_FLOAT__

```
#define __SIZEOF_FLOAT__ 4
```

9.47.1.383 __SIZEOF_INT128__

```
#define __SIZEOF_INT128__ 16
```

9.47.1.384 __SIZEOF_INT__

```
#define __SIZEOF_INT__ 4
```

9.47.1.385 __SIZEOF_LONG__

```
#define __SIZEOF_LONG__ 8
```

9.47.1.386 __SIZEOF_LONG_DOUBLE__

```
#define __SIZEOF_LONG_DOUBLE__ 8
```

9.47.1.387 __SIZEOF_LONG_LONG__

```
#define __SIZEOF_LONG_LONG__ 8
```

9.47.1.388 __SIZEOF_POINTER__

```
#define __SIZEOF_POINTER__ 8
```

9.47.1.389 __SIZEOF_PTRDIFF_T__

```
#define __SIZEOF_PTRDIFF_T__ 8
```

9.47.1.390 __SIZEOF_SHORT__

```
#define __SIZEOF_SHORT__ 2
```

9.47.1.391 __SIZEOF_SIZE_T__

```
#define __SIZEOF_SIZE_T__ 8
```

9.47.1.392 __SIZEOF_WCHAR_T__

```
#define __SIZEOF_WCHAR_T__ 4
```

9.47.1.393 __SIZEOF_WINT_T__

```
#define __SIZEOF_WINT_T__ 4
```

9.47.1.394 __SSP__

```
#define __SSP__ 1
```

9.47.1.395 __STDC__

```
#define __STDC__ 1
```

9.47.1.396 __STDC_EMBED_EMPTY__

```
#define __STDC_EMBED_EMPTY__ 2
```

9.47.1.397 __STDC_EMBED_FOUND__

```
#define __STDC_EMBED_FOUND__ 1
```

9.47.1.398 __STDC_EMBED_NOT_FOUND__

```
#define __STDC_EMBED_NOT_FOUND__ 0
```

9.47.1.399 __STDC_HOSTED__

```
#define __STDC_HOSTED__ 1
```

9.47.1.400 __STDC_NO_THREADS__

```
#define __STDC_NO_THREADS__ 1
```

9.47.1.401 __STDC_UTF_16__

```
#define __STDC_UTF_16__ 1
```

9.47.1.402 __STDC_UTF_32__

```
#define __STDC_UTF_32__ 1
```

9.47.1.403 __STDCPP_DEFAULT_NEW_ALIGNMENT__

```
#define __STDCPP_DEFAULT_NEW_ALIGNMENT__ 16UL
```

9.47.1.404 __STDCPP_THREADS__

```
#define __STDCPP_THREADS__ 1
```

9.47.1.405 __strong

```
#define __strong
```

9.47.1.406 __UINT16_C_SUFFIX__

```
#define __UINT16_C_SUFFIX__
```

9.47.1.407 __UINT16_FMTo__

```
#define __UINT16_FMTo__ "ho"
```

9.47.1.408 __UINT16_FMTu__

```
#define __UINT16_FMTu__ "hu"
```

9.47.1.409 __UINT16_FMTX__

```
#define __UINT16_FMTX__ "hX"
```

9.47.1.410 __UINT16_FMTx__

```
#define __UINT16_FMTx__ "hx"
```

9.47.1.411 __UINT16_MAX__

```
#define __UINT16_MAX__ 65535
```

9.47.1.412 __UINT16_TYPE__

```
#define __UINT16_TYPE__ unsigned short
```

9.47.1.413 __UINT32_C_SUFFIX__

```
#define __UINT32_C_SUFFIX__ U
```

9.47.1.414 __UINT32_FMTo__

```
#define __UINT32_FMTo__ "o"
```

9.47.1.415 __UINT32_FMTu__

```
#define __UINT32_FMTu__ "u"
```

9.47.1.416 __UINT32_FMTX__

```
#define __UINT32_FMTX__ "X"
```

9.47.1.417 __UINT32_FMTx__

```
#define __UINT32_FMTx__ "x"
```

9.47.1.418 __UINT32_MAX__

```
#define __UINT32_MAX__ 4294967295U
```

9.47.1.419 __UINT32_TYPE__

```
#define __UINT32_TYPE__ unsigned int
```

9.47.1.420 __UINT64_C_SUFFIX__

```
#define __UINT64_C_SUFFIX__ ULL
```

9.47.1.421 __UINT64_FMTo__

```
#define __UINT64_FMTo__ "llo"
```

9.47.1.422 __UINT64_FMTu__

```
#define __UINT64_FMTu__ "llu"
```

9.47.1.423 __UINT64_FMTX__

```
#define __UINT64_FMTX__ "llX"
```

9.47.1.424 __UINT64_FMTx__

```
#define __UINT64_FMTx__ "llx"
```

9.47.1.425 __UINT64_MAX__

```
#define __UINT64_MAX__ 18446744073709551615ULL
```

9.47.1.426 __UINT64_TYPE__

```
#define __UINT64_TYPE__ long long unsigned int
```

9.47.1.427 __UINT8_C_SUFFIX__

```
#define __UINT8_C_SUFFIX__
```

9.47.1.428 __UINT8_FMTo__

```
#define __UINT8_FMTo__ "hho"
```

9.47.1.429 __UINT8_FMTu__

```
#define __UINT8_FMTu__ "hhu"
```

9.47.1.430 __UINT8_FMTX__

```
#define __UINT8_FMTX__ "hhX"
```

9.47.1.431 __UINT8_FMTx__

```
#define __UINT8_FMTx__ "hhx"
```

9.47.1.432 __UINT8_MAX__

```
#define __UINT8_MAX__ 255
```

9.47.1.433 __UINT8_TYPE__

```
#define __UINT8_TYPE__ unsigned char
```

9.47.1.434 __UINT_FAST16_FMTo__

```
#define __UINT_FAST16_FMTo__ "ho"
```

9.47.1.435 __UINT_FAST16_FMTu__

```
#define __UINT_FAST16_FMTu__ "hu"
```

9.47.1.436 __UINT_FAST16_FMTX__

```
#define __UINT_FAST16_FMTX__ "hX"
```

9.47.1.437 __UINT_FAST16_FMTx__

```
#define __UINT_FAST16_FMTx__ "hx"
```

9.47.1.438 __UINT_FAST16_MAX__

```
#define __UINT_FAST16_MAX__ 65535
```

9.47.1.439 __UINT_FAST16_TYPE__

```
#define __UINT_FAST16_TYPE__ unsigned short
```

9.47.1.440 __UINT_FAST32_FMTo__

```
#define __UINT_FAST32_FMTo__ "o"
```

9.47.1.441 __UINT_FAST32_FMTu__

```
#define __UINT_FAST32_FMTu__ "u"
```

9.47.1.442 __UINT_FAST32_FMTX__

```
#define __UINT_FAST32_FMTX__ "X"
```

9.47.1.443 __UINT_FAST32_FMTx__

```
#define __UINT_FAST32_FMTx__ "x"
```

9.47.1.444 __UINT_FAST32_MAX__

```
#define __UINT_FAST32_MAX__ 4294967295U
```

9.47.1.445 __UINT_FAST32_TYPE__

```
#define __UINT_FAST32_TYPE__ unsigned int
```

9.47.1.446 __UINT_FAST64_FMTo__

```
#define __UINT_FAST64_FMTo__ "llo"
```

9.47.1.447 __UINT_FAST64_FMTu__

```
#define __UINT_FAST64_FMTu__ "llu"
```

9.47.1.448 __UINT_FAST64_FMTX__

```
#define __UINT_FAST64_FMTX__ "llX"
```

9.47.1.449 __UINT_FAST64_FMTx__

```
#define __UINT_FAST64_FMTx__ "llx"
```

9.47.1.450 __UINT_FAST64_MAX__

```
#define __UINT_FAST64_MAX__ 18446744073709551615ULL
```

9.47.1.451 __UINT_FAST64_TYPE__

```
#define __UINT_FAST64_TYPE__ long long unsigned int
```

9.47.1.452 __UINT_FAST8_FMTo__

```
#define __UINT_FAST8_FMTo__ "hho"
```

9.47.1.453 __UINT_FAST8_FMTu__

```
#define __UINT_FAST8_FMTu__ "hhu"
```

9.47.1.454 __UINT_FAST8_FMTX__

```
#define __UINT_FAST8_FMTX__ "hhX"
```

9.47.1.455 __UINT_FAST8_FMTx__

```
#define __UINT_FAST8_FMTx__ "hhx"
```

9.47.1.456 __UINT_FAST8_MAX__

```
#define __UINT_FAST8_MAX__ 255
```

9.47.1.457 __UINT_FAST8_TYPE__

```
#define __UINT_FAST8_TYPE__ unsigned char
```

9.47.1.458 __UINT_LEAST16_FMTo__

```
#define __UINT_LEAST16_FMTo__ "ho"
```

9.47.1.459 __UINT_LEAST16_FMTu__

```
#define __UINT_LEAST16_FMTu__ "hu"
```

9.47.1.460 __UINT_LEAST16_FMTX__

```
#define __UINT_LEAST16_FMTX__ "hX"
```

9.47.1.461 __UINT_LEAST16_FMTx__

```
#define __UINT_LEAST16_FMTx__ "hx"
```

9.47.1.462 __UINT_LEAST16_MAX__

```
#define __UINT_LEAST16_MAX__ 65535
```

9.47.1.463 __UINT_LEAST16_TYPE__

```
#define __UINT_LEAST16_TYPE__ unsigned short
```

9.47.1.464 __UINT_LEAST32_FMTo__

```
#define __UINT_LEAST32_FMTo__ "o"
```

9.47.1.465 __UINT_LEAST32_FMTu__

```
#define __UINT_LEAST32_FMTu__ "u"
```

9.47.1.466 __UINT_LEAST32_FMTX__

```
#define __UINT_LEAST32_FMTX__ "X"
```

9.47.1.467 __UINT_LEAST32_FMTx__

```
#define __UINT_LEAST32_FMTx__ "x"
```

9.47.1.468 __UINT_LEAST32_MAX__

```
#define __UINT_LEAST32_MAX__ 4294967295U
```

9.47.1.469 __UINT_LEAST32_TYPE__

```
#define __UINT_LEAST32_TYPE__ unsigned int
```

9.47.1.470 __UINT_LEAST64_FMTo__

```
#define __UINT_LEAST64_FMTo__ "llo"
```

9.47.1.471 __UINT_LEAST64_FMTu__

```
#define __UINT_LEAST64_FMTu__ "llu"
```

9.47.1.472 __UINT_LEAST64_FMTX__

```
#define __UINT_LEAST64_FMTX__ "llX"
```

9.47.1.473 __UINT_LEAST64_FMTx__

```
#define __UINT_LEAST64_FMTx__ "llx"
```

9.47.1.474 __UINT_LEAST64_MAX__

```
#define __UINT_LEAST64_MAX__ 18446744073709551615ULL
```

9.47.1.475 __UINT_LEAST64_TYPE__

```
#define __UINT_LEAST64_TYPE__ long long unsigned int
```

9.47.1.476 __UINT_LEAST8_FMTo__

```
#define __UINT_LEAST8_FMTo__ "hho"
```

9.47.1.477 __UINT_LEAST8_FMTu__

```
#define __UINT_LEAST8_FMTu__ "hhu"
```

9.47.1.478 __UINT_LEAST8_FMTX__

```
#define __UINT_LEAST8_FMTX__ "hhX"
```

9.47.1.479 __UINT_LEAST8_FMTx__

```
#define __UINT_LEAST8_FMTx__ "hhx"
```

9.47.1.480 __UINT_LEAST8_MAX__

```
#define __UINT_LEAST8_MAX__ 255
```

9.47.1.481 __UINT_LEAST8_TYPE__

```
#define __UINT_LEAST8_TYPE__ unsigned char
```

9.47.1.482 __UINTMAX_C_SUFFIX__

```
#define __UINTMAX_C_SUFFIX__ UL
```

9.47.1.483 __UINTMAX_FMTo__

```
#define __UINTMAX_FMTo__ "lo"
```

9.47.1.484 __UINTMAX_FMTu__

```
#define __UINTMAX_FMTu__ "lu"
```

9.47.1.485 __UINTMAX_FMTX__

```
#define __UINTMAX_FMTX__ "lX"
```

9.47.1.486 __UINTMAX_FMTx__

```
#define __UINTMAX_FMTx__ "lx"
```

9.47.1.487 __UINTMAX_MAX__

```
#define __UINTMAX_MAX__ 18446744073709551615UL
```

9.47.1.488 __UINTMAX_TYPE__

```
#define __UINTMAX_TYPE__ long unsigned int
```

9.47.1.489 __UINTMAX_WIDTH__

```
#define __UINTMAX_WIDTH__ 64
```

9.47.1.490 __UINTPTR_FMTo__

```
#define __UINTPTR_FMTo__ "lo"
```

9.47.1.491 __UINTPTR_FMTu__

```
#define __UINTPTR_FMTu__ "lu"
```

9.47.1.492 __UINTPTR_FMTX__

```
#define __UINTPTR_FMTX__ "lX"
```

9.47.1.493 __UINTPTR_FMTx__

```
#define __UINTPTR_FMTx__ "lx"
```

9.47.1.494 __UINTPTR_MAX__

```
#define __UINTPTR_MAX__ 18446744073709551615UL
```

9.47.1.495 __UINTPTR_TYPE__

```
#define __UINTPTR_TYPE__ long unsigned int
```

9.47.1.496 __UINTPTR_WIDTH__

```
#define __UINTPTR_WIDTH__ 64
```

9.47.1.497 __unsafe_unretained

```
#define __unsafe_unretained
```

9.47.1.498 __USER_LABEL_PREFIX__

```
#define __USER_LABEL_PREFIX__ _
```

9.47.1.499 __VERSION__

```
#define __VERSION__ "Apple LLVM 17.0.0 (clang-1700.0.13.3)"
```

9.47.1.500 __WCHAR_MAX__

```
#define __WCHAR_MAX__ 2147483647
```

9.47.1.501 __WCHAR_TYPE__

```
#define __WCHAR_TYPE__ int
```

9.47.1.502 __WCHAR_WIDTH__

```
#define __WCHAR_WIDTH__ 32
```

9.47.1.503 __weak

```
#define __weak __attribute__((objc_gc(weak)))
```

9.47.1.504 __WINT_MAX__

```
#define __WINT_MAX__ 2147483647
```

9.47.1.505 __WINT_TYPE__

```
#define __WINT_TYPE__ int
```

9.47.1.506 __WINT_WIDTH__

```
#define __WINT_WIDTH__ 32
```

9.47.1.507 _LP64

```
#define _LP64 1
```

9.47.1.508 QT_CHARTS_LIB

```
#define QT_CHARTS_LIB 1
```

9.47.1.509 QT_CORE_LIB

```
#define QT_CORE_LIB 1
```

9.47.1.510 QT_GUI_LIB

```
#define QT_GUI_LIB 1
```

9.47.1.511 QT_NETWORK_LIB

```
#define QT_NETWORK_LIB 1
```

9.47.1.512 QT_NO_DEBUG

```
#define QT_NO_DEBUG 1
```

9.47.1.513 QT_OPENGL_LIB

```
#define QT_OPENGL_LIB 1
```

9.47.1.514 QT_OPENGLWIDGETS_LIB

```
#define QT_OPENGLWIDGETS_LIB 1
```

9.47.1.515 QT_WIDGETS_LIB

```
#define QT_WIDGETS_LIB 1
```

9.47.1.516 SIZEOF_DPTR

```
#define SIZEOF_DPTR (sizeof(void*))
```

9.47.1.517 TARGET_IPHONE_SIMULATOR

```
#define TARGET_IPHONE_SIMULATOR 0
```

9.47.1.518 TARGET_OS_ARROW

```
#define TARGET_OS_ARROW 1
```

9.47.1.519 TARGET_OS_BRIDGE

```
#define TARGET_OS_BRIDGE 0
```

9.47.1.520 TARGET_OS_DRIVERKIT

```
#define TARGET_OS_DRIVERKIT 0
```

9.47.1.521 TARGET_OS_EMBEDDED

```
#define TARGET_OS_EMBEDDED 0
```

9.47.1.522 TARGET_OS_IOS

```
#define TARGET_OS_IOS 0
```

9.47.1.523 TARGET_OS_IOSMAC

```
#define TARGET_OS_IOSMAC 0
```

9.47.1.524 TARGET_OS_IPHONE

```
#define TARGET_OS_IPHONE 0
```

9.47.1.525 TARGET_OS_LINUX

```
#define TARGET_OS_LINUX 0
```

9.47.1.526 TARGET_OS_MAC

```
#define TARGET_OS_MAC 1
```

9.47.1.527 TARGET_OS_MACCATALYST

```
#define TARGET_OS_MACCATALYST 0
```

9.47.1.528 TARGET_OS_NANO

```
#define TARGET_OS_NANO 0
```

9.47.1.529 TARGET_OS_OSX

```
#define TARGET_OS_OSX 1
```

9.47.1.530 TARGET_OS_SIMULATOR

```
#define TARGET_OS_SIMULATOR 0
```

9.47.1.531 TARGET_OS_TV

```
#define TARGET_OS_TV 0
```

9.47.1.532 TARGET_OS_UIKITFORMAC

```
#define TARGET_OS_UIKITFORMAC 0
```

9.47.1.533 TARGET_OS_UNIX

```
#define TARGET_OS_UNIX 0
```

9.47.1.534 TARGET_OS_VISION

```
#define TARGET_OS_VISION 0
```

9.47.1.535 TARGET_OS_WATCH

```
#define TARGET_OS_WATCH 0
```

9.47.1.536 TARGET_OS_WIN32

```
#define TARGET_OS_WIN32 0
```

9.47.1.537 TARGET_OS_WINDOWS

```
#define TARGET_OS_WINDOWS 0
```

9.47.1.538 TARGET_OS_XR

```
#define TARGET_OS_XR 0
```

9.48 moc_predefs.h

浏览该文件的文档.

```
00001 #define QT_CHARTS_LIB 1
00002 #define QT_CORE_LIB 1
00003 #define QT_GUI_LIB 1
00004 #define QT_NETWORK_LIB 1
00005 #define QT_NO_DEBUG 1
00006 #define QT_OPENGLWIDGETS_LIB 1
00007 #define QT_OPENGL_LIB 1
00008 #define QT_WIDGETS_LIB 1
00009 #define SIZEOF_DPTR (sizeof(void*))
00010 #define TARGET_IPHONE_SIMULATOR 0
00011 #define TARGET_OS_ARROW 1
00012 #define TARGET_OS_BRIDGE 0
00013 #define TARGET_OS_DRIVERKIT 0
00014 #define TARGET_OS_EMBEDDED 0
00015 #define TARGET_OS_IOS 0
00016 #define TARGET_OS_IOSMAC 0
00017 #define TARGET_OS_IPHONE 0
00018 #define TARGET_OS_LINUX 0
00019 #define TARGET_OS_MAC 1
00020 #define TARGET_OS_MACCATALYST 0
00021 #define TARGET_OS_NANO 0
00022 #define TARGET_OS_OSX 1
00023 #define TARGET_OS_SIMULATOR 0
00024 #define TARGET_OS_TV 0
00025 #define TARGET_OS_UIKITFORMAC 0
00026 #define TARGET_OS_UNIX 0
00027 #define TARGET_OS_VISION 0
00028 #define TARGET_OS_WATCH 0
00029 #define TARGET_OS_WIN32 0
```

```
00030 #define TARGET_OS_WINDOWS 0
00031 #define TARGET_OS_XR 0
00032 #define _LP64 1
00033 #define __AARCH64EL__ 1
00034 #define __AARCH64_CMODEL_SMALL__ 1
00035 #define __AARCH64_SIMD__ 1
00036 #define __APPLE_CC__ 6000
00037 #define __APPLE__ 1
00038 #define __ARM64_ARCH_8__ 1
00039 #define __ARM_64BIT_STATE 1
00040 #define __ARM_ACLE 200
00041 #define __ARM_ALIGN_MAX_STACK_PWR 4
00042 #define __ARM_ARCH 8
00043 #define __ARM_ARCH_8_3__ 1
00044 #define __ARM_ARCH_8_4__ 1
00045 #define __ARM_ARCH_8_5__ 1
00046 #define __ARM_ARCH_ISA_A64 1
00047 #define __ARM_ARCH_PROFILE 'A'
00048 #define __ARM_FEATURE_AES 1
00049 #define __ARM_FEATURE_ATOMICS 1
00050 #define __ARM_FEATURE_BTI 1
00051 #define __ARM_FEATURE_CLZ 1
00052 #define __ARM_FEATURE_COMPLEX 1
00053 #define __ARM_FEATURE_CRC32 1
00054 #define __ARM_FEATURE_CRYPTO 1
00055 #define __ARM_FEATURE_DIRECTED_ROUNDING 1
00056 #define __ARM_FEATURE_DIV 1
00057 #define __ARM_FEATURE_DOTPROD 1
00058 #define __ARM_FEATURE_FMA 1
00059 #define __ARM_FEATURE_FP16_FML 1
00060 #define __ARM_FEATURE_FP16_SCALAR_ARITHMETIC 1
00061 #define __ARM_FEATURE_FP16_VECTOR_ARITHMETIC 1
00062 #define __ARM_FEATURE_FRINT 1
00063 #define __ARM_FEATURE_IDIV 1
00064 #define __ARM_FEATURE_JCVT 1
00065 #define __ARM_FEATURE_LDREX 0xF
00066 #define __ARM_FEATURE_NUMERIC_MAXMIN 1
00067 #define __ARM_FEATURE_PAUTH 1
00068 #define __ARM_FEATURE_QRDMX 1
00069 #define __ARM_FEATURE_RCPC 1
00070 #define __ARM_FEATURE_SHA2 1
00071 #define __ARM_FEATURE_SHA3 1
00072 #define __ARM_FEATURE_SHA512 1
00073 #define __ARM_FEATURE_UNALIGNED 1
00074 #define __ARM_FP 0xE
00075 #define __ARM_FP16_ARGS 1
00076 #define __ARM_FP16_FORMAT_IEEE 1
00077 #define __ARM_NEON 1
00078 #define __ARM_NEON_FP 0xE
00079 #define __ARM_NEON__ 1
00080 #define __ARM_PCS_AAPCS64 1
00081 #define __ARM_SIZEOF_MINIMAL_ENUM 4
00082 #define __ARM_SIZEOF_WCHAR_T 4
00083 #define __ARM_STATE_ZA 1
00084 #define __ARM_STATE_ZT0 1
00085 #define __ATOMIC_ACQUIRE 2
00086 #define __ATOMIC_ACQ_REL 4
00087 #define __ATOMIC_CONSUME 1
00088 #define __ATOMIC_RELAXED 0
00089 #define __ATOMIC_RELEASE 3
00090 #define __ATOMIC_SEQ_CST 5
00091 #define __BIGGEST_ALIGNMENT__ 8
00092 #define __BITINT_MAXWIDTH__ 128
00093 #define __BLOCKS__ 1
00094 #define __BOOL_WIDTH__ 8
00095 #define __BYTE_ORDER__ __ORDER_LITTLE_ENDIAN__
00096 #define __CHAR16_TYPE__ unsigned short
00097 #define __CHAR32_TYPE__ unsigned int
00098 #define __CHAR_BIT__ 8
00099 #define __CLANG_ATOMIC_BOOL_LOCK_FREE 2
00100 #define __CLANG_ATOMIC_CHAR16_T_LOCK_FREE 2
00101 #define __CLANG_ATOMIC_CHAR32_T_LOCK_FREE 2
00102 #define __CLANG_ATOMIC_CHAR_LOCK_FREE 2
00103 #define __CLANG_ATOMIC_INT_LOCK_FREE 2
00104 #define __CLANG_ATOMIC_LLONG_LOCK_FREE 2
00105 #define __CLANG_ATOMIC_LONG_LOCK_FREE 2
00106 #define __CLANG_ATOMIC_POINTER_LOCK_FREE 2
00107 #define __CLANG_ATOMIC_SHORT_LOCK_FREE 2
00108 #define __CLANG_ATOMIC_WCHAR_T_LOCK_FREE 2
00109 #define __CONSTANT_CFSTRINGS__ 1
00110 #define __DBL_DECIMAL_DIG__ 17
00111 #define __DBL_DENORM_MIN__ 4.9406564584124654e-324
00112 #define __DBL_DIG__ 15
00113 #define __DBL_EPSILON__ 2.2204460492503131e-16
00114 #define __DBL_HAS_DENORM__ 1
00115 #define __DBL_HAS_INFINITY__ 1
00116 #define __DBL_HAS_QUIET_NAN__ 1
```

```
00117 #define __DBL_MANT_DIG__ 53
00118 #define __DBL_MAX_10_EXP__ 308
00119 #define __DBL_MAX_EXP__ 1024
00120 #define __DBL_MAX__ 1.7976931348623157e+308
00121 #define __DBL_MIN_10_EXP__ (-307)
00122 #define __DBL_MIN_EXP__ (-1021)
00123 #define __DBL_MIN__ 2.2250738585072014e-308
00124 #define __DBL_NORM_MAX__ 1.7976931348623157e+308
00125 #define __DECIMAL_DIG__ __LDBL_DECIMAL_DIG__
00126 #define __DEPRECATED 1
00127 #define __DYNAMIC__ 1
00128 #define __ENVIRONMENT_MAC_OS_X_VERSION_MIN_REQUIRED__ 150000
00129 #define __ENVIRONMENT_OS_VERSION_MIN_REQUIRED__ 150000
00130 #define __EXCEPTIONS 1
00131 #define __FINITE_MATH_ONLY__ 0
00132 #define __FLT16_DECIMAL_DIG__ 5
00133 #define __FLT16_DENORM_MIN__ 5.9604644775390625e-8F16
00134 #define __FLT16_DIG__ 3
00135 #define __FLT16_EPSILON__ 9.765625e-4F16
00136 #define __FLT16_HAS_DENORM__ 1
00137 #define __FLT16_HAS_INFINITY__ 1
00138 #define __FLT16_HAS_QUIET_NAN__ 1
00139 #define __FLT16_MANT_DIG__ 11
00140 #define __FLT16_MAX_10_EXP__ 4
00141 #define __FLT16_MAX_EXP__ 16
00142 #define __FLT16_MAX__ 6.5504e+4F16
00143 #define __FLT16_MIN_10_EXP__ (-4)
00144 #define __FLT16_MIN_EXP__ (-13)
00145 #define __FLT16_MIN__ 6.103515625e-5F16
00146 #define __FLT16_NORM_MAX__ 6.5504e+4F16
00147 #define __FLT_DECIMAL_DIG__ 9
00148 #define __FLT_DENORM_MIN__ 1.40129846e-45F
00149 #define __FLT_DIG__ 6
00150 #define __FLT_EPSILON__ 1.19209290e-7F
00151 #define __FLT_HAS_DENORM__ 1
00152 #define __FLT_HAS_INFINITY__ 1
00153 #define __FLT_HAS_QUIET_NAN__ 1
00154 #define __FLT_MANT_DIG__ 24
00155 #define __FLT_MAX_10_EXP__ 38
00156 #define __FLT_MAX_EXP__ 128
00157 #define __FLT_MAX__ 3.40282347e+38F
00158 #define __FLT_MIN_10_EXP__ (-37)
00159 #define __FLT_MIN_EXP__ (-125)
00160 #define __FLT_MIN__ 1.17549435e-38F
00161 #define __FLT_NORM_MAX__ 3.40282347e+38F
00162 #define __FLT_RADIX__ 2
00163 #define __FPCLASS_NEGINF 0x0004
00164 #define __FPCLASS_NEGNORMAL 0x0008
00165 #define __FPCLASS_NEGSUBNORMAL 0x0010
00166 #define __FPCLASS_NEGZERO 0x0020
00167 #define __FPCLASS_POSINF 0x0200
00168 #define __FPCLASS_POSNORMAL 0x0100
00169 #define __FPCLASS_POSSUBNORMAL 0x0080
00170 #define __FPCLASS_POSZERO 0x0040
00171 #define __FPCLASS_QNAN 0x0002
00172 #define __FPCLASS_SNAN 0x0001
00173 #define __FP_FAST_FMA 1
00174 #define __FP_FAST_FMAF 1
00175 #define __GCC_ASM_FLAG_OUTPUTS__ 1
00176 #define __GCC_ATOMIC_BOOL_LOCK_FREE 2
00177 #define __GCC_ATOMIC_CHAR16_T_LOCK_FREE 2
00178 #define __GCC_ATOMIC_CHAR32_T_LOCK_FREE 2
00179 #define __GCC_ATOMIC_CHAR_LOCK_FREE 2
00180 #define __GCC_ATOMIC_INT_LOCK_FREE 2
00181 #define __GCC_ATOMIC_LLONG_LOCK_FREE 2
00182 #define __GCC_ATOMIC_LONG_LOCK_FREE 2
00183 #define __GCC_ATOMIC_POINTER_LOCK_FREE 2
00184 #define __GCC_ATOMIC_SHORT_LOCK_FREE 2
00185 #define __GCC_ATOMIC_TEST_AND_SET_TRUEVAL 1
00186 #define __GCC_ATOMIC_WCHAR_T_LOCK_FREE 2
00187 #define __GCC_CONSTRUCTIVE_SIZE 64
00188 #define __GCC_DESTRUCTIVE_SIZE 64
00189 #define __GCC_HAVE_DWARF2_CFI_ASM 1
00190 #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_1 1
00191 #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_16 1
00192 #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_2 1
00193 #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_4 1
00194 #define __GCC_HAVE_SYNC_COMPARE_AND_SWAP_8 1
00195 #define __GLIBCXX_BITSIZE_INT_N_0 128
00196 #define __GLIBCXX_TYPE_INT_N_0 __int128
00197 #define __GNUC_GNU_INLINE__ 1
00198 #define __GNUC_MINOR__ 2
00199 #define __GNUC_PATCHLEVEL__ 1
00200 #define __GNUC__ 4
00201 #define __GNUG__ 4
00202 #define __GXX_ABI_VERSION 1002
00203 #define __GXX_EXPERIMENTAL_CXX0X__ 1
```

```
00204 #define __GXX_RTTI 1
00205 #define __GXX_WEAK__ 1
00206 #define __HAVE_FUNCTION_MULTI_VERSIONING 1
00207 #define __INT16_C_SUFFIX__
00208 #define __INT16_FMTd__ "hd"
00209 #define __INT16_FMTi__ "hi"
00210 #define __INT16_MAX__ 32767
00211 #define __INT16_TYPE__ short
00212 #define __INT32_C_SUFFIX__
00213 #define __INT32_FMTd__ "d"
00214 #define __INT32_FMTi__ "i"
00215 #define __INT32_MAX__ 2147483647
00216 #define __INT32_TYPE__ int
00217 #define __INT64_C_SUFFIX__ LL
00218 #define __INT64_FMTd__ "lld"
00219 #define __INT64_FMTi__ "lli"
00220 #define __INT64_MAX__ 9223372036854775807LL
00221 #define __INT64_TYPE__ long long int
00222 #define __INT8_C_SUFFIX__
00223 #define __INT8_FMTd__ "hhd"
00224 #define __INT8_FMTi__ "hhi"
00225 #define __INT8_MAX__ 127
00226 #define __INT8_TYPE__ signed char
00227 #define __INTMAX_C_SUFFIX__ L
00228 #define __INTMAX_FMTd__ "ld"
00229 #define __INTMAX_FMTi__ "li"
00230 #define __INTMAX_MAX__ 9223372036854775807L
00231 #define __INTMAX_TYPE__ long int
00232 #define __INTMAX_WIDTH__ 64
00233 #define __INTPTR_FMTd__ "ld"
00234 #define __INTPTR_FMTi__ "li"
00235 #define __INTPTR_MAX__ 9223372036854775807L
00236 #define __INTPTR_TYPE__ long int
00237 #define __INTPTR_WIDTH__ 64
00238 #define __INT_FAST16_FMTd__ "hd"
00239 #define __INT_FAST16_FMTi__ "hi"
00240 #define __INT_FAST16_MAX__ 32767
00241 #define __INT_FAST16_TYPE__ short
00242 #define __INT_FAST16_WIDTH__ 16
00243 #define __INT_FAST32_FMTd__ "d"
00244 #define __INT_FAST32_FMTi__ "i"
00245 #define __INT_FAST32_MAX__ 2147483647
00246 #define __INT_FAST32_TYPE__ int
00247 #define __INT_FAST32_WIDTH__ 32
00248 #define __INT_FAST64_FMTd__ "lld"
00249 #define __INT_FAST64_FMTi__ "lli"
00250 #define __INT_FAST64_MAX__ 9223372036854775807LL
00251 #define __INT_FAST64_TYPE__ long long int
00252 #define __INT_FAST64_WIDTH__ 64
00253 #define __INT_FAST8_FMTd__ "hhd"
00254 #define __INT_FAST8_FMTi__ "hhi"
00255 #define __INT_FAST8_MAX__ 127
00256 #define __INT_FAST8_TYPE__ signed char
00257 #define __INT_FAST8_WIDTH__ 8
00258 #define __INT_LEAST16_FMTd__ "hd"
00259 #define __INT_LEAST16_FMTi__ "hi"
00260 #define __INT_LEAST16_MAX__ 32767
00261 #define __INT_LEAST16_TYPE__ short
00262 #define __INT_LEAST16_WIDTH__ 16
00263 #define __INT_LEAST32_FMTd__ "d"
00264 #define __INT_LEAST32_FMTi__ "i"
00265 #define __INT_LEAST32_MAX__ 2147483647
00266 #define __INT_LEAST32_TYPE__ int
00267 #define __INT_LEAST32_WIDTH__ 32
00268 #define __INT_LEAST64_FMTd__ "lld"
00269 #define __INT_LEAST64_FMTi__ "lli"
00270 #define __INT_LEAST64_MAX__ 9223372036854775807LL
00271 #define __INT_LEAST64_TYPE__ long long int
00272 #define __INT_LEAST64_WIDTH__ 64
00273 #define __INT_LEAST8_FMTd__ "hhd"
00274 #define __INT_LEAST8_FMTi__ "hhi"
00275 #define __INT_LEAST8_MAX__ 127
00276 #define __INT_LEAST8_TYPE__ signed char
00277 #define __INT_LEAST8_WIDTH__ 8
00278 #define __INT_MAX__ 2147483647
00279 #define __INT_WIDTH__ 32
00280 #define __LDBL_DECIMAL_DIG__ 17
00281 #define __LDBL_DENORM_MIN__ 4.9406564584124654e-324L
00282 #define __LDBL_DIG__ 15
00283 #define __LDBL_EPSILON__ 2.2204460492503131e-16L
00284 #define __LDBL_HAS_DENORM__ 1
00285 #define __LDBL_HAS_INFINITY__ 1
00286 #define __LDBL_HAS_QUIET_NAN__ 1
00287 #define __LDBL_MANT_DIG__ 53
00288 #define __LDBL_MAX_10_EXP__ 308
00289 #define __LDBL_MAX_EXP__ 1024
00290 #define __LDBL_MAX__ 1.7976931348623157e+308L
```

```
00291 #define __LDBL_MIN_10_EXP__ (-307)
00292 #define __LDBL_MIN_EXP__ (-1021)
00293 #define __LDBL_MIN__ 2.2250738585072014e-308L
00294 #define __LDBL_NORM_MAX__ 1.7976931348623157e+308L
00295 #define __LITTLE_ENDIAN__ 1
00296 #define __LLONG_WIDTH__ 64
00297 #define __LONG_LONG_MAX__ 9223372036854775807LL
00298 #define __LONG_MAX__ 9223372036854775807L
00299 #define __LONG_WIDTH__ 64
00300 #define __LP64__ 1
00301 #define __MACH__ 1
00302 #define __MEMORY_SCOPE_DEVICE 1
00303 #define __MEMORY_SCOPE_SINGLE 4
00304 #define __MEMORY_SCOPE_SYSTEM 0
00305 #define __MEMORY_SCOPE_WRKGRP 2
00306 #define __MEMORY_SCOPE_WVFRNT 3
00307 #define __NO_INLINE__ 1
00308 #define __NO_MATH_ERRNO__ 1
00309 #define __OBJC_BOOL_IS_BOOL 1
00310 #define __OPENCL_MEMORY_SCOPE_ALL_SVM_DEVICES 3
00311 #define __OPENCL_MEMORY_SCOPE_DEVICE 2
00312 #define __OPENCL_MEMORY_SCOPE_SUB_GROUP 4
00313 #define __OPENCL_MEMORY_SCOPE_WORK_GROUP 1
00314 #define __OPENCL_MEMORY_SCOPE_WORK_ITEM 0
00315 #define __ORDER_BIG_ENDIAN__ 4321
00316 #define __ORDER_LITTLE_ENDIAN__ 1234
00317 #define __ORDER_PDP_ENDIAN__ 3412
00318 #define __PIC__ 2
00319 #define __POINTER_WIDTH__ 64
00320 #define __PRAGMA_REDEFINE_EXTNAME 1
00321 #define __PTRDIFF_FMTd__ "ld"
00322 #define __PTRDIFF_FMTi__ "li"
00323 #define __PTRDIFF_MAX__ 9223372036854775807L
00324 #define __PTRDIFF_TYPE__ long int
00325 #define __PTRDIFF_WIDTH__ 64
00326 #define __REGISTER_PREFIX__
00327 #define __SCHAR_MAX__ 127
00328 #define __SHRT_MAX__ 32767
00329 #define __SHRT_WIDTH__ 16
00330 #define __SIG_ATOMIC_MAX__ 2147483647
00331 #define __SIG_ATOMIC_WIDTH__ 32
00332 #define __SIZEOF_DOUBLE__ 8
00333 #define __SIZEOF_FLOAT__ 4
00334 #define __SIZEOF_INT128__ 16
00335 #define __SIZEOF_INT__ 4
00336 #define __SIZEOF_LONG_DOUBLE__ 8
00337 #define __SIZEOF_LONG_LONG__ 8
00338 #define __SIZEOF_LONG__ 8
00339 #define __SIZEOF_POINTER__ 8
00340 #define __SIZEOF_PTRDIFF_T__ 8
00341 #define __SIZEOF_SHORT__ 2
00342 #define __SIZEOF_SIZE_T__ 8
00343 #define __SIZEOF_WCHAR_T__ 4
00344 #define __SIZEOF_WINT_T__ 4
00345 #define __SIZE_FMTX__ "lX"
00346 #define __SIZE_FMTo__ "lo"
00347 #define __SIZE_FMTu__ "lu"
00348 #define __SIZE_FMTx__ "lx"
00349 #define __SIZE_MAX__ 18446744073709551615UL
00350 #define __SIZE_TYPE__ long unsigned int
00351 #define __SIZE_WIDTH__ 64
00352 #define __SSP__ 1
00353 #define __STDCPP_DEFAULT_NEW_ALIGNMENT__ 16UL
00354 #define __STDCPP_THREADS__ 1
00355 #define __STDC_EMBED_EMPTY__ 2
00356 #define __STDC_EMBED_FOUND__ 1
00357 #define __STDC_EMBED_NOT_FOUND__ 0
00358 #define __STDC_HOSTED__ 1
00359 #define __STDC_NO_THREADS__ 1
00360 #define __STDC_UTF_16__ 1
00361 #define __STDC_UTF_32__ 1
00362 #define __STDC__ 1
00363 #define __UINT16_C_SUFFIX__
00364 #define __UINT16_FMTX__ "hX"
00365 #define __UINT16_FMTo__ "ho"
00366 #define __UINT16_FMTu__ "hu"
00367 #define __UINT16_FMTx__ "hx"
00368 #define __UINT16_MAX__ 65535
00369 #define __UINT16_TYPE__ unsigned short
00370 #define __UINT32_C_SUFFIX__ U
00371 #define __UINT32_FMTX__ "X"
00372 #define __UINT32_FMTo__ "o"
00373 #define __UINT32_FMTu__ "u"
00374 #define __UINT32_FMTx__ "x"
00375 #define __UINT32_MAX__ 4294967295U
00376 #define __UINT32_TYPE__ unsigned int
00377 #define __UINT64_C_SUFFIX__ ULL
```

```
00378 #define __UINT64_FMTX__ "llX"
00379 #define __UINT64_FMTo__ "llo"
00380 #define __UINT64_FMTu__ "llu"
00381 #define __UINT64_FMTx__ "llx"
00382 #define __UINT64_MAX__ 18446744073709551615ULL
00383 #define __UINT64_TYPE__ long long unsigned int
00384 #define __UINT8_C_SUFFIX__
00385 #define __UINT8_FMTX__ "hhX"
00386 #define __UINT8_FMTo__ "hho"
00387 #define __UINT8_FMTu__ "hhu"
00388 #define __UINT8_FMTx__ "hhx"
00389 #define __UINT8_MAX__ 255
00390 #define __UINT8_TYPE__ unsigned char
00391 #define __UINTMAX_C_SUFFIX__ UL
00392 #define __UINTMAX_FMTX__ "lX"
00393 #define __UINTMAX_FMTo__ "lo"
00394 #define __UINTMAX_FMTu__ "lu"
00395 #define __UINTMAX_FMTx__ "lx"
00396 #define __UINTMAX_MAX__ 18446744073709551615UL
00397 #define __UINTMAX_TYPE__ long unsigned int
00398 #define __UINTMAX_WIDTH__ 64
00399 #define __UINTPTR_FMTX__ "lX"
00400 #define __UINTPTR_FMTo__ "lo"
00401 #define __UINTPTR_FMTu__ "lu"
00402 #define __UINTPTR_FMTx__ "lx"
00403 #define __UINTPTR_MAX__ 18446744073709551615UL
00404 #define __UINTPTR_TYPE__ long unsigned int
00405 #define __UINTPTR_WIDTH__ 64
00406 #define __UINT_FAST16_FMTX__ "hX"
00407 #define __UINT_FAST16_FMTo__ "ho"
00408 #define __UINT_FAST16_FMTu__ "hu"
00409 #define __UINT_FAST16_FMTx__ "hx"
00410 #define __UINT_FAST16_MAX__ 65535
00411 #define __UINT_FAST16_TYPE__ unsigned short
00412 #define __UINT_FAST32_FMTX__ "X"
00413 #define __UINT_FAST32_FMTo__ "o"
00414 #define __UINT_FAST32_FMTu__ "u"
00415 #define __UINT_FAST32_FMTx__ "x"
00416 #define __UINT_FAST32_MAX__ 4294967295U
00417 #define __UINT_FAST32_TYPE__ unsigned int
00418 #define __UINT_FAST64_FMTX__ "llX"
00419 #define __UINT_FAST64_FMTo__ "llo"
00420 #define __UINT_FAST64_FMTu__ "llu"
00421 #define __UINT_FAST64_FMTx__ "llx"
00422 #define __UINT_FAST64_MAX__ 18446744073709551615ULL
00423 #define __UINT_FAST64_TYPE__ long long unsigned int
00424 #define __UINT_FAST8_FMTX__ "hhX"
00425 #define __UINT_FAST8_FMTo__ "hho"
00426 #define __UINT_FAST8_FMTu__ "hhu"
00427 #define __UINT_FAST8_FMTx__ "hhx"
00428 #define __UINT_FAST8_MAX__ 255
00429 #define __UINT_FAST8_TYPE__ unsigned char
00430 #define __UINT_LEAST16_FMTX__ "hX"
00431 #define __UINT_LEAST16_FMTo__ "ho"
00432 #define __UINT_LEAST16_FMTu__ "hu"
00433 #define __UINT_LEAST16_FMTx__ "hx"
00434 #define __UINT_LEAST16_MAX__ 65535
00435 #define __UINT_LEAST16_TYPE__ unsigned short
00436 #define __UINT_LEAST32_FMTX__ "X"
00437 #define __UINT_LEAST32_FMTo__ "o"
00438 #define __UINT_LEAST32_FMTu__ "u"
00439 #define __UINT_LEAST32_FMTx__ "x"
00440 #define __UINT_LEAST32_MAX__ 4294967295U
00441 #define __UINT_LEAST32_TYPE__ unsigned int
00442 #define __UINT_LEAST64_FMTX__ "llX"
00443 #define __UINT_LEAST64_FMTo__ "llo"
00444 #define __UINT_LEAST64_FMTu__ "llu"
00445 #define __UINT_LEAST64_FMTx__ "llx"
00446 #define __UINT_LEAST64_MAX__ 18446744073709551615ULL
00447 #define __UINT_LEAST64_TYPE__ long long unsigned int
00448 #define __UINT_LEAST8_FMTX__ "hhX"
00449 #define __UINT_LEAST8_FMTo__ "hho"
00450 #define __UINT_LEAST8_FMTu__ "hhu"
00451 #define __UINT_LEAST8_FMTx__ "hhx"
00452 #define __UINT_LEAST8_MAX__ 255
00453 #define __UINT_LEAST8_TYPE__ unsigned char
00454 #define __USER_LABEL_PREFIX__ _
00455 #define __VERSION__ "Apple LLVM 17.0.0 (clang-1700.0.13.3)"
00456 #define __WCHAR_MAX__ 2147483647
00457 #define __WCHAR_TYPE__ int
00458 #define __WCHAR_WIDTH__ 32
00459 #define __WINT_MAX__ 2147483647
00460 #define __WINT_TYPE__ int
00461 #define __WINT_WIDTH__ 32
00462 #define __aarch64__ 1
00463 #define __apple_build_version__ 17000013
00464 #define __arm64 1
```

```
00465 #define __arm64__ 1
00466 #define __block __attribute__((__blocks__(byref)))
00467 #define __clang__ 1
00468 #define __clang_literal_encoding__ "UTF-8"
00469 #define __clang_major__ 17
00470 #define __clang_minor__ 0
00471 #define __clang_patchlevel__ 0
00472 #define __clang_version__ "17.0.0 (clang-1700.0.13.3)"
00473 #define __clang_wide_literal_encoding__ "UTF-32"
00474 #define __cplusplus 201703L
00475 #define __cpp_aggregate_bases 201603L
00476 #define __cpp_aggregate_nsdmi 201304L
00477 #define __cpp_alias_templates 200704L
00478 #define __cpp_aligned_new 201606L
00479 #define __cpp_attributes 200809L
00480 #define __cpp_binary_literals 201304L
00481 #define __cpp_capture_star_this 201603L
00482 #define __cpp_constexpr 201603L
00483 #define __cpp_constexpr_in_decltype 201711L
00484 #define __cpp_decltype 200707L
00485 #define __cpp_decltype_auto 201304L
00486 #define __cpp_deduction_guides 201703L
00487 #define __cpp_delegating_constructors 200604L
00488 #define __cpp_deleted_function 202403L
00489 #define __cpp_digit_separators 201309L
00490 #define __cpp_enumerator_attributes 201411L
00491 #define __cpp_exceptions 199711L
00492 #define __cpp_fold_expressions 201603L
00493 #define __cpp_generic_lambdas 201304L
00494 #define __cpp_guaranteed_copy_elision 201606L
00495 #define __cpp_hex_float 201603L
00496 #define __cpp_if_constexpr 201606L
00497 #define __cpp_impl_destroying_delete 201806L
00498 #define __cpp_inheriting_constructors 201511L
00499 #define __cpp_init_captures 201304L
00500 #define __cpp_initializer_lists 200806L
00501 #define __cpp_inline_variables 201606L
00502 #define __cpp_lambdas 200907L
00503 #define __cpp_named_character_escapes 202207L
00504 #define __cpp_namespace_attributes 201411L
00505 #define __cpp_nested_namespace_definitions 201411L
00506 #define __cpp_noexcept_function_type 201510L
00507 #define __cpp_nontype_template_args 201411L
00508 #define __cpp_nontype_template_parameter_auto 201606L
00509 #define __cpp_nsdmi 200809L
00510 #define __cpp_pack_indexing 202311L
00511 #define __cpp_placeholder_variables 202306L
00512 #define __cpp_range_based_for 201603L
00513 #define __cpp_raw_strings 200710L
00514 #define __cpp_ref_qualifiers 200710L
00515 #define __cpp_return_type_deduction 201304L
00516 #define __cpp_rtti 199711L
00517 #define __cpp_rvalue_references 200610L
00518 #define __cpp_static_assert 201411L
00519 #define __cpp_static_call_operator 202207L
00520 #define __cpp_structured_bindings 202403L
00521 #define __cpp_template_auto 201606L
00522 #define __cpp_template_template_args 201611L
00523 #define __cpp_threadsafe_static_init 200806L
00524 #define __cpp_unicode_characters 200704L
00525 #define __cpp_unicode_literals 200710L
00526 #define __cpp_user_defined_literals 200809L
00527 #define __cpp_variable_templates 201304L
00528 #define __cpp_variadic_templates 200704L
00529 #define __cpp_variadic_using 201611L
00530 #define __llvm__ 1
00531 #define __nonnull _Nonnull
00532 #define __null_unspecified _Null_unspecified
00533 #define __nullable _Nullable
00534 #define __pic__ 2
00535 #define __private_extern__ extern
00536 #define __strong
00537 #define __unsafe_unretained
00538 #define __weak __attribute__((objc_gc(weak)))
```

9.49 build/NetScanner_autogen/mocs_compilation.cpp 文件参考

```
#include "EWIEGA46WW/moc_deviceanalyzer.cpp"
#include "EWIEGA46WW/moc_mainwindow.cpp"
#include "EWIEGA46WW/moc_networkscanner.cpp"
```

```
#include "EWIEGA46WW/moc_networktopology.cpp"
#include "EWIEGA46WW/moc_scanhistory.cpp"
```

mocs_compilation.cpp 的引用(Include)关系图:

9.50 NetScanner_autogen/mocs_compilation.cpp 文件参考

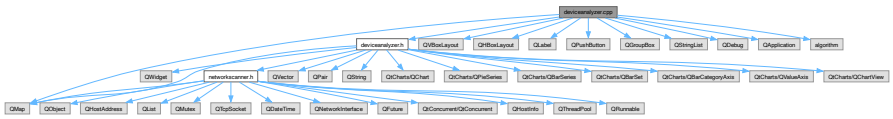
```
#include "JRIAJ772TK/moc_deviceanalyzer.cpp"
#include "JRIAJ772TK/moc_mainwindow.cpp"
#include "JRIAJ772TK/moc_networkscanner.cpp"
#include "JRIAJ772TK/moc_networktopology.cpp"
#include "JRIAJ772TK/moc_scanhistory.cpp"
#include "EWIEGA46WW/moc_deviceanalyzer.cpp"
#include "EWIEGA46WW/moc_mainwindow.cpp"
```

mocs_compilation.cpp 的引用(Include)关系图:

9.51 deviceanalyzer.cpp 文件参考

```
#include "deviceanalyzer.h"
#include <QVBoxLayout>
#include <QHBoxLayout>
#include <QLabel>
#include <QPushButton>
#include <QGroupBox>
#include <QMap>
#include <QStringList>
#include <QDebug>
#include <QApplication>
#include <algorithm>
```

deviceanalyzer.cpp 的引用(Include)关系图:

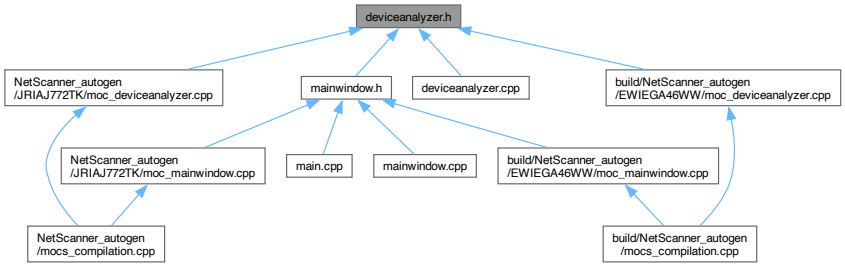


9.52 deviceanalyzer.h 文件参考

```
#include <QWidget>
#include <QMap>
#include <QVector>
#include <QPair>
#include <QString>
#include <QtCharts/QChart>
#include <QtCharts/QPieSeries>
#include <QtCharts/QBarSeries>
#include <QtCharts/QBarSet>
#include <QtCharts/QBarCategoryAxis>
#include <QtCharts/QValueAxis>
#include <QtCharts/QChartView>
#include "networkscanner.h"
```

deviceanalyzer.h 的引用(Include)关系图:

此图展示该文件被哪些文件直接或间接地引用了:



类

- class DeviceAnalyzer

9.53 deviceanalyzer.h

浏览该文件的文档.

```
00001 #ifndef DEVICEANALYZER_H
00002 #define DEVICEANALYZER_H
00003
00004 #include <QWidget>
00005 #include <QMap>
00006 #include <QVector>
00007 #include <QPair>
00008 #include <QString>
00009 #include <QtCharts/QChart>
00010 #include <QtCharts/QPieSeries>
00011 #include <QtCharts/QBarSeries>
00012 #include <QtCharts/QBarSet>
```

```
00013 #include <QtCharts/QBarCategoryAxis>
00014 #include <QtCharts/QValueAxis>
00015 #include <QtCharts/QChartView>
00016 #include "networkscanner.h"
00017
00018 // 设备分析器类 - 提供对扫描结果的统计分析
00019 class DeviceAnalyzer : public QWidget
00020 {
00021     Q_OBJECT
00022
00023 public:
00024     DeviceAnalyzer(QWidget *parent = nullptr);
00025
00026     // 分析扫描结果并更新统计图表
00027     void analyzeHosts(const QList<HostInfo> &hosts);
00028     void clear();
00029
00030     // 获取各种统计信息
00031     int getTotalHostsCount() const { return m_totalHosts; }
00032     int getReachableHostsCount() const { return m_reachableHosts; }
00033     int getUnreachableHostsCount() const { return m_totalHosts - m_reachableHosts; }
00034
00035     // 获取各种图表
00036     QChart* getDeviceTypeChart() const { return m_deviceTypeChart; }
00037     QChart* getPortDistributionChart() const { return m_portDistributionChart; }
00038     QChart* getVendorDistributionChart() const { return m_vendorDistributionChart; }
00039
00040     // 创建安全风险报告
00041     QString generateSecurityReport(const QList<HostInfo> &hosts);
00042
00043 signals:
00044     void analysisCompleted();
00045
00046 private:
00047     // 扫描统计
00048     int m_totalHosts;
00049     int m_reachableHosts;
00050
00051     // 设备类型分布图表
00052     QChart *m_deviceTypeChart;
00053     QPieSeries *m_deviceTypeSeries;
00054
00055     // 端口分布图表
00056     QChart *m_portDistributionChart;
00057     QBarSeries *m_portSeries;
00058
00059     // 厂商分布图表
00060     QChart *m_vendorDistributionChart;
00061     QPieSeries *m_vendorSeries;
00062
00063     // 创建各类图表
00064     void createDeviceTypeChart();
00065     void createPortDistributionChart();
00066     void createVendorDistributionChart();
00067
00068     // 设备类型判断
00069     QString determineDeviceType(const HostInfo &host);
00070 };
00071
00072 #endif // DEVICEANALYZER_H
```

9.54 main.cpp 文件参考

```
#include "mainwindow.h"
#include <QApplication>
#include <QFontDatabase>
#include <QSplashScreen>
#include <QPixmap>
#include <QTimer>
#include <QFont>
#include <QStyleFactory>
```

main.cpp 的引用(Include)关系图:

函数

- int main (int argc, char ∗argv[ ])

9.54.1 函数说明

9.54.1.1 main()

```
int main (
            int argc,
            char * argv[ ])
```

9.55 mainwindow.cpp 文件参考

```
#include "mainwindow.h"
#include <QColor>
#include <QDateTime>
#include <QStandardPaths>
#include <QTimer>
#include <QPalette>
#include <QPixmap>
#include <QPainter>
#include <QStyleFactory>
#include <QInputDialog>
#include <QCalendarWidget>
#include <QScrollArea>
#include <QTimeEdit>
```

mainwindow.cpp 的引用(Include)关系图:

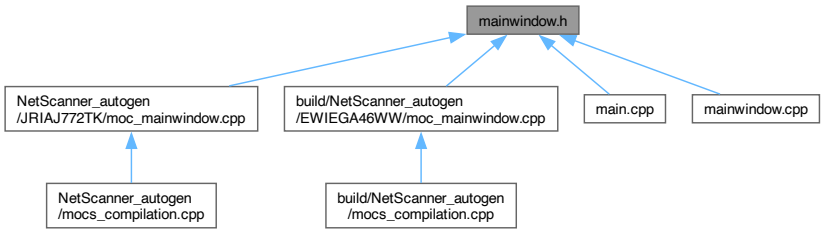
9.56 mainwindow.h 文件参考

```
#include <QMainWindow>
#include <QTableWidget>
#include <QPushButton>
#include <QProgressBar>
#include <QLabel>
#include <QVBoxLayout>
#include <QHBoxLayout>
#include <QStatusBar>
#include <QHeaderView>
#include <QMessageBox>
#include <QLineEdit>
#include <QComboBox>
#include <QSpinBox>
#include <QCheckBox>
#include <QGroupBox>
```

```
#include <QFileDialog>
#include <QAction>
#include <QMenu>
#include <QMenuBar>
#include <QTabWidget>
#include <QTextEdit>
#include <QSettings>
#include <QSplitter>
#include <QStyledItemDelegate>
#include <QApplication>
#include <QtCharts/QChartView>
#include <QtCharts/QChart>
#include <QGraphicsView>
#include <QGraphicsScene>
#include "networkscanner.h"
#include "networktopology.h"
#include "deviceanalyzer.h"
#include "scanhistory.h"
```

mainwindow.h 的引用(Include)关系图:

此图展示该文件被哪些文件直接或间接地引用了:



类

- class MainWindow

9.57 mainwindow.h

浏览该文件的文档.

```
00001 #ifndef MAINWINDOW_H
00002 #define MAINWINDOW_H
00003
00004 #include <QMainWindow>
00005 #include <QTableWidget>
00006 #include <QPushButton>
00007 #include <QProgressBar>
00008 #include <QLabel>
00009 #include <QVBoxLayout>
00010 #include <QHBoxLayout>
00011 #include <QStatusBar>
00012 #include <QHeaderView>
```

```
00013 #include <QMessageBox>
00014 #include <QLineEdit>
00015 #include <QComboBox>
00016 #include <QSpinBox>
00017 #include <QCheckBox>
00018 #include <QGroupBox>
00019 #include <QFileDialog>
00020 #include <QAction>
00021 #include <QMenu>
00022 #include <QMenuBar>
00023 #include <QTabWidget>
00024 #include <QTextEdit>
00025 #include <QSettings>
00026 #include <QSplitter>
00027 #include <QStyledItemDelegate>
00028 #include <QApplication>
00029 #include <QtCharts/QChartView>
00030 #include <QtCharts/QChart>
00031 #include <QGraphicsView>
00032 #include <QGraphicsScene>
00033
00034 #include "networkscanner.h"
00035 #include "networktopology.h"
00036 #include "deviceanalyzer.h"
00037 #include "scanhistory.h"
00038
00039 // QtCharts命名空间已经在deviceanalyzer.h中引入
00040 // using namespace QtCharts;
00041
00042 class MainWindow : public QMainWindow
00043 {
00044     Q_OBJECT
00045
00046 public:
00047     MainWindow(QWidget *parent = nullptr);
00048     ~MainWindow();
00049
00050 private slots:
00051     void startScan();
00052     void stopScan();
00053     void onHostFound(const HostInfo &host);
00054     void onScanStarted();
00055     void onScanFinished();
00056     void onScanProgress(int progress);
00057     void onScanError(const QString &errorMessage);
00058
00059     // 现有功能槽
00060     void saveResults();
00061     void clearResults();
00062     void showSettings();
00063     void applySettings();
00064     void showAbout();
00065     void showHostDetails(int row, int column);
00066     void exportToCSV();
00067     void togglePortScanOptions(bool checked);
00068     void toggleRangeOptions(bool checked);
00069
00070     // 新增功能槽
00071     void showTopologyView();
00072     void showStatisticsView();
00073     void showHistoryView();
00074     void generateSecurityReport();
00075     void saveTopologyImage();
00076     void toggleDarkMode(bool enable);
00077     void compareScanResults();
00078     void scheduleScan();
00079     void saveHistoryToFile();
00080     void loadHistoryFromFile();
00081     void updateNetworkTopology();
00082     void refreshTopology();
00083     void filterResults();
00084     void clearFilters();
00085     void onThemeChanged();
00086
00087 private:
00088     void createUI();
00089     void createMenus();
00090     void createSettingsDialog();
00091     void createTopologyTab();
00092     void createStatisticsTab();
00093     void createHistoryTab();
00094     void createDetailsTab();
00095     void createSecurityTab();
00096     void setupConnections();
00097     void updatePortsList();
00098     void loadSettings();
00099     void saveSettings();
```

```
00100     void updateStatistics();
00101     void applyTheme(bool darkMode);
00102
00103     // UI元素
00104     QWidget *m_centralWidget;
00105     QTabWidget *m_tabWidget;
00106
00107     // 扫描结果标签页
00108     QWidget *m_scanTab;
00109     QVBoxLayout *m_mainLayout;
00110     QHBoxLayout *m_controlLayout;
00111     QTableWidget *m_resultsTable;
00112     QPushButton *m_scanButton;
00113     QPushButton *m_stopButton;
00114     QPushButton *m_clearButton;
00115     QPushButton *m_saveButton;
00116     QProgressBar *m_progressBar;
00117     QLabel *m_statusLabel;
00118     QStatusBar *m_statusBar;
00119
00120     // 扫描设置标签页
00121     QWidget *m_settingsTab;
00122     QVBoxLayout *m_settingsLayout;
00123
00124     // 端口设置区域
00125     QGroupBox *m_portsGroupBox;
00126     QCheckBox *m_customPortsCheckBox;
00127     QLineEdit *m_portsLineEdit;
00128     QSpinBox *m_timeoutSpinBox;
00129
00130     // IP范围设置区域
00131     QGroupBox *m_rangeGroupBox;
00132     QCheckBox *m_customRangeCheckBox;
00133     QLineEdit *m_startIPLineEdit;
00134     QLineEdit *m_endIPLineEdit;
00135
00136     // 主机详情标签页
00137     QWidget *m_detailsTab;
00138     QVBoxLayout *m_detailsLayout;
00139     QTextEdit *m_detailsTextEdit;
00140
00141     // 网络拓扑标签页
00142     QWidget *m_topologyTab;
00143     NetworkTopology *m_networkTopology;
00144
00145     // 统计分析标签页
00146     QWidget *m_statisticsTab;
00147     DeviceAnalyzer *m_deviceAnalyzer;
00148     QChartView *m_deviceTypeChartView;
00149     QChartView *m_vendorChartView;
00150     QChartView *m_portDistributionChartView;
00151     QTextEdit *m_securityReportText;
00152
00153     // 扫描历史标签页
00154     QWidget *m_historyTab;
00155     ScanHistory *m_scanHistory;
00156     QComboBox *m_sessionComboBox;
00157     QTableWidget *m_historyTable;
00158
00159     // 菜单项
00160     QMenu *m_fileMenu;
00161     QMenu *m_viewMenu;
00162     QMenu *m_toolsMenu;
00163     QMenu *m_helpMenu;
00164     QAction *m_exportAction;
00165     QAction *m_saveHistoryAction;
00166     QAction *m_loadHistoryAction;
00167     QAction *m_saveTopologyAction;
00168     QAction *m_exitAction;
00169     QAction *m_settingsAction;
00170     QAction *m_darkModeAction;
00171     QAction *m_scheduleScanAction;
00172     QAction *m_aboutAction;
00173
00174     // 过滤控件
00175     QWidget *m_filterWidget;
00176     QLineEdit *m_filterIPLineEdit;
00177     QComboBox *m_filterVendorComboBox;
00178     QComboBox *m_filterTypeComboBox;
00179     QPushButton *m_filterButton;
00180     QPushButton *m_clearFilterButton;
00181
00182     // 网络扫描器
00183     NetworkScanner *m_scanner;
00184
00185     // 扫描的主机数量
00186     int m_hostsFound;
```

```
00187
00188     // 当前查看的主机索引
00189     int m_currentHostIndex;
00190
00191     // 主题设置
00192     bool m_darkModeEnabled;
00193 };
00194
00195 #endif // MAINWINDOW_H
```

9.58 networkscanner.cpp 文件参考

网络扫描器类的实现

```
#include "networkscanner.h"
#include <QDebug>
#include <QTime>
#include <QTimer>
#include <QMutexLocker>
#include <QProcess>
#include <QFile>
#include <QTextStream>
#include <QCoreApplication>
#include <QDir>
#include <QRegularExpression>
#include <QNetworkInterface>
#include <QThread>
#include <QEventLoop>
#include <QElapsedTimer>
#include <QMessageBox>
```

networkscanner.cpp 的引用(Include)关系图:

9.58.1 详细描述

网络扫描器类的实现

提供网络设备发现和端口扫描功能的实现

作者

Network Scanner Team

版本

2.1.0

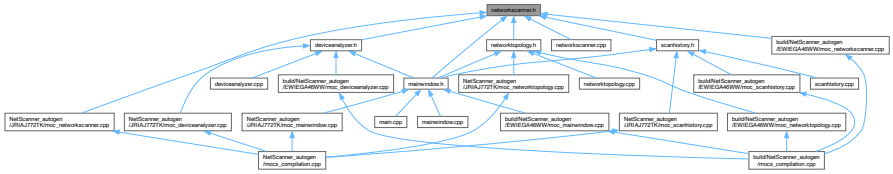
9.59 networkscanner.h 文件参考

网络扫描器类定义

```
#include <QObject>
#include <QHostAddress>
#include <QList>
#include <QMap>
#include <QMutex>
#include <QTcpSocket>
#include <QDateTime>
#include <QNetworkInterface>
#include <QFuture>
#include <QtConcurrent/QtConcurrent>
#include <QHostInfo>
#include <QThreadPool>
#include <QRunnable>
```

networkscanner.h 的引用(Include)关系图:

此图展示该文件被哪些文件直接或间接地引用了:



类

- struct HostInfo
  存储主机信息的结构体
- class ScanStrategy
  扫描策略类
- class ScanTask
  扫描任务类
- class NetworkScanner
  网络扫描器类

9.59.1 详细描述

网络扫描器类定义

提供网络设备发现和端口扫描功能

作者

Network Scanner Team

版本

2.1.0

9.60 networkscanner.h

浏览该文件的文档.

```
00001 //
00002 // networkscanner.h
00003 //
00004
00012
00013 #ifndef NETWORKSCANNER_H
00014 #define NETWORKSCANNER_H
00015
00016 #include <QObject>
00017 #include <QHostAddress>
00018 #include <QList>
00019 #include <QMap>
00020 #include <QMutex>
00021 #include <QTcpSocket>
00022 #include <QDateTime>
00023 #include <QNetworkInterface>
00024 #include <QFuture>
00025 #include <QtConcurrent/QtConcurrent>
00026 #include <QHostInfo>
00027 #include <QThreadPool>
00028 #include <QRunnable>
00029
00035 struct HostInfo {
00036     QString ipAddress;
00037     QString hostName;
00038     QString macAddress;
00039     QString macVendor;
00040     bool isReachable;
00041     QDateTime scanTime;
00042     QMap<int, bool> openPorts;
00043 };
00044
00050 class ScanStrategy {
00051 public:
00056     enum ScanMode {
00057         QUICK_SCAN,
00058         STANDARD_SCAN,
00059         DEEP_SCAN
00060     };
00061
00066     ScanStrategy(ScanMode mode = STANDARD_SCAN);
00067
00072     QList<int> getPortsToScan() const;
00073
00079     int getScanTimeout(const QString &ip) const;
00080
00085     int getMaxParallelTasks() const;
00086
00092     void updateHostResponseTime(const QString &ip, int responseTime);
00093
00098     ScanMode getMode() const { return m_mode; }
00099
00104     void setMode(ScanMode mode) { m_mode = mode; }
00105
00106 private:
00107     ScanMode m_mode;
00108     QMap<QString, int> m_hostResponseTimes;
00109 };
00110
00116 class ScanTask : public QRunnable {
00117 public:
00125     ScanTask(QObject* parent, const QHostAddress &address,
00126              const QList<int> &ports, int timeout);
00127
00132     void run() override;
00133
```

```
00134     QObject* m_parent;
00135     QHostAddress m_address;
00136     QList<int> m_ports;
00137     int m_timeout;
00138 };
00139
00145 class NetworkScanner : public QObject
00146 {
00147     Q_OBJECT
00148
00149 public:
00154     NetworkScanner(QObject *parent = nullptr);
00155
00159     ~NetworkScanner();
00160
00165     void setCustomPortsToScan(const QList<int> &ports);
00166
00171     void setScanTimeout(int msecs);
00172
00178     void setCustomIPRange(const QString &startIP, const QString &endIP);
00179
00184     void setScanStrategy(ScanStrategy::ScanMode mode);
00185
00190     QList<HostInfo> getScannedHosts() const;
00191
00196     void saveResultsToFile(const QString &filename) const;
00197
00201     void startScan();
00202
00206     void stopScan();
00207
00212     bool isScanning() const;
00213
00219     QList<QHostAddress> quickPingScan(const QList<QHostAddress> &addresses);
00220
00227     bool isHostReachable(const QHostAddress &address, int timeout);
00228
00236     bool isReachableOnPorts(const QHostAddress &address, const QList<int> &ports, int timeout);
00237
00242     void scanHost(const QHostAddress &address);
00243
00249     QString lookupHostName(const QHostAddress &address);
00250
00256     QString lookupMacAddress(const QHostAddress &address);
00257
00263     QString lookupMacVendor(const QString &macAddress);
00264
00270     QString generatePseudoMACFromIP(const QString &ip);
00271
00272 signals:
00277     void hostFound(const HostInfo &host);
00278
00282     void scanStarted();
00283
00287     void scanFinished();
00288
00293     void scanProgress(int progress);
00294
00299     void scanError(const QString &errorMessage);
00300
00301 public slots:
00306     void onScanTaskFinished(const HostInfo &hostInfo);
00307
00311     void updateScanProgress();
00312
00317     void onHostNameLookedUp(const QHostInfo &hostInfo);
00318
00319 private:
00324     QList<QHostAddress> getLocalNetworkAddresses();
00325
00331     QString normalizeMacAddress(const QString &macAddress);
00332
00339     bool pingTargetWithTimeout(const QString &ip, int timeout);
00340
00346     QString getMacAddressFromSystemCalls(const QString &ip);
00347
00352     QList<QPair<QHostAddress, QString>> performARPScan();
00353
00358     void scanHostPorts(HostInfo &hostInfo);
00359
00364     QList<QHostAddress> getAddressesToScan();
00365
00369     void processScanResults();
00370
00371     bool m_isScanning;
00372
00373     int m_scannedHosts;
```

```
00374     int m_totalHosts;
00375     int m_scanTimeout;
00376
00377     bool m_useCustomRange;
00378     QHostAddress m_startIPRange;
00379     QHostAddress m_endIPRange;
00380
00381     QList<int> m_portsToScan;
00382
00383     QList<HostInfo> m_scannedHostsList;
00384     QMutex m_mutex;
00385
00386     QList<QFuture<void>> m_scanFutures;
00387     QThreadPool m_threadPool;
00388
00389     QMap<QString, QString> m_macAddressCache;
00390
00391     ScanStrategy m_scanStrategy;
00392
00393     QList<QHostAddress> m_activeHosts;
00394
00405     static bool executeProcess(const QString &program, const QStringList &args, QString &stdOutOutput,
      QString &stdErrOutput, int startTimeout = 2000, int finishTimeout = 5000);
00406 };
00407
00408 #endif // NETWORKSCANNER_H
```

9.61 networktopology.cpp 文件参考

```
#include "networktopology.h"
#include <QPainter>
#include <QGraphicsSceneMouseEvent>
#include <QStyleOptionGraphicsItem>
#include <QVBoxLayout>
#include <QHBoxLayout>
#include <QLabel>
#include <QPushButton>
#include <QComboBox>
#include <QApplication>
#include <QPalette>
#include <QDebug>
#include <QtMath>
#include <QGraphicsItemAnimation>
#include <QTimeLine>
#include <QToolTip>
#include <QScrollBar>
#include <QProcess>
#include <QRegularExpression>
```

networktopology.cpp 的引用(Include)关系图:

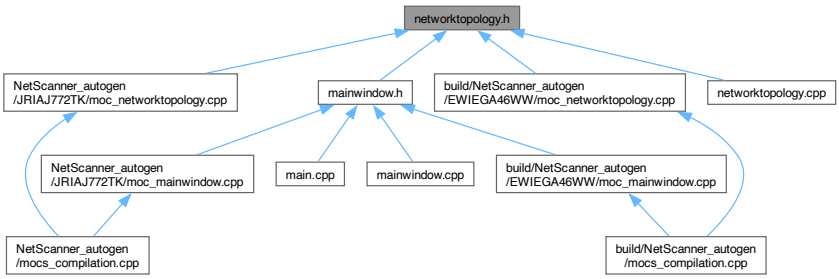
9.62 networktopology.h 文件参考

```
#include <QWidget>
#include <QGraphicsView>
#include <QGraphicsScene>
#include <QGraphicsItem>
```

```
#include <QMap>
#include <QPair>
#include <QProcess>
#include "networkscanner.h"
```

networktopology.h 的引用(Include)关系图:

此图展示该文件被哪些文件直接或间接地引用了:



类

- class TopologyAnalyzer
- class DeviceNode
- class ConnectionLine
- class NetworkTopologyView
- class NetworkTopology

枚举

- enum DeviceType {
  DEVICE_UNKNOWN , DEVICE_ROUTER , DEVICE_SERVER , DEVICE_PC ,
  DEVICE_MOBILE , DEVICE_PRINTER , DEVICE_IOT }
- enum ConnectionType {
  CONNECTION_UNKNOWN , CONNECTION_DIRECT , CONNECTION_WIRELESS , CONNECTION_VPN ,
  CONNECTION_ROUTED }

9.62.1 枚举类型说明

9.62.1.1 ConnectionType

```
enum ConnectionType
```

枚举值

CONNECTION_UNKNOWN	
CONNECTION_DIRECT	
CONNECTION_WIRELESS	
CONNECTION_VPN	
CONNECTION_ROUTED	

9.62.1.2 DeviceType

```
enum DeviceType
```

枚举值

DEVICE_UNKNOWN	
DEVICE_ROUTER	
DEVICE_SERVER	
DEVICE_PC	
DEVICE_MOBILE	
DEVICE_PRINTER	
DEVICE_IOT	

9.63 networktopology.h

浏览该文件的文档.

```
00001 #ifndef NETWORKTOPOLOGY_H
00002 #define NETWORKTOPOLOGY_H
00003
00004 #include <QWidget>
00005 #include <QGraphicsView>
00006 #include <QGraphicsScene>
00007 #include <QGraphicsItem>
00008 #include <QMap>
00009 #include <QPair>
00010 #include <QProcess>
00011 #include "networkscanner.h"
00012
00013 // 定义设备类型枚举
00014 enum DeviceType {
00015     DEVICE_UNKNOWN,
00016     DEVICE_ROUTER,
00017     DEVICE_SERVER,
00018     DEVICE_PC,
00019     DEVICE_MOBILE,
00020     DEVICE_PRINTER,
00021     DEVICE_IOT
00022 };
00023
00024 // 设备连接类型
00025 enum ConnectionType {
00026     CONNECTION_UNKNOWN,
00027     CONNECTION_DIRECT,      // 直接连接
00028     CONNECTION_WIRELESS,    // 无线连接
00029     CONNECTION_VPN,         // VPN连接
00030     CONNECTION_ROUTED       // 通过路由器连接
00031 };
00032
00033 // 拓扑分析器类
00034 class TopologyAnalyzer {
00035 public:
00036     TopologyAnalyzer();
00037
```

```
00038     // 推断设备连接关系
00039     QMap<QString, QStringList> inferDeviceConnections(const QList<HostInfo> &hosts);
00040
00041     // 分析网络层次结构
00042     QMap<int, QStringList> analyzeTTLLayers(const QList<HostInfo> &hosts);
00043
00044     // 分析子网结构
00045     QMap<QString, QStringList> analyzeSubnets(const QList<HostInfo> &hosts);
00046
00047     // 基于响应时间的设备聚类
00048     QList<QStringList> clusterDevicesByResponseTime(const QList<HostInfo> &hosts);
00049
00050     // 获取特定IP的TTL值
00051     int getTTLValue(const QString &ipAddress);
00052
00053     // 执行traceroute命令获取路径信息
00054     QStringList performTraceRoute(const QString &targetIP);
00055
00056     // 计算子网掩码 - 改为公共方法
00057     QString calculateSubnet(const QString &ip, int prefixLength = 24);
00058
00059     // 判断两个IP是否在同一子网
00060     bool inSameSubnet(const QString &ip1, const QString &ip2, int prefixLength = 24);
00061
00062 private:
00063     // 已移除private中的calculateSubnet声明
00064 };
00065
00066 // 设备节点
00067 class DeviceNode : public QGraphicsItem
00068 {
00069 public:
00070     DeviceNode(const HostInfo &host, DeviceType type = DEVICE_UNKNOWN);
00071
00072     QRectF boundingRect() const override;
00073     void paint(QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget) override;
00074
00075     void setDeviceType(DeviceType type);
00076     DeviceType deviceType() const { return m_type; }
00077
00078     QString ipAddress() const { return m_host.ipAddress; }
00079     HostInfo hostInfo() const { return m_host; }
00080     void setPosition(const QPointF &pos);
00081
00082     // 新增：设置网络层级
00083     void setNetworkLayer(int layer) { m_networkLayer = layer; }
00084     int networkLayer() const { return m_networkLayer; }
00085
00086     // 新增：设置子网组
00087     void setSubnetGroup(const QString &subnet) { m_subnetGroup = subnet; }
00088     QString subnetGroup() const { return m_subnetGroup; }
00089
00090 protected:
00091     void mousePressEvent(QGraphicsSceneMouseEvent *event) override;
00092     void mouseMoveEvent(QGraphicsSceneMouseEvent *event) override;
00093     void mouseReleaseEvent(QGraphicsSceneMouseEvent *event) override;
00094     void hoverEnterEvent(QGraphicsSceneHoverEvent *event) override;
00095     void hoverLeaveEvent(QGraphicsSceneHoverEvent *event) override;
00096
00097 private:
00098     HostInfo m_host;
00099     DeviceType m_type;
00100     bool m_highlight;
00101     QPointF m_dragStartPosition;
00102     int m_networkLayer;       // 网络层级
00103     QString m_subnetGroup;    // 子网组
00104 };
00105
00106 // 连接线
00107 class ConnectionLine : public QGraphicsItem
00108 {
00109 public:
00110     ConnectionLine(DeviceNode *source, DeviceNode *target,
00111                    ConnectionType type = CONNECTION_DIRECT);
00112
00113     QRectF boundingRect() const override;
00114     void paint(QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget) override;
00115
00116     void updatePosition();
00117
00118     // 设置连接类型
00119     void setConnectionType(ConnectionType type);
00120     ConnectionType connectionType() const { return m_connectionType; }
00121
00122 private:
00123     DeviceNode *m_source;
00124     DeviceNode *m_target;
```

```
00125     ConnectionType m_connectionType;
00126 };
00127
00128 // 拓扑图视图
00129 class NetworkTopologyView : public QGraphicsView
00130 {
00131     Q_OBJECT
00132
00133 public:
00134     NetworkTopologyView(QWidget *parent = nullptr);
00135
00136     void setHosts(const QList<HostInfo> &hosts);
00137     void clear();
00138     void autoLayout();
00139
00140     // 新增布局方法
00141     void hierarchicalLayout(const QMap<int, QStringList> &layers);
00142     void groupedLayout(const QMap<QString, QStringList> &groups);
00143
00144 signals:
00145     void nodeSelected(const HostInfo &host);
00146
00147 private:
00148     QGraphicsScene *m_scene;
00149     QMap<QString, DeviceNode*> m_nodes;
00150     QList<ConnectionLine*> m_connections;
00151
00152     TopologyAnalyzer m_analyzer;
00153
00154     DeviceType determineDeviceType(const HostInfo &host);
00155     void createConnection(DeviceNode *source, DeviceNode *target,
00156                           ConnectionType type = CONNECTION_DIRECT);
00157 };
00158
00159 // 网络拓扑组件（包含拓扑图和控制面板）
00160 class NetworkTopology : public QWidget
00161 {
00162     Q_OBJECT
00163
00164 public:
00165     NetworkTopology(QWidget *parent = nullptr);
00166
00167     void updateTopology(const QList<HostInfo> &hosts);
00168     void clear();
00169
00170     // 缩放和视图控制方法
00171     void scale(qreal factor);
00172     void resetView();
00173
00174     // 新增：切换布局方式
00175     void setLayoutMode(int mode);
00176
00177 signals:
00178     void deviceSelected(const HostInfo &host);
00179
00180 private:
00181     NetworkTopologyView *m_topologyView;
00182     QWidget *m_controlPanel;
00183
00184     enum LayoutMode {
00185         LAYOUT_AUTO,
00186         LAYOUT_HIERARCHICAL,
00187         LAYOUT_GROUPED
00188     };
00189
00190     LayoutMode m_layoutMode;
00191     QList<HostInfo> m_currentHosts;  // 缓存当前主机列表，用于布局切换
00192 };
00193
00194 #endif // NETWORKTOPOLOGY_H
```

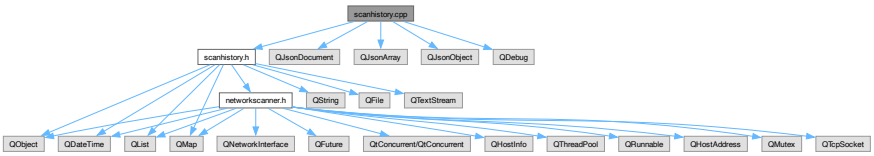
9.64 README.dox 文件参考

9.65 README.md 文件参考

9.66 scanhistory.cpp 文件参考

```
#include "scanhistory.h"
#include <QJsonDocument>
#include <QJsonArray>
#include <QJsonObject>
#include <QDebug>
```

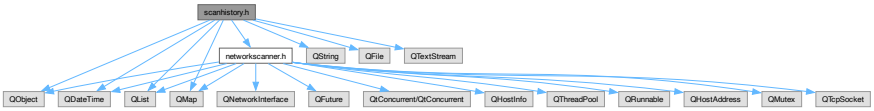
scanhistory.cpp 的引用(Include)关系图:



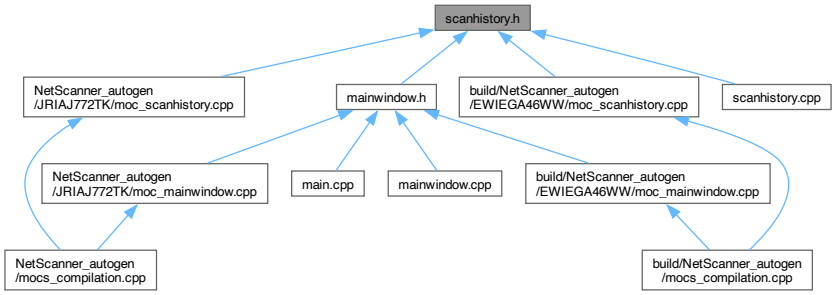
9.67 scanhistory.h 文件参考

```
#include <QObject>
#include <QDateTime>
#include <QList>
#include <QMap>
#include <QString>
#include <QFile>
#include <QTextStream>
#include "networkscanner.h"
```

scanhistory.h 的引用(Include)关系图:



此图展示该文件被哪些文件直接或间接地引用了:



类

- struct ScanSession
- class ScanHistory

9.68 scanhistory.h

浏览该文件的文档.

```
00001 #ifndef SCANHISTORY_H
00002 #define SCANHISTORY_H
00003
00004 #include <QObject>
00005 #include <QDateTime>
00006 #include <QList>
00007 #include <QMap>
00008 #include <QString>
00009 #include <QFile>
00010 #include <QTextStream>
00011 #include "networkscanner.h"
00012
00013 // 扫描会话数据结构
00014 struct ScanSession {
00015     QDateTime scanTime;
00016     QString description;
00017     QList<HostInfo> hosts;
00018
00019     // 会话统计信息
00020     int totalHosts() const { return hosts.size(); }
00021     int reachableHosts() const;
00022     int unreachableHosts() const { return totalHosts() - reachableHosts(); }
00023     QMap<int, int> portDistribution() const;
00024 };
00025
00026 // 扫描历史管理类
00027 class ScanHistory : public QObject
00028 {
00029     Q_OBJECT
00030
00031 public:
00032     explicit ScanHistory(QObject *parent = nullptr);
00033
00034     // 添加扫描会话
00035     void addSession(const QList<HostInfo> &hosts, const QString &description = QString());
00036
00037     // 获取历史扫描会话
00038     QList<ScanSession> getSessions() const { return m_sessions; }
00039
00040     // 获取特定会话
00041     ScanSession getSession(int index) const;
00042
00043     // 获取会话数量
00044     int sessionCount() const { return m_sessions.size(); }
00045
```

```
00046     // 清除所有历史记录
00047     void clearHistory();
00048
00049     // 删除特定会话
00050     void removeSession(int index);
00051
00052     // 将两次扫描结果进行比较，找出新增和消失的主机
00053     QPair<QList<HostInfo>, QList<HostInfo>> compareScans(int sessionIndex1, int sessionIndex2) const;
00054
00055     // 保存和加载扫描历史
00056     bool saveToFile(const QString &filename) const;
00057     bool loadFromFile(const QString &filename);
00058
00059 signals:
00060     void historyChanged();
00061
00062 private:
00063     QList<ScanSession> m_sessions;
00064 };
00065
00066 #endif // SCANHISTORY_H
```

Index

_LP64
moc_predefs.h, 229, 299
__AARCH64EL__
moc_predefs.h, 178, 249
__AARCH64_CMODEL_SMALL__
moc_predefs.h, 178, 248
__AARCH64_SIMD__
moc_predefs.h, 178, 248
__APPLE_CC__
moc_predefs.h, 179, 249
__APPLE__
moc_predefs.h, 178, 249
__ARM64_ARCH_8__
moc_predefs.h, 179, 249
__ARM_64BIT_STATE
moc_predefs.h, 179, 249
__ARM_ACLE
moc_predefs.h, 179, 249
__ARM_ALIGN_MAX_STACK_PWR
moc_predefs.h, 179, 249
__ARM_ARCH
moc_predefs.h, 179, 250
__ARM_ARCH_8_3__
moc_predefs.h, 179, 250
__ARM_ARCH_8_4__
moc_predefs.h, 180, 250
__ARM_ARCH_8_5__
moc_predefs.h, 180, 250
__ARM_ARCH_ISA_A64
moc_predefs.h, 180, 250
__ARM_ARCH_PROFILE
moc_predefs.h, 180, 250
__ARM_FEATURE_AES
moc_predefs.h, 180, 250
__ARM_FEATURE_ATOMICS
moc_predefs.h, 180, 250
__ARM_FEATURE_BTI
moc_predefs.h, 180, 250
__ARM_FEATURE_CLZ
moc_predefs.h, 180, 250
__ARM_FEATURE_COMPLEX
moc_predefs.h, 180, 251
__ARM_FEATURE_CRC32
moc_predefs.h, 180, 251
__ARM_FEATURE_CRYPTO
moc_predefs.h, 181, 251
__ARM_FEATURE_DIRECTED_ROUNDING
moc_predefs.h, 181, 251
__ARM_FEATURE_DIV
moc_predefs.h, 181, 251
__ARM_FEATURE_DOTPROD
moc_predefs.h, 181, 251
__ARM_FEATURE_FMA
moc_predefs.h, 181, 251
__ARM_FEATURE_FP16_FML
moc_predefs.h, 181, 251
__ARM_FEATURE_FP16_SCALAR_ARITHMETIC
moc_predefs.h, 181, 251
__ARM_FEATURE_FP16_VECTOR_ARITHMETIC
moc_predefs.h, 181, 251
__ARM_FEATURE_FRINT
moc_predefs.h, 181, 252
__ARM_FEATURE_IDIV
moc_predefs.h, 181, 252
__ARM_FEATURE_JCVT
moc_predefs.h, 182, 252
__ARM_FEATURE_LDREX
moc_predefs.h, 182, 252
__ARM_FEATURE_NUMERIC_MAXMIN
moc_predefs.h, 182, 252
__ARM_FEATURE_PAUTH
moc_predefs.h, 182, 252
__ARM_FEATURE_QRDMX
moc_predefs.h, 182, 252
__ARM_FEATURE_RCPC
moc_predefs.h, 182, 252
__ARM_FEATURE_SHA2
moc_predefs.h, 182, 252
__ARM_FEATURE_SHA3
moc_predefs.h, 182, 252
__ARM_FEATURE_SHA512
moc_predefs.h, 182, 253
__ARM_FEATURE_UNALIGNED
moc_predefs.h, 182, 253
__ARM_FP
moc_predefs.h, 183, 253
__ARM_FP16_ARGS
moc_predefs.h, 183, 253
__ARM_FP16_FORMAT_IEEE
moc_predefs.h, 183, 253
__ARM_NEON
moc_predefs.h, 183, 253
__ARM_NEON_FP
moc_predefs.h, 183, 253
__ARM_NEON__
moc_predefs.h, 183, 253
__ARM_PCS_AAPCS64
moc_predefs.h, 183, 253

__ARM_SIZEOF_MINIMAL_ENUM
moc_predefs.h, 183, 253
__ARM_SIZEOF_WCHAR_T
moc_predefs.h, 183, 254
__ARM_STATE_ZA
moc_predefs.h, 183, 254
__ARM_STATE_ZT0
moc_predefs.h, 184, 254
__ATOMIC_ACQUIRE
moc_predefs.h, 184, 254
__ATOMIC_ACQ_REL
moc_predefs.h, 184, 254
__ATOMIC_CONSUME
moc_predefs.h, 184, 254
__ATOMIC_RELAXED
moc_predefs.h, 184, 254
__ATOMIC_RELEASE
moc_predefs.h, 184, 254
__ATOMIC_SEQ_CST
moc_predefs.h, 184, 254
__BIGGEST_ALIGNMENT__
moc_predefs.h, 184, 254
__BITINT_MAXWIDTH__
moc_predefs.h, 184, 255
__BLOCKS__
moc_predefs.h, 185, 255
__BOOL_WIDTH__
moc_predefs.h, 185, 255
__BYTE_ORDER__
moc_predefs.h, 185, 255
__CHAR16_TYPE__
moc_predefs.h, 185, 255
__CHAR32_TYPE__
moc_predefs.h, 185, 255
__CHAR_BIT__
moc_predefs.h, 185, 255
__CLANG_ATOMIC_BOOL_LOCK_FREE
moc_predefs.h, 185, 255
__CLANG_ATOMIC_CHAR16_T_LOCK_FREE
moc_predefs.h, 185, 256
__CLANG_ATOMIC_CHAR32_T_LOCK_FREE
moc_predefs.h, 185, 256
__CLANG_ATOMIC_CHAR_LOCK_FREE
moc_predefs.h, 186, 256
__CLANG_ATOMIC_INT_LOCK_FREE
moc_predefs.h, 186, 256
__CLANG_ATOMIC_LLONG_LOCK_FREE
moc_predefs.h, 186, 256
__CLANG_ATOMIC_LONG_LOCK_FREE
moc_predefs.h, 186, 256
__CLANG_ATOMIC_POINTER_LOCK_FREE
moc_predefs.h, 186, 256
__CLANG_ATOMIC_SHORT_LOCK_FREE
moc_predefs.h, 186, 256
__CLANG_ATOMIC_WCHAR_T_LOCK_FREE
moc_predefs.h, 186, 256
__CONSTANT_CFSTRINGS__
moc_predefs.h, 187, 257
__DBL_DECIMAL_DIG__
moc_predefs.h, 193, 263
__DBL_DENORM_MIN__
moc_predefs.h, 193, 263
__DBL_DIG__
moc_predefs.h, 193, 263
__DBL_EPSILON__
moc_predefs.h, 193, 263
__DBL_HAS_DENORM__
moc_predefs.h, 193, 263
__DBL_HAS_INFINITY__
moc_predefs.h, 193, 263
__DBL_HAS_QUIET_NAN__
moc_predefs.h, 193, 263
__DBL_MANT_DIG__
moc_predefs.h, 193, 263
__DBL_MAX_10_EXP__
moc_predefs.h, 193, 264
__DBL_MAX_EXP__
moc_predefs.h, 194, 264
__DBL_MAX__
moc_predefs.h, 193, 264
__DBL_MIN_10_EXP__
moc_predefs.h, 194, 264
__DBL_MIN_EXP__
moc_predefs.h, 194, 264
__DBL_MIN__
moc_predefs.h, 194, 264
__DBL_NORM_MAX__
moc_predefs.h, 194, 264
__DECIMAL_DIG__
moc_predefs.h, 194, 264
__DEPRECATED
moc_predefs.h, 194, 264
__DYNAMIC__
moc_predefs.h, 194, 264
__ENVIRONMENT_MAC_OS_X_VERSION_MIN_REQUIRED__
moc_predefs.h, 194, 265
__ENVIRONMENT_OS_VERSION_MIN_REQUIRED__
moc_predefs.h, 194, 265
__EXCEPTIONS
moc_predefs.h, 195, 265
__FINITE_MATH_ONLY__
moc_predefs.h, 195, 265
__FLT16_DECIMAL_DIG__
moc_predefs.h, 195, 265
__FLT16_DENORM_MIN__
moc_predefs.h, 195, 265
__FLT16_DIG__
moc_predefs.h, 195, 265
__FLT16_EPSILON__
moc_predefs.h, 195, 265
__FLT16_HAS_DENORM__
moc_predefs.h, 195, 265
__FLT16_HAS_INFINITY__
moc_predefs.h, 195, 265
__FLT16_HAS_QUIET_NAN__
moc_predefs.h, 195, 266

__FLT16_MANT_DIG__
moc_predefs.h, 195, 266
__FLT16_MAX_10_EXP__
moc_predefs.h, 196, 266
__FLT16_MAX_EXP__
moc_predefs.h, 196, 266
__FLT16_MAX__
moc_predefs.h, 196, 266
__FLT16_MIN_10_EXP__
moc_predefs.h, 196, 266
__FLT16_MIN_EXP__
moc_predefs.h, 196, 266
__FLT16_MIN__
moc_predefs.h, 196, 266
__FLT16_NORM_MAX__
moc_predefs.h, 196, 266
__FLT_DECIMAL_DIG__
moc_predefs.h, 196, 266
__FLT_DENORM_MIN__
moc_predefs.h, 196, 267
__FLT_DIG__
moc_predefs.h, 196, 267
__FLT_EPSILON__
moc_predefs.h, 197, 267
__FLT_HAS_DENORM__
moc_predefs.h, 197, 267
__FLT_HAS_INFINITY__
moc_predefs.h, 197, 267
__FLT_HAS_QUIET_NAN__
moc_predefs.h, 197, 267
__FLT_MANT_DIG__
moc_predefs.h, 197, 267
__FLT_MAX_10_EXP__
moc_predefs.h, 197, 267
__FLT_MAX_EXP__
moc_predefs.h, 197, 267
__FLT_MAX__
moc_predefs.h, 197, 267
__FLT_MIN_10_EXP__
moc_predefs.h, 197, 268
__FLT_MIN_EXP__
moc_predefs.h, 198, 268
__FLT_MIN__
moc_predefs.h, 197, 268
__FLT_NORM_MAX__
moc_predefs.h, 198, 268
__FLT_RADIX__
moc_predefs.h, 198, 268
__FPCLASS_NEGINF
moc_predefs.h, 198, 268
__FPCLASS_NEGNORMAL
moc_predefs.h, 198, 268
__FPCLASS_NEGSUBNORMAL
moc_predefs.h, 198, 268
__FPCLASS_NEGZERO
moc_predefs.h, 198, 269
__FPCLASS_POSINF
moc_predefs.h, 198, 269
__FPCLASS_POSNORMAL
moc_predefs.h, 199, 269
__FPCLASS_POSSUBNORMAL
moc_predefs.h, 199, 269
__FPCLASS_POSZERO
moc_predefs.h, 199, 269
__FPCLASS_QNAN
moc_predefs.h, 199, 269
__FPCLASS_SNAN
moc_predefs.h, 199, 269
__FP_FAST_FMA
moc_predefs.h, 198, 268
__FP_FAST_FMAF
moc_predefs.h, 198, 268
__GCC_ASM_FLAG_OUTPUTS__
moc_predefs.h, 199, 269
__GCC_ATOMIC_BOOL_LOCK_FREE
moc_predefs.h, 199, 269
__GCC_ATOMIC_CHAR16_T_LOCK_FREE
moc_predefs.h, 199, 269
__GCC_ATOMIC_CHAR32_T_LOCK_FREE
moc_predefs.h, 199, 270
__GCC_ATOMIC_CHAR_LOCK_FREE
moc_predefs.h, 199, 270
__GCC_ATOMIC_INT_LOCK_FREE
moc_predefs.h, 200, 270
__GCC_ATOMIC_LLONG_LOCK_FREE
moc_predefs.h, 200, 270
__GCC_ATOMIC_LONG_LOCK_FREE
moc_predefs.h, 200, 270
__GCC_ATOMIC_POINTER_LOCK_FREE
moc_predefs.h, 200, 270
__GCC_ATOMIC_SHORT_LOCK_FREE
moc_predefs.h, 200, 270
__GCC_ATOMIC_TEST_AND_SET_TRUEVAL
moc_predefs.h, 200, 270
__GCC_ATOMIC_WCHAR_T_LOCK_FREE
moc_predefs.h, 200, 270
__GCC_CONSTRUCTIVE_SIZE
moc_predefs.h, 200, 270
__GCC_DESTRUCTIVE_SIZE
moc_predefs.h, 200, 271
__GCC_HAVE_DWARF2_CFI_ASM
moc_predefs.h, 200, 271
__GCC_HAVE_SYNC_COMPARE_AND_SWAP_1
moc_predefs.h, 201, 271
__GCC_HAVE_SYNC_COMPARE_AND_SWAP_16
moc_predefs.h, 201, 271
__GCC_HAVE_SYNC_COMPARE_AND_SWAP_2
moc_predefs.h, 201, 271
__GCC_HAVE_SYNC_COMPARE_AND_SWAP_4
moc_predefs.h, 201, 271
__GCC_HAVE_SYNC_COMPARE_AND_SWAP_8
moc_predefs.h, 201, 271
__GLIBCXX_BITSIZE_INT_N_0
moc_predefs.h, 201, 271
__GLIBCXX_TYPE_INT_N_0
moc_predefs.h, 201, 271

__GNUC_GNU_INLINE__
moc_predefs.h, 201, 272
__GNUC_MINOR__
moc_predefs.h, 201, 272
__GNUC_PATCHLEVEL__
moc_predefs.h, 202, 272
__GNUC__
moc_predefs.h, 201, 271
__GNUG__
moc_predefs.h, 202, 272
__GXX_ABI_VERSION
moc_predefs.h, 202, 272
__GXX_EXPERIMENTAL_CXX0X__
moc_predefs.h, 202, 272
__GXX_RTTI
moc_predefs.h, 202, 272
__GXX_WEAK__
moc_predefs.h, 202, 272
__HAVE_FUNCTION_MULTI_VERSIONING
moc_predefs.h, 202, 272
__INT16_C_SUFFIX__
moc_predefs.h, 202, 272
__INT16_FMTd__
moc_predefs.h, 202, 273
__INT16_FMTi__
moc_predefs.h, 202, 273
__INT16_MAX__
moc_predefs.h, 203, 273
__INT16_TYPE__
moc_predefs.h, 203, 273
__INT32_C_SUFFIX__
moc_predefs.h, 203, 273
__INT32_FMTd__
moc_predefs.h, 203, 273
__INT32_FMTi__
moc_predefs.h, 203, 273
__INT32_MAX__
moc_predefs.h, 203, 273
__INT32_TYPE__
moc_predefs.h, 203, 273
__INT64_C_SUFFIX__
moc_predefs.h, 203, 273
__INT64_FMTd__
moc_predefs.h, 203, 274
__INT64_FMTi__
moc_predefs.h, 203, 274
__INT64_MAX__
moc_predefs.h, 204, 274
__INT64_TYPE__
moc_predefs.h, 204, 274
__INT8_C_SUFFIX__
moc_predefs.h, 204, 274
__INT8_FMTd__
moc_predefs.h, 204, 274
__INT8_FMTi__
moc_predefs.h, 204, 274
__INT8_MAX__
moc_predefs.h, 204, 274
__INT8_TYPE__
moc_predefs.h, 204, 274
__INTMAX_C_SUFFIX__
moc_predefs.h, 208, 279
__INTMAX_FMTd__
moc_predefs.h, 209, 279
__INTMAX_FMTi__
moc_predefs.h, 209, 279
__INTMAX_MAX__
moc_predefs.h, 209, 279
__INTMAX_TYPE__
moc_predefs.h, 209, 279
__INTMAX_WIDTH__
moc_predefs.h, 209, 279
__INTPTR_FMTd__
moc_predefs.h, 209, 279
__INTPTR_FMTi__
moc_predefs.h, 209, 279
__INTPTR_MAX__
moc_predefs.h, 209, 279
__INTPTR_TYPE__
moc_predefs.h, 209, 280
__INTPTR_WIDTH__
moc_predefs.h, 209, 280
__INT_FAST16_FMTd__
moc_predefs.h, 204, 274
__INT_FAST16_FMTi__
moc_predefs.h, 204, 275
__INT_FAST16_MAX__
moc_predefs.h, 204, 275
__INT_FAST16_TYPE__
moc_predefs.h, 205, 275
__INT_FAST16_WIDTH__
moc_predefs.h, 205, 275
__INT_FAST32_FMTd__
moc_predefs.h, 205, 275
__INT_FAST32_FMTi__
moc_predefs.h, 205, 275
__INT_FAST32_MAX__
moc_predefs.h, 205, 275
__INT_FAST32_TYPE__
moc_predefs.h, 205, 275
__INT_FAST32_WIDTH__
moc_predefs.h, 205, 275
__INT_FAST64_FMTd__
moc_predefs.h, 205, 275
__INT_FAST64_FMTi__
moc_predefs.h, 205, 276
__INT_FAST64_MAX__
moc_predefs.h, 205, 276
__INT_FAST64_TYPE__
moc_predefs.h, 206, 276
__INT_FAST64_WIDTH__
moc_predefs.h, 206, 276
__INT_FAST8_FMTd__
moc_predefs.h, 206, 276
__INT_FAST8_FMTi__
moc_predefs.h, 206, 276

__INT_FAST8_MAX__
moc_predefs.h, 206, 276
__INT_FAST8_TYPE__
moc_predefs.h, 206, 276
__INT_FAST8_WIDTH__
moc_predefs.h, 206, 276
__INT_LEAST16_FMTd__
moc_predefs.h, 206, 276
__INT_LEAST16_FMTi__
moc_predefs.h, 206, 277
__INT_LEAST16_MAX__
moc_predefs.h, 206, 277
__INT_LEAST16_TYPE__
moc_predefs.h, 207, 277
__INT_LEAST16_WIDTH__
moc_predefs.h, 207, 277
__INT_LEAST32_FMTd__
moc_predefs.h, 207, 277
__INT_LEAST32_FMTi__
moc_predefs.h, 207, 277
__INT_LEAST32_MAX__
moc_predefs.h, 207, 277
__INT_LEAST32_TYPE__
moc_predefs.h, 207, 277
__INT_LEAST32_WIDTH__
moc_predefs.h, 207, 277
__INT_LEAST64_FMTd__
moc_predefs.h, 207, 277
__INT_LEAST64_FMTi__
moc_predefs.h, 207, 278
__INT_LEAST64_MAX__
moc_predefs.h, 207, 278
__INT_LEAST64_TYPE__
moc_predefs.h, 208, 278
__INT_LEAST64_WIDTH__
moc_predefs.h, 208, 278
__INT_LEAST8_FMTd__
moc_predefs.h, 208, 278
__INT_LEAST8_FMTi__
moc_predefs.h, 208, 278
__INT_LEAST8_MAX__
moc_predefs.h, 208, 278
__INT_LEAST8_TYPE__
moc_predefs.h, 208, 278
__INT_LEAST8_WIDTH__
moc_predefs.h, 208, 278
__INT_MAX__
moc_predefs.h, 208, 278
__INT_WIDTH__
moc_predefs.h, 208, 279
__LDBL_DECIMAL_DIG__
moc_predefs.h, 210, 280
__LDBL_DENORM_MIN__
moc_predefs.h, 210, 280
__LDBL_DIG__
moc_predefs.h, 210, 280
__LDBL_EPSILON__
moc_predefs.h, 210, 280
__LDBL_HAS_DENORM__
moc_predefs.h, 210, 280
__LDBL_HAS_INFINITY__
moc_predefs.h, 210, 280
__LDBL_HAS_QUIET_NAN__
moc_predefs.h, 210, 280
__LDBL_MANT_DIG__
moc_predefs.h, 210, 280
__LDBL_MAX_10_EXP__
moc_predefs.h, 210, 281
__LDBL_MAX_EXP__
moc_predefs.h, 211, 281
__LDBL_MAX__
moc_predefs.h, 210, 281
__LDBL_MIN_10_EXP__
moc_predefs.h, 211, 281
__LDBL_MIN_EXP__
moc_predefs.h, 211, 281
__LDBL_MIN__
moc_predefs.h, 211, 281
__LDBL_NORM_MAX__
moc_predefs.h, 211, 281
__LITTLE_ENDIAN__
moc_predefs.h, 211, 281
__LLONG_WIDTH__
moc_predefs.h, 211, 281
__LONG_LONG_MAX__
moc_predefs.h, 211, 282
__LONG_MAX__
moc_predefs.h, 211, 282
__LONG_WIDTH__
moc_predefs.h, 212, 282
__LP64__
moc_predefs.h, 212, 282
__MACH__
moc_predefs.h, 212, 282
__MEMORY_SCOPE_DEVICE
moc_predefs.h, 212, 282
__MEMORY_SCOPE_SINGLE
moc_predefs.h, 212, 282
__MEMORY_SCOPE_SYSTEM
moc_predefs.h, 212, 282
__MEMORY_SCOPE_WRKGRP
moc_predefs.h, 212, 282
__MEMORY_SCOPE_WVFRNT
moc_predefs.h, 212, 282
__NO_INLINE__
moc_predefs.h, 212, 283
__NO_MATH_ERRNO__
moc_predefs.h, 212, 283
__OBJC_BOOL_IS_BOOL
moc_predefs.h, 213, 283
__OPENCL_MEMORY_SCOPE_ALL_SVM_DEVICES
moc_predefs.h, 213, 283
__OPENCL_MEMORY_SCOPE_DEVICE
moc_predefs.h, 213, 283
__OPENCL_MEMORY_SCOPE_SUB_GROUP
moc_predefs.h, 213, 283

__OPENCL_MEMORY_SCOPE_WORK_GROUP
moc_predefs.h, 213, 283
__OPENCL_MEMORY_SCOPE_WORK_ITEM
moc_predefs.h, 213, 284
__ORDER_BIG_ENDIAN__
moc_predefs.h, 213, 284
__ORDER_LITTLE_ENDIAN__
moc_predefs.h, 214, 284
__ORDER_PDP_ENDIAN__
moc_predefs.h, 214, 284
__PIC__
moc_predefs.h, 214, 284
__POINTER_WIDTH__
moc_predefs.h, 214, 284
__PRAGMA_REDEFINE_EXTNAME
moc_predefs.h, 214, 284
__PTRDIFF_FMTd__
moc_predefs.h, 214, 284
__PTRDIFF_FMTi__
moc_predefs.h, 214, 285
__PTRDIFF_MAX__
moc_predefs.h, 214, 285
__PTRDIFF_TYPE__
moc_predefs.h, 215, 285
__PTRDIFF_WIDTH__
moc_predefs.h, 215, 285
__REGISTER_PREFIX__
moc_predefs.h, 215, 285
__SCHAR_MAX__
moc_predefs.h, 215, 285
__SHRT_MAX__
moc_predefs.h, 215, 285
__SHRT_WIDTH__
moc_predefs.h, 215, 285
__SIG_ATOMIC_MAX__
moc_predefs.h, 215, 285
__SIG_ATOMIC_WIDTH__
moc_predefs.h, 215, 285
__SIZEOF_DOUBLE__
moc_predefs.h, 216, 286
__SIZEOF_FLOAT__
moc_predefs.h, 216, 286
__SIZEOF_INT128__
moc_predefs.h, 216, 286
__SIZEOF_INT__
moc_predefs.h, 216, 287
__SIZEOF_LONG_DOUBLE__
moc_predefs.h, 217, 287
__SIZEOF_LONG_LONG__
moc_predefs.h, 217, 287
__SIZEOF_LONG__
moc_predefs.h, 216, 287
__SIZEOF_POINTER__
moc_predefs.h, 217, 287
__SIZEOF_PTRDIFF_T__
moc_predefs.h, 217, 287
__SIZEOF_SHORT__
moc_predefs.h, 217, 287
__SIZEOF_SIZE_T__
moc_predefs.h, 217, 287
__SIZEOF_WCHAR_T__
moc_predefs.h, 217, 287
__SIZEOF_WINT_T__
moc_predefs.h, 217, 287
__SIZE_FMTX__
moc_predefs.h, 216, 286
__SIZE_FMTo__
moc_predefs.h, 215, 286
__SIZE_FMTu__
moc_predefs.h, 215, 286
__SIZE_FMTx__
moc_predefs.h, 216, 286
__SIZE_MAX__
moc_predefs.h, 216, 286
__SIZE_TYPE__
moc_predefs.h, 216, 286
__SIZE_WIDTH__
moc_predefs.h, 216, 286
__SSP__
moc_predefs.h, 217, 288
__STDCPP_DEFAULT_NEW_ALIGNMENT__
moc_predefs.h, 218, 288
__STDCPP_THREADS__
moc_predefs.h, 218, 289
__STDC_EMBED_EMPTY__
moc_predefs.h, 218, 288
__STDC_EMBED_FOUND__
moc_predefs.h, 218, 288
__STDC_EMBED_NOT_FOUND__
moc_predefs.h, 218, 288
__STDC_HOSTED__
moc_predefs.h, 218, 288
__STDC_NO_THREADS__
moc_predefs.h, 218, 288
__STDC_UTF_16__
moc_predefs.h, 218, 288
__STDC_UTF_32__
moc_predefs.h, 218, 288
__STDC__
moc_predefs.h, 217, 288
__UINT16_C_SUFFIX__
moc_predefs.h, 219, 289
__UINT16_FMTX__
moc_predefs.h, 219, 289
__UINT16_FMTo__
moc_predefs.h, 219, 289
__UINT16_FMTu__
moc_predefs.h, 219, 289
__UINT16_FMTx__
moc_predefs.h, 219, 289
__UINT16_MAX__
moc_predefs.h, 219, 289
__UINT16_TYPE__
moc_predefs.h, 219, 289
__UINT32_C_SUFFIX__
moc_predefs.h, 219, 289

__UINT32_FMTX__
moc_predefs.h, 220, 290
__UINT32_FMTo__
moc_predefs.h, 219, 290
__UINT32_FMTu__
moc_predefs.h, 219, 290
__UINT32_FMTx__
moc_predefs.h, 220, 290
__UINT32_MAX__
moc_predefs.h, 220, 290
__UINT32_TYPE__
moc_predefs.h, 220, 290
__UINT64_C_SUFFIX__
moc_predefs.h, 220, 290
__UINT64_FMTX__
moc_predefs.h, 220, 290
__UINT64_FMTo__
moc_predefs.h, 220, 290
__UINT64_FMTu__
moc_predefs.h, 220, 290
__UINT64_FMTx__
moc_predefs.h, 220, 291
__UINT64_MAX__
moc_predefs.h, 220, 291
__UINT64_TYPE__
moc_predefs.h, 221, 291
__UINT8_C_SUFFIX__
moc_predefs.h, 221, 291
__UINT8_FMTX__
moc_predefs.h, 221, 291
__UINT8_FMTo__
moc_predefs.h, 221, 291
__UINT8_FMTu__
moc_predefs.h, 221, 291
__UINT8_FMTx__
moc_predefs.h, 221, 291
__UINT8_MAX__
moc_predefs.h, 221, 291
__UINT8_TYPE__
moc_predefs.h, 221, 291
__UINTMAX_C_SUFFIX__
moc_predefs.h, 226, 296
__UINTMAX_FMTX__
moc_predefs.h, 226, 297
__UINTMAX_FMTo__
moc_predefs.h, 226, 296
__UINTMAX_FMTu__
moc_predefs.h, 226, 297
__UINTMAX_FMTx__
moc_predefs.h, 227, 297
__UINTMAX_MAX__
moc_predefs.h, 227, 297
__UINTMAX_TYPE__
moc_predefs.h, 227, 297
__UINTMAX_WIDTH__
moc_predefs.h, 227, 297
__UINTPTR_FMTX__
moc_predefs.h, 227, 297
__UINTPTR_FMTo__
moc_predefs.h, 227, 297
__UINTPTR_FMTu__
moc_predefs.h, 227, 297
__UINTPTR_FMTx__
moc_predefs.h, 227, 297
__UINTPTR_MAX__
moc_predefs.h, 227, 298
__UINTPTR_TYPE__
moc_predefs.h, 227, 298
__UINTPTR_WIDTH__
moc_predefs.h, 228, 298
__UINT_FAST16_FMTX__
moc_predefs.h, 222, 292
__UINT_FAST16_FMTo__
moc_predefs.h, 221, 292
__UINT_FAST16_FMTu__
moc_predefs.h, 221, 292
__UINT_FAST16_FMTx__
moc_predefs.h, 222, 292
__UINT_FAST16_MAX__
moc_predefs.h, 222, 292
__UINT_FAST16_TYPE__
moc_predefs.h, 222, 292
__UINT_FAST32_FMTX__
moc_predefs.h, 222, 292
__UINT_FAST32_FMTo__
moc_predefs.h, 222, 292
__UINT_FAST32_FMTu__
moc_predefs.h, 222, 292
__UINT_FAST32_FMTx__
moc_predefs.h, 222, 292
__UINT_FAST32_MAX__
moc_predefs.h, 222, 293
__UINT_FAST32_TYPE__
moc_predefs.h, 222, 293
__UINT_FAST64_FMTX__
moc_predefs.h, 223, 293
__UINT_FAST64_FMTo__
moc_predefs.h, 223, 293
__UINT_FAST64_FMTu__
moc_predefs.h, 223, 293
__UINT_FAST64_FMTx__
moc_predefs.h, 223, 293
__UINT_FAST64_MAX__
moc_predefs.h, 223, 293
__UINT_FAST64_TYPE__
moc_predefs.h, 223, 293
__UINT_FAST8_FMTX__
moc_predefs.h, 223, 294
__UINT_FAST8_FMTo__
moc_predefs.h, 223, 293
__UINT_FAST8_FMTu__
moc_predefs.h, 223, 293
__UINT_FAST8_FMTx__
moc_predefs.h, 223, 294
__UINT_FAST8_MAX__
moc_predefs.h, 224, 294

__UINT_FAST8_TYPE__
moc_predefs.h, 224, 294
__UINT_LEAST16_FMTX__
moc_predefs.h, 224, 294
__UINT_LEAST16_FMTo__
moc_predefs.h, 224, 294
__UINT_LEAST16_FMTu__
moc_predefs.h, 224, 294
__UINT_LEAST16_FMTx__
moc_predefs.h, 224, 294
__UINT_LEAST16_MAX__
moc_predefs.h, 224, 294
__UINT_LEAST16_TYPE__
moc_predefs.h, 224, 294
__UINT_LEAST32_FMTX__
moc_predefs.h, 225, 295
__UINT_LEAST32_FMTo__
moc_predefs.h, 224, 295
__UINT_LEAST32_FMTu__
moc_predefs.h, 224, 295
__UINT_LEAST32_FMTx__
moc_predefs.h, 225, 295
__UINT_LEAST32_MAX__
moc_predefs.h, 225, 295
__UINT_LEAST32_TYPE__
moc_predefs.h, 225, 295
__UINT_LEAST64_FMTX__
moc_predefs.h, 225, 295
__UINT_LEAST64_FMTo__
moc_predefs.h, 225, 295
__UINT_LEAST64_FMTu__
moc_predefs.h, 225, 295
__UINT_LEAST64_FMTx__
moc_predefs.h, 225, 295
__UINT_LEAST64_MAX__
moc_predefs.h, 225, 296
__UINT_LEAST64_TYPE__
moc_predefs.h, 225, 296
__UINT_LEAST8_FMTX__
moc_predefs.h, 226, 296
__UINT_LEAST8_FMTo__
moc_predefs.h, 226, 296
__UINT_LEAST8_FMTu__
moc_predefs.h, 226, 296
__UINT_LEAST8_FMTx__
moc_predefs.h, 226, 296
__UINT_LEAST8_MAX__
moc_predefs.h, 226, 296
__UINT_LEAST8_TYPE__
moc_predefs.h, 226, 296
__USER_LABEL_PREFIX__
moc_predefs.h, 228, 298
__VERSION__
moc_predefs.h, 228, 298
__WCHAR_MAX__
moc_predefs.h, 228, 298
__WCHAR_TYPE__
moc_predefs.h, 228, 298
__WCHAR_WIDTH__
moc_predefs.h, 228, 298
__WINT_MAX__
moc_predefs.h, 228, 299
__WINT_TYPE__
moc_predefs.h, 228, 299
__WINT_WIDTH__
moc_predefs.h, 229, 299
__aarch64__
moc_predefs.h, 178, 248
__apple_build_version__
moc_predefs.h, 179, 249
__arm64
moc_predefs.h, 179, 249
__arm64__
moc_predefs.h, 179, 249
__block
moc_predefs.h, 184, 255
__clang__
moc_predefs.h, 185, 255
__clang_literal_encoding__
moc_predefs.h, 186, 256
__clang_major__
moc_predefs.h, 186, 257
__clang_minor__
moc_predefs.h, 186, 257
__clang_patchlevel__
moc_predefs.h, 187, 257
__clang_version__
moc_predefs.h, 187, 257
__clang_wide_literal_encoding__
moc_predefs.h, 187, 257
__cplusplus
moc_predefs.h, 187, 257
__cpp_aggregate_bases
moc_predefs.h, 187, 257
__cpp_aggregate_nsdmi
moc_predefs.h, 187, 257
__cpp_alias_templates
moc_predefs.h, 187, 257
__cpp_aligned_new
moc_predefs.h, 187, 258
__cpp_attributes
moc_predefs.h, 187, 258
__cpp_binary_literals
moc_predefs.h, 188, 258
__cpp_capture_star_this
moc_predefs.h, 188, 258
__cpp_constexpr
moc_predefs.h, 188, 258
__cpp_constexpr_in_decltype
moc_predefs.h, 188, 258
__cpp_decltype
moc_predefs.h, 188, 258
__cpp_decltype_auto
moc_predefs.h, 188, 258
__cpp_deduction_guides
moc_predefs.h, 188, 258

__cpp_delegating_constructors
moc_predefs.h, 188, 258
__cpp_deleted_function
moc_predefs.h, 188, 259
__cpp_digit_separators
moc_predefs.h, 188, 259
__cpp_enumerator_attributes
moc_predefs.h, 189, 259
__cpp_exceptions
moc_predefs.h, 189, 259
__cpp_fold_expressions
moc_predefs.h, 189, 259
__cpp_generic_lambdas
moc_predefs.h, 189, 259
__cpp_guaranteed_copy_elision
moc_predefs.h, 189, 259
__cpp_hex_float
moc_predefs.h, 189, 259
__cpp_if_constexpr
moc_predefs.h, 189, 259
__cpp_impl_destroying_delete
moc_predefs.h, 189, 259
__cpp_inheriting_constructors
moc_predefs.h, 189, 260
__cpp_init_captures
moc_predefs.h, 189, 260
__cpp_initializer_lists
moc_predefs.h, 190, 260
__cpp_inline_variables
moc_predefs.h, 190, 260
__cpp_lambdas
moc_predefs.h, 190, 260
__cpp_named_character_escapes
moc_predefs.h, 190, 260
__cpp_namespace_attributes
moc_predefs.h, 190, 260
__cpp_nested_namespace_definitions
moc_predefs.h, 190, 260
__cpp_noexcept_function_type
moc_predefs.h, 190, 260
__cpp_nontype_template_args
moc_predefs.h, 190, 260
__cpp_nontype_template_parameter_auto
moc_predefs.h, 190, 261
__cpp_nsdmi
moc_predefs.h, 190, 261
__cpp_pack_indexing
moc_predefs.h, 191, 261
__cpp_placeholder_variables
moc_predefs.h, 191, 261
__cpp_range_based_for
moc_predefs.h, 191, 261
__cpp_raw_strings
moc_predefs.h, 191, 261
__cpp_ref_qualifiers
moc_predefs.h, 191, 261
__cpp_return_type_deduction
moc_predefs.h, 191, 261
__cpp_rtti
moc_predefs.h, 191, 261
__cpp_rvalue_references
moc_predefs.h, 191, 261
__cpp_static_assert
moc_predefs.h, 191, 262
__cpp_static_call_operator
moc_predefs.h, 191, 262
__cpp_structured_bindings
moc_predefs.h, 192, 262
__cpp_template_auto
moc_predefs.h, 192, 262
__cpp_template_template_args
moc_predefs.h, 192, 262
__cpp_threadsafe_static_init
moc_predefs.h, 192, 262
__cpp_unicode_characters
moc_predefs.h, 192, 262
__cpp_unicode_literals
moc_predefs.h, 192, 262
__cpp_user_defined_literals
moc_predefs.h, 192, 262
__cpp_variable_templates
moc_predefs.h, 192, 262
__cpp_variadic_templates
moc_predefs.h, 192, 263
__cpp_variadic_using
moc_predefs.h, 192, 263
__has_include
CMakeCXXCompilerId.cpp, 124
__llvm__
moc_predefs.h, 211, 281
__nonnull
moc_predefs.h, 213, 283
__null_unspecified
moc_predefs.h, 213, 283
__nullable
moc_predefs.h, 213, 283
__pic__
moc_predefs.h, 214, 284
__private_extern__
moc_predefs.h, 214, 284
__strong
moc_predefs.h, 218, 289
__unsafe_unretained
moc_predefs.h, 228, 298
__weak
moc_predefs.h, 228, 298
~MainWindow
MainWindow, 45
~NetworkScanner
NetworkScanner, 73

addSession
ScanHistory, 106
analysisCompleted
DeviceAnalyzer, 28
analyzeHosts
DeviceAnalyzer, 28

analyzeSubnets
TopologyAnalyzer, 120
analyzeTTLLayers
TopologyAnalyzer, 120
applySettings
MainWindow, 46
applyTheme
MainWindow, 46
ARCHITECTURE_ID
CMakeCXXCompilerId.cpp, 124
autoLayout
NetworkTopologyView, 97

boundingRect
ConnectionLine, 23
DeviceNode, 35
build/CMakeFiles/3.31.5/CompilerIdCXX/CMakeCXXCompilerId.cpp, 123
build/CMakeFiles/NetScanner.dir/deviceanalyzer.cpp.o.d, 127
build/CMakeFiles/NetScanner.dir/main.cpp.o.d, 127
build/CMakeFiles/NetScanner.dir/mainwindow.cpp.o.d, 127
build/CMakeFiles/NetScanner.dir/NetScanner_autogen/mocs_compilation.cpp.o.d, 127
build/CMakeFiles/NetScanner.dir/networkscanner.cpp.o.d, 127
build/CMakeFiles/NetScanner.dir/networktopology.cpp.o.d, 127
build/CMakeFiles/NetScanner.dir/scanhistory.cpp.o.d, 127
build/NetScanner_autogen/EWIEGA46WW/moc_deviceanalyzer.cpp, 127, 128
build/NetScanner_autogen/EWIEGA46WW/moc_deviceanalyzer.cpp.d, 135
build/NetScanner_autogen/EWIEGA46WW/moc_mainwindow.cpp, 135, 136
build/NetScanner_autogen/EWIEGA46WW/moc_mainwindow.cpp.d, 148
build/NetScanner_autogen/EWIEGA46WW/moc_networkscanner.cpp, 148, 149
build/NetScanner_autogen/EWIEGA46WW/moc_networkscanner.cpp.d, 155
build/NetScanner_autogen/EWIEGA46WW/moc_networktopology.cpp, 155, 156
build/NetScanner_autogen/EWIEGA46WW/moc_networktopology.cpp.d, 163
build/NetScanner_autogen/EWIEGA46WW/moc_scanhistory.cpp, 163, 164
build/NetScanner_autogen/EWIEGA46WW/moc_scanhistory.cpp.d, 168
build/NetScanner_autogen/moc_predefs.h, 168, 232
build/NetScanner_autogen/mocs_compilation.cpp, 308

calculateSubnet
TopologyAnalyzer, 121
clear
DeviceAnalyzer, 28
NetworkTopology, 93
NetworkTopologyView, 97
clearFilters
MainWindow, 46
clearHistory
ScanHistory, 106
clearResults
MainWindow, 47
clusterDevicesByResponseTime
TopologyAnalyzer, 121
CMakeCXXCompilerId.cpp
__has_include, 124
ARCHITECTURE_ID, 124
COMPILER_ID, 124
CXX_STD, 124
CXX_STD_11, 124
CXX_STD_14, 124
CXX_STD_17, 124
CXX_STD_20, 124
CXX_STD_23, 124
CXX_STD_98, 124
DEC, 125
HEX, 125
info_arch, 126
info_compiler, 126
info_language_extensions_default, 126
info_language_standard_default, 126
info_platform, 126
main, 126
PLATFORM_ID, 125
STRINGIFY, 125
STRINGIFY_HELPER, 125
compareScanResults
MainWindow, 47
compareScans
ScanHistory, 106
COMPILER_ID
CMakeCXXCompilerId.cpp, 124
CONNECTION_DIRECT
networktopology.h, 322
CONNECTION_ROUTED
networktopology.h, 322
CONNECTION_UNKNOWN
networktopology.h, 322
CONNECTION_VPN
networktopology.h, 322
CONNECTION_WIRELESS
networktopology.h, 322
ConnectionLine, 21
boundingRect, 23
ConnectionLine, 23
connectionType, 23
m_connectionType, 24
m_source, 24
m_target, 24
paint, 23
setConnectionType, 23
updatePosition, 23
ConnectionType

networktopology.h, 321
connectionType
ConnectionLine, 23
createConnection
NetworkTopologyView, 98
createDetailsTab
MainWindow, 47
createDeviceTypeChart
DeviceAnalyzer, 29
createHistoryTab
MainWindow, 48
createMenus
MainWindow, 48
createPortDistributionChart
DeviceAnalyzer, 29
createSecurityTab
MainWindow, 49
createSettingsDialog
MainWindow, 49
createStatisticsTab
MainWindow, 50
createTopologyTab
MainWindow, 50
createUI
MainWindow, 51
createVendorDistributionChart
DeviceAnalyzer, 29
CXX_STD
CMakeCXXCompilerId.cpp, 124
CXX_STD_11
CMakeCXXCompilerId.cpp, 124
CXX_STD_14
CMakeCXXCompilerId.cpp, 124
CXX_STD_17
CMakeCXXCompilerId.cpp, 124
CXX_STD_20
CMakeCXXCompilerId.cpp, 124
CXX_STD_23
CMakeCXXCompilerId.cpp, 124
CXX_STD_98
CMakeCXXCompilerId.cpp, 124

DEC
CMakeCXXCompilerId.cpp, 125
DEEP_SCAN
ScanStrategy, 112
description
ScanSession, 110
determineDeviceType
DeviceAnalyzer, 30
NetworkTopologyView, 98
DEVICE_IOT
networktopology.h, 322
DEVICE_MOBILE
networktopology.h, 322
DEVICE_PC
networktopology.h, 322
DEVICE_PRINTER
networktopology.h, 322
DEVICE_ROUTER
networktopology.h, 322
DEVICE_SERVER
networktopology.h, 322
DEVICE_UNKNOWN
networktopology.h, 322
DeviceAnalyzer, 24
analysisCompleted, 28
analyzeHosts, 28
clear, 28
createDeviceTypeChart, 29
createPortDistributionChart, 29
createVendorDistributionChart, 29
determineDeviceType, 30
DeviceAnalyzer, 27
generateSecurityReport, 30
getDeviceTypeChart, 30
getPortDistributionChart, 30
getReachableHostsCount, 30
getTotalHostsCount, 30
getUnreachableHostsCount, 31
getVendorDistributionChart, 31
m_deviceTypeChart, 31
m_deviceTypeSeries, 31
m_portDistributionChart, 31
m_portSeries, 31
m_reachableHosts, 31
m_totalHosts, 31
m_vendorDistributionChart, 31
m_vendorSeries, 31
deviceanalyzer.cpp, 309
deviceanalyzer.h, 310
DeviceNode, 32
boundingRect, 35
DeviceNode, 35
deviceType, 35
hostInfo, 35
hoverEnterEvent, 36
hoverLeaveEvent, 36
ipAddress, 36
m_dragStartPosition, 38
m_highlight, 38
m_host, 38
m_networkLayer, 38
m_subnetGroup, 38
m_type, 38
mouseMoveEvent, 36
mousePressEvent, 36
mouseReleaseEvent, 36
networkLayer, 36
paint, 36
setDeviceType, 37
setNetworkLayer, 37
setPosition, 37
setSubnetGroup, 37
subnetGroup, 37
deviceSelected
NetworkTopology, 93

DeviceType
networktopology.h, 322
deviceType
DeviceNode, 35

executeProcess
NetworkScanner, 74
exportToCSV
MainWindow, 51

filterResults
MainWindow, 52

generatePseudoMACFromIP
NetworkScanner, 74
generateSecurityReport
DeviceAnalyzer, 30
MainWindow, 52
getAddressesToScan
NetworkScanner, 74
getDeviceTypeChart
DeviceAnalyzer, 30
getLocalNetworkAddresses
NetworkScanner, 75
getMacAddressFromSystemCalls
NetworkScanner, 75
getMaxParallelTasks
ScanStrategy, 113
getMode
ScanStrategy, 113
getPortDistributionChart
DeviceAnalyzer, 30
getPortsToScan
ScanStrategy, 113
getReachableHostsCount
DeviceAnalyzer, 30
getScannedHosts
NetworkScanner, 75
getScanTimeout
ScanStrategy, 113
getSession
ScanHistory, 107
getSessions
ScanHistory, 107
getTotalHostsCount
DeviceAnalyzer, 30
getTTLValue
TopologyAnalyzer, 121
getUnreachableHostsCount
DeviceAnalyzer, 31
getVendorDistributionChart
DeviceAnalyzer, 31
groupedLayout
NetworkTopologyView, 98

HEX
CMakeCXXCompilerId.cpp, 125
hierarchicalLayout
NetworkTopologyView, 99
historyChanged
ScanHistory, 107
hostFound
NetworkScanner, 76
HostInfo, 39
hostName, 40
ipAddress, 40
isReachable, 40
macAddress, 40
macVendor, 40
openPorts, 40
scanTime, 40
hostInfo
DeviceNode, 35
hostName
HostInfo, 40
hosts
ScanSession, 110
hoverEnterEvent
DeviceNode, 36
hoverLeaveEvent
DeviceNode, 36

inferDeviceConnections
TopologyAnalyzer, 121
info_arch
CMakeCXXCompilerId.cpp, 126
info_compiler
CMakeCXXCompilerId.cpp, 126
info_language_extensions_default
CMakeCXXCompilerId.cpp, 126
info_language_standard_default
CMakeCXXCompilerId.cpp, 126
info_platform
CMakeCXXCompilerId.cpp, 126
inSameSubnet
TopologyAnalyzer, 122
ipAddress
DeviceNode, 36
HostInfo, 40
isHostReachable
NetworkScanner, 76
isReachable
HostInfo, 40
isReachableOnPorts
NetworkScanner, 76
isScanning
NetworkScanner, 77

LAYOUT_AUTO
NetworkTopology, 92
LAYOUT_GROUPED
NetworkTopology, 92
LAYOUT_HIERARCHICAL
NetworkTopology, 92
LayoutMode
NetworkTopology, 92
loadFromFile
ScanHistory, 107

loadHistoryFromFile
MainWindow, 52
loadSettings
MainWindow, 53
lookupHostName
NetworkScanner, 77
lookupMacAddress
NetworkScanner, 77
lookupMacVendor
NetworkScanner, 78

m_aboutAction
MainWindow, 62
m_activeHosts
NetworkScanner, 88
m_address
ScanTask, 117
m_analyzer
NetworkTopologyView, 100
m_centralWidget
MainWindow, 62
m_clearButton
MainWindow, 62
m_clearFilterButton
MainWindow, 62
m_connections
NetworkTopologyView, 100
m_connectionType
ConnectionLine, 24
m_controlLayout
MainWindow, 63
m_controlPanel
NetworkTopology, 94
m_currentHostIndex
MainWindow, 63
m_currentHosts
NetworkTopology, 94
m_customPortsCheckBox
MainWindow, 63
m_customRangeCheckBox
MainWindow, 63
m_darkModeAction
MainWindow, 63
m_darkModeEnabled
MainWindow, 63
m_detailsLayout
MainWindow, 63
m_detailsTab
MainWindow, 63
m_detailsTextEdit
MainWindow, 63
m_deviceAnalyzer
MainWindow, 63
m_deviceTypeChart
DeviceAnalyzer, 31
m_deviceTypeChartView
MainWindow, 64
m_deviceTypeSeries
DeviceAnalyzer, 31
m_dragStartPosition
DeviceNode, 38
m_endIPLineEdit
MainWindow, 64
m_endIPRange
NetworkScanner, 88
m_exitAction
MainWindow, 64
m_exportAction
MainWindow, 64
m_fileMenu
MainWindow, 64
m_filterButton
MainWindow, 64
m_filterIPLineEdit
MainWindow, 64
m_filterTypeComboBox
MainWindow, 64
m_filterVendorComboBox
MainWindow, 64
m_filterWidget
MainWindow, 64
m_helpMenu
MainWindow, 65
m_highlight
DeviceNode, 38
m_historyTab
MainWindow, 65
m_historyTable
MainWindow, 65
m_host
DeviceNode, 38
m_hostResponseTimes
ScanStrategy, 114
m_hostsFound
MainWindow, 65
m_isScanning
NetworkScanner, 88
m_layoutMode
NetworkTopology, 94
m_loadHistoryAction
MainWindow, 65
m_macAddressCache
NetworkScanner, 88
m_mainLayout
MainWindow, 65
m_mode
ScanStrategy, 114
m_mutex
NetworkScanner, 88
m_networkLayer
DeviceNode, 38
m_networkTopology
MainWindow, 65
m_nodes
NetworkTopologyView, 100
m_parent
ScanTask, 117

m_portDistributionChart
DeviceAnalyzer, 31
m_portDistributionChartView
MainWindow, 65
m_ports
ScanTask, 118
m_portSeries
DeviceAnalyzer, 31
m_portsGroupBox
MainWindow, 65
m_portsLineEdit
MainWindow, 65
m_portsToScan
NetworkScanner, 88
m_progressBar
MainWindow, 66
m_rangeGroupBox
MainWindow, 66
m_reachableHosts
DeviceAnalyzer, 31
m_resultsTable
MainWindow, 66
m_saveButton
MainWindow, 66
m_saveHistoryAction
MainWindow, 66
m_saveTopologyAction
MainWindow, 66
m_scanButton
MainWindow, 66
m_scanFutures
NetworkScanner, 88
m_scanHistory
MainWindow, 66
m_scannedHosts
NetworkScanner, 88
m_scannedHostsList
NetworkScanner, 89
m_scanner
MainWindow, 66
m_scanStrategy
NetworkScanner, 89
m_scanTab
MainWindow, 66
m_scanTimeout
NetworkScanner, 89
m_scene
NetworkTopologyView, 100
m_scheduleScanAction
MainWindow, 67
m_securityReportText
MainWindow, 67
m_sessionComboBox
MainWindow, 67
m_sessions
ScanHistory, 108
m_settingsAction
MainWindow, 67
m_settingsLayout
MainWindow, 67
m_settingsTab
MainWindow, 67
m_source
ConnectionLine, 24
m_startIPLineEdit
MainWindow, 67
m_startIPRange
NetworkScanner, 89
m_statisticsTab
MainWindow, 67
m_statusBar
MainWindow, 67
m_statusLabel
MainWindow, 67
m_stopButton
MainWindow, 68
m_subnetGroup
DeviceNode, 38
m_tabWidget
MainWindow, 68
m_target
ConnectionLine, 24
m_threadPool
NetworkScanner, 89
m_timeout
ScanTask, 118
m_timeoutSpinBox
MainWindow, 68
m_toolsMenu
MainWindow, 68
m_topologyTab
MainWindow, 68
m_topologyView
NetworkTopology, 94
m_totalHosts
DeviceAnalyzer, 31
NetworkScanner, 89
m_type
DeviceNode, 38
m_useCustomRange
NetworkScanner, 89
m_vendorChartView
MainWindow, 68
m_vendorDistributionChart
DeviceAnalyzer, 31
m_vendorSeries
DeviceAnalyzer, 31
m_viewMenu
MainWindow, 68
macAddress
HostInfo, 40
macVendor
HostInfo, 40
main
CMakeCXXCompilerId.cpp, 126
main.cpp, 312

main.cpp, 311
    main, 312
MainWindow, 41
    ~MainWindow, 45
    applySettings, 46
    applyTheme, 46
    clearFilters, 46
    clearResults, 47
    compareScanResults, 47
    createDetailsTab, 47
    createHistoryTab, 48
    createMenus, 48
    createSecurityTab, 49
    createSettingsDialog, 49
    createStatisticsTab, 50
    createTopologyTab, 50
    createUI, 51
    exportToCSV, 51
    filterResults, 52
    generateSecurityReport, 52
    loadHistoryFromFile, 52
    loadSettings, 53
    m_aboutAction, 62
    m_centralWidget, 62
    m_clearButton, 62
    m_clearFilterButton, 62
    m_controlLayout, 63
    m_currentHostIndex, 63
    m_customPortsCheckBox, 63
    m_customRangeCheckBox, 63
    m_darkModeAction, 63
    m_darkModeEnabled, 63
    m_detailsLayout, 63
    m_detailsTab, 63
    m_detailsTextEdit, 63
    m_deviceAnalyzer, 63
    m_deviceTypeChartView, 64
    m_endIPLineEdit, 64
    m_exitAction, 64
    m_exportAction, 64
    m_fileMenu, 64
    m_filterButton, 64
    m_filterIPLineEdit, 64
    m_filterTypeComboBox, 64
    m_filterVendorComboBox, 64
    m_filterWidget, 64
    m_helpMenu, 65
    m_historyTab, 65
    m_historyTable, 65
    m_hostsFound, 65
    m_loadHistoryAction, 65
    m_mainLayout, 65
    m_networkTopology, 65
    m_portDistributionChartView, 65
    m_portsGroupBox, 65
    m_portsLineEdit, 65
    m_progressBar, 66
    m_rangeGroupBox, 66
    m_resultsTable, 66
    m_saveButton, 66
    m_saveHistoryAction, 66
    m_saveTopologyAction, 66
    m_scanButton, 66
    m_scanHistory, 66
    m_scanner, 66
    m_scanTab, 66
    m_scheduleScanAction, 67
    m_securityReportText, 67
    m_sessionComboBox, 67
    m_settingsAction, 67
    m_settingsLayout, 67
    m_settingsTab, 67
    m_startIPLineEdit, 67
    m_statisticsTab, 67
    m_statusBar, 67
    m_statusLabel, 67
    m_stopButton, 68
    m_tabWidget, 68
    m_timeoutSpinBox, 68
    m_toolsMenu, 68
    m_topologyTab, 68
    m_vendorChartView, 68
    m_viewMenu, 68
    MainWindow, 45
    onHostFound, 53
    onScanError, 53
    onScanFinished, 54
    onScanProgress, 54
    onScanStarted, 54
    onThemeChanged, 55
    refreshTopology, 55
    saveHistoryToFile, 55
    saveResults, 55
    saveSettings, 56
    saveTopologyImage, 56
    scheduleScan, 56
    setupConnections, 57
    showAbout, 58
    showHistoryView, 59
    showHostDetails, 59
    showSettings, 59
    showStatisticsView, 59
    showTopologyView, 59
    startScan, 60
    stopScan, 60
    toggleDarkMode, 60
    togglePortScanOptions, 61
    toggleRangeOptions, 61
    updateNetworkTopology, 61
    updatePortsList, 62
    updateStatistics, 62
mainwindow.cpp, 312
mainwindow.h, 312
moc_deviceanalyzer.cpp
    Q_CONSTINIT, 128, 131, 133
moc_mainwindow.cpp

Q_CONSTINIT, 136, 140, 144
moc_networkscanner.cpp
Q_CONSTINIT, 149, 152
moc_networktopology.cpp
Q_CONSTINIT, 156, 160
moc_predefs.h
_LP64, 229, 299
__AARCH64EL__, 178, 249
__AARCH64_CMODEL_SMALL__, 178, 248
__AARCH64_SIMD__, 178, 248
__APPLE_CC__, 179, 249
__APPLE__, 178, 249
__ARM64_ARCH_8__, 179, 249
__ARM_64BIT_STATE, 179, 249
__ARM_ACLE, 179, 249
__ARM_ALIGN_MAX_STACK_PWR, 179, 249
__ARM_ARCH, 179, 250
__ARM_ARCH_8_3__, 179, 250
__ARM_ARCH_8_4__, 180, 250
__ARM_ARCH_8_5__, 180, 250
__ARM_ARCH_ISA_A64, 180, 250
__ARM_ARCH_PROFILE, 180, 250
__ARM_FEATURE_AES, 180, 250
__ARM_FEATURE_ATOMICS, 180, 250
__ARM_FEATURE_BTI, 180, 250
__ARM_FEATURE_CLZ, 180, 250
__ARM_FEATURE_COMPLEX, 180, 251
__ARM_FEATURE_CRC32, 180, 251
__ARM_FEATURE_CRYPTO, 181, 251
__ARM_FEATURE_DIRECTED_ROUNDING, 181, 251
__ARM_FEATURE_DIV, 181, 251
__ARM_FEATURE_DOTPROD, 181, 251
__ARM_FEATURE_FMA, 181, 251
__ARM_FEATURE_FP16_FML, 181, 251
__ARM_FEATURE_FP16_SCALAR_ARITHMETIC, 181, 251
__ARM_FEATURE_FP16_VECTOR_ARITHMETIC, 181, 251
__ARM_FEATURE_FRINT, 181, 252
__ARM_FEATURE_IDIV, 181, 252
__ARM_FEATURE_JCVT, 182, 252
__ARM_FEATURE_LDREX, 182, 252
__ARM_FEATURE_NUMERIC_MAXMIN, 182, 252
__ARM_FEATURE_PAUTH, 182, 252
__ARM_FEATURE_QRDMX, 182, 252
__ARM_FEATURE_RCPC, 182, 252
__ARM_FEATURE_SHA2, 182, 252
__ARM_FEATURE_SHA3, 182, 252
__ARM_FEATURE_SHA512, 182, 253
__ARM_FEATURE_UNALIGNED, 182, 253
__ARM_FP, 183, 253
__ARM_FP16_ARGS, 183, 253
__ARM_FP16_FORMAT_IEEE, 183, 253
__ARM_NEON, 183, 253
__ARM_NEON_FP, 183, 253
__ARM_NEON__, 183, 253
__ARM_PCS_AAPCS64, 183, 253
__ARM_SIZEOF_MINIMAL_ENUM, 183, 253
__ARM_SIZEOF_WCHAR_T, 183, 254
__ARM_STATE_ZA, 183, 254
__ARM_STATE_ZT0, 184, 254
__ATOMIC_ACQUIRE, 184, 254
__ATOMIC_ACQ_REL, 184, 254
__ATOMIC_CONSUME, 184, 254
__ATOMIC_RELAXED, 184, 254
__ATOMIC_RELEASE, 184, 254
__ATOMIC_SEQ_CST, 184, 254
__BIGGEST_ALIGNMENT__, 184, 254
__BITINT_MAXWIDTH__, 184, 255
__BLOCKS__, 185, 255
__BOOL_WIDTH__, 185, 255
__BYTE_ORDER__, 185, 255
__CHAR16_TYPE__, 185, 255
__CHAR32_TYPE__, 185, 255
__CHAR_BIT__, 185, 255
__CLANG_ATOMIC_BOOL_LOCK_FREE, 185, 255
__CLANG_ATOMIC_CHAR16_T_LOCK_FREE, 185, 256
__CLANG_ATOMIC_CHAR32_T_LOCK_FREE, 185, 256
__CLANG_ATOMIC_CHAR_LOCK_FREE, 186, 256
__CLANG_ATOMIC_INT_LOCK_FREE, 186, 256
__CLANG_ATOMIC_LLONG_LOCK_FREE, 186, 256
__CLANG_ATOMIC_LONG_LOCK_FREE, 186, 256
__CLANG_ATOMIC_POINTER_LOCK_FREE, 186, 256
__CLANG_ATOMIC_SHORT_LOCK_FREE, 186, 256
__CLANG_ATOMIC_WCHAR_T_LOCK_FREE, 186, 256
__CONSTANT_CFSTRINGS__, 187, 257
__DBL_DECIMAL_DIG__, 193, 263
__DBL_DENORM_MIN__, 193, 263
__DBL_DIG__, 193, 263
__DBL_EPSILON__, 193, 263
__DBL_HAS_DENORM__, 193, 263
__DBL_HAS_INFINITY__, 193, 263
__DBL_HAS_QUIET_NAN__, 193, 263
__DBL_MANT_DIG__, 193, 263
__DBL_MAX_10_EXP__, 193, 264
__DBL_MAX_EXP__, 194, 264
__DBL_MAX__, 193, 264
__DBL_MIN_10_EXP__, 194, 264
__DBL_MIN_EXP__, 194, 264
__DBL_MIN__, 194, 264
__DBL_NORM_MAX__, 194, 264
__DECIMAL_DIG__, 194, 264
__DEPRECATED, 194, 264
__DYNAMIC__, 194, 264
__ENVIRONMENT_MAC_OS_X_VERSION_MIN_REQUIRED__, 194, 265
__ENVIRONMENT_OS_VERSION_MIN_REQUIRED__, 194, 265
__EXCEPTIONS, 195, 265
__FINITE_MATH_ONLY__, 195, 265

__FLT16_DECIMAL_DIG__, 195, 265
__FLT16_DENORM_MIN__, 195, 265
__FLT16_DIG__, 195, 265
__FLT16_EPSILON__, 195, 265
__FLT16_HAS_DENORM__, 195, 265
__FLT16_HAS_INFINITY__, 195, 265
__FLT16_HAS_QUIET_NAN__, 195, 266
__FLT16_MANT_DIG__, 195, 266
__FLT16_MAX_10_EXP__, 196, 266
__FLT16_MAX_EXP__, 196, 266
__FLT16_MAX__, 196, 266
__FLT16_MIN_10_EXP__, 196, 266
__FLT16_MIN_EXP__, 196, 266
__FLT16_MIN__, 196, 266
__FLT16_NORM_MAX__, 196, 266
__FLT_DECIMAL_DIG__, 196, 266
__FLT_DENORM_MIN__, 196, 267
__FLT_DIG__, 196, 267
__FLT_EPSILON__, 197, 267
__FLT_HAS_DENORM__, 197, 267
__FLT_HAS_INFINITY__, 197, 267
__FLT_HAS_QUIET_NAN__, 197, 267
__FLT_MANT_DIG__, 197, 267
__FLT_MAX_10_EXP__, 197, 267
__FLT_MAX_EXP__, 197, 267
__FLT_MAX__, 197, 267
__FLT_MIN_10_EXP__, 197, 268
__FLT_MIN_EXP__, 198, 268
__FLT_MIN__, 197, 268
__FLT_NORM_MAX__, 198, 268
__FLT_RADIX__, 198, 268
__FPCLASS_NEGINF, 198, 268
__FPCLASS_NEGNORMAL, 198, 268
__FPCLASS_NEGSUBNORMAL, 198, 268
__FPCLASS_NEGZERO, 198, 269
__FPCLASS_POSINF, 198, 269
__FPCLASS_POSNORMAL, 199, 269
__FPCLASS_POSSUBNORMAL, 199, 269
__FPCLASS_POSZERO, 199, 269
__FPCLASS_QNAN, 199, 269
__FPCLASS_SNAN, 199, 269
__FP_FAST_FMA, 198, 268
__FP_FAST_FMAF, 198, 268
__GCC_ASM_FLAG_OUTPUTS__, 199, 269
__GCC_ATOMIC_BOOL_LOCK_FREE, 199, 269
__GCC_ATOMIC_CHAR16_T_LOCK_FREE, 199, 269
__GCC_ATOMIC_CHAR32_T_LOCK_FREE, 199, 270
__GCC_ATOMIC_CHAR_LOCK_FREE, 199, 270
__GCC_ATOMIC_INT_LOCK_FREE, 200, 270
__GCC_ATOMIC_LLONG_LOCK_FREE, 200, 270
__GCC_ATOMIC_LONG_LOCK_FREE, 200, 270
__GCC_ATOMIC_POINTER_LOCK_FREE, 200, 270
__GCC_ATOMIC_SHORT_LOCK_FREE, 200, 270
__GCC_ATOMIC_TEST_AND_SET_TRUEVAL, 200, 270
__GCC_ATOMIC_WCHAR_T_LOCK_FREE, 200, 270
__GCC_CONSTRUCTIVE_SIZE, 200, 270
__GCC_DESTRUCTIVE_SIZE, 200, 271
__GCC_HAVE_DWARF2_CFI_ASM, 200, 271
__GCC_HAVE_SYNC_COMPARE_AND_SWAP_1, 201, 271
__GCC_HAVE_SYNC_COMPARE_AND_SWAP_16, 201, 271
__GCC_HAVE_SYNC_COMPARE_AND_SWAP_2, 201, 271
__GCC_HAVE_SYNC_COMPARE_AND_SWAP_4, 201, 271
__GCC_HAVE_SYNC_COMPARE_AND_SWAP_8, 201, 271
__GLIBCXX_BITSIZE_INT_N_0, 201, 271
__GLIBCXX_TYPE_INT_N_0, 201, 271
__GNUC_GNU_INLINE__, 201, 272
__GNUC_MINOR__, 201, 272
__GNUC_PATCHLEVEL__, 202, 272
__GNUC__, 201, 271
__GNUG__, 202, 272
__GXX_ABI_VERSION, 202, 272
__GXX_EXPERIMENTAL_CXX0X__, 202, 272
__GXX_RTTI, 202, 272
__GXX_WEAK__, 202, 272
__HAVE_FUNCTION_MULTI_VERSIONING, 202, 272
__INT16_C_SUFFIX__, 202, 272
__INT16_FMTd__, 202, 273
__INT16_FMTi__, 202, 273
__INT16_MAX__, 203, 273
__INT16_TYPE__, 203, 273
__INT32_C_SUFFIX__, 203, 273
__INT32_FMTd__, 203, 273
__INT32_FMTi__, 203, 273
__INT32_MAX__, 203, 273
__INT32_TYPE__, 203, 273
__INT64_C_SUFFIX__, 203, 273
__INT64_FMTd__, 203, 274
__INT64_FMTi__, 203, 274
__INT64_MAX__, 204, 274
__INT64_TYPE__, 204, 274
__INT8_C_SUFFIX__, 204, 274
__INT8_FMTd__, 204, 274
__INT8_FMTi__, 204, 274
__INT8_MAX__, 204, 274
__INT8_TYPE__, 204, 274
__INTMAX_C_SUFFIX__, 208, 279
__INTMAX_FMTd__, 209, 279
__INTMAX_FMTi__, 209, 279
__INTMAX_MAX__, 209, 279
__INTMAX_TYPE__, 209, 279
__INTMAX_WIDTH__, 209, 279
__INTPTR_FMTd__, 209, 279
__INTPTR_FMTi__, 209, 279
__INTPTR_MAX__, 209, 279
__INTPTR_TYPE__, 209, 280

__INTPTR_WIDTH__, 209, 280
__INT_FAST16_FMTd__, 204, 274
__INT_FAST16_FMTi__, 204, 275
__INT_FAST16_MAX__, 204, 275
__INT_FAST16_TYPE__, 205, 275
__INT_FAST16_WIDTH__, 205, 275
__INT_FAST32_FMTd__, 205, 275
__INT_FAST32_FMTi__, 205, 275
__INT_FAST32_MAX__, 205, 275
__INT_FAST32_TYPE__, 205, 275
__INT_FAST32_WIDTH__, 205, 275
__INT_FAST64_FMTd__, 205, 275
__INT_FAST64_FMTi__, 205, 276
__INT_FAST64_MAX__, 205, 276
__INT_FAST64_TYPE__, 206, 276
__INT_FAST64_WIDTH__, 206, 276
__INT_FAST8_FMTd__, 206, 276
__INT_FAST8_FMTi__, 206, 276
__INT_FAST8_MAX__, 206, 276
__INT_FAST8_TYPE__, 206, 276
__INT_FAST8_WIDTH__, 206, 276
__INT_LEAST16_FMTd__, 206, 276
__INT_LEAST16_FMTi__, 206, 277
__INT_LEAST16_MAX__, 206, 277
__INT_LEAST16_TYPE__, 207, 277
__INT_LEAST16_WIDTH__, 207, 277
__INT_LEAST32_FMTd__, 207, 277
__INT_LEAST32_FMTi__, 207, 277
__INT_LEAST32_MAX__, 207, 277
__INT_LEAST32_TYPE__, 207, 277
__INT_LEAST32_WIDTH__, 207, 277
__INT_LEAST64_FMTd__, 207, 277
__INT_LEAST64_FMTi__, 207, 278
__INT_LEAST64_MAX__, 207, 278
__INT_LEAST64_TYPE__, 208, 278
__INT_LEAST64_WIDTH__, 208, 278
__INT_LEAST8_FMTd__, 208, 278
__INT_LEAST8_FMTi__, 208, 278
__INT_LEAST8_MAX__, 208, 278
__INT_LEAST8_TYPE__, 208, 278
__INT_LEAST8_WIDTH__, 208, 278
__INT_MAX__, 208, 278
__INT_WIDTH__, 208, 279
__LDBL_DECIMAL_DIG__, 210, 280
__LDBL_DENORM_MIN__, 210, 280
__LDBL_DIG__, 210, 280
__LDBL_EPSILON__, 210, 280
__LDBL_HAS_DENORM__, 210, 280
__LDBL_HAS_INFINITY__, 210, 280
__LDBL_HAS_QUIET_NAN__, 210, 280
__LDBL_MANT_DIG__, 210, 280
__LDBL_MAX_10_EXP__, 210, 281
__LDBL_MAX_EXP__, 211, 281
__LDBL_MAX__, 210, 281
__LDBL_MIN_10_EXP__, 211, 281
__LDBL_MIN_EXP__, 211, 281
__LDBL_MIN__, 211, 281
__LDBL_NORM_MAX__, 211, 281
__LITTLE_ENDIAN__, 211, 281
__LLONG_WIDTH__, 211, 281
__LONG_LONG_MAX__, 211, 282
__LONG_MAX__, 211, 282
__LONG_WIDTH__, 212, 282
__LP64__, 212, 282
__MACH__, 212, 282
__MEMORY_SCOPE_DEVICE, 212, 282
__MEMORY_SCOPE_SINGLE, 212, 282
__MEMORY_SCOPE_SYSTEM, 212, 282
__MEMORY_SCOPE_WRKGRP, 212, 282
__MEMORY_SCOPE_WVFRNT, 212, 282
__NO_INLINE__, 212, 283
__NO_MATH_ERRNO__, 212, 283
__OBJC_BOOL_IS_BOOL, 213, 283
__OPENCL_MEMORY_SCOPE_ALL_SVM_DEVICES, 213, 283
__OPENCL_MEMORY_SCOPE_DEVICE, 213, 283
__OPENCL_MEMORY_SCOPE_SUB_GROUP, 213, 283
__OPENCL_MEMORY_SCOPE_WORK_GROUP, 213, 283
__OPENCL_MEMORY_SCOPE_WORK_ITEM, 213, 284
__ORDER_BIG_ENDIAN__, 213, 284
__ORDER_LITTLE_ENDIAN__, 214, 284
__ORDER_PDP_ENDIAN__, 214, 284
__PIC__, 214, 284
__POINTER_WIDTH__, 214, 284
__PRAGMA_REDEFINE_EXTNAME, 214, 284
__PTRDIFF_FMTd__, 214, 284
__PTRDIFF_FMTi__, 214, 285
__PTRDIFF_MAX__, 214, 285
__PTRDIFF_TYPE__, 215, 285
__PTRDIFF_WIDTH__, 215, 285
__REGISTER_PREFIX__, 215, 285
__SCHAR_MAX__, 215, 285
__SHRT_MAX__, 215, 285
__SHRT_WIDTH__, 215, 285
__SIG_ATOMIC_MAX__, 215, 285
__SIG_ATOMIC_WIDTH__, 215, 285
__SIZEOF_DOUBLE__, 216, 286
__SIZEOF_FLOAT__, 216, 286
__SIZEOF_INT128__, 216, 286
__SIZEOF_INT__, 216, 287
__SIZEOF_LONG_DOUBLE__, 217, 287
__SIZEOF_LONG_LONG__, 217, 287
__SIZEOF_LONG__, 216, 287
__SIZEOF_POINTER__, 217, 287
__SIZEOF_PTRDIFF_T__, 217, 287
__SIZEOF_SHORT__, 217, 287
__SIZEOF_SIZE_T__, 217, 287
__SIZEOF_WCHAR_T__, 217, 287
__SIZEOF_WINT_T__, 217, 287
__SIZE_FMTX__, 216, 286
__SIZE_FMTo__, 215, 286
__SIZE_FMTu__, 215, 286
__SIZE_FMTx__, 216, 286

__SIZE_MAX__, 216, 286
__SIZE_TYPE__, 216, 286
__SIZE_WIDTH__, 216, 286
__SSP__, 217, 288
__STDCPP_DEFAULT_NEW_ALIGNMENT__, 218, 288
__STDCPP_THREADS__, 218, 289
__STDC_EMBED_EMPTY__, 218, 288
__STDC_EMBED_FOUND__, 218, 288
__STDC_EMBED_NOT_FOUND__, 218, 288
__STDC_HOSTED__, 218, 288
__STDC_NO_THREADS__, 218, 288
__STDC_UTF_16__, 218, 288
__STDC_UTF_32__, 218, 288
__STDC__, 217, 288
__UINT16_C_SUFFIX__, 219, 289
__UINT16_FMTX__, 219, 289
__UINT16_FMTo__, 219, 289
__UINT16_FMTu__, 219, 289
__UINT16_FMTx__, 219, 289
__UINT16_MAX__, 219, 289
__UINT16_TYPE__, 219, 289
__UINT32_C_SUFFIX__, 219, 289
__UINT32_FMTX__, 220, 290
__UINT32_FMTo__, 219, 290
__UINT32_FMTu__, 219, 290
__UINT32_FMTx__, 220, 290
__UINT32_MAX__, 220, 290
__UINT32_TYPE__, 220, 290
__UINT64_C_SUFFIX__, 220, 290
__UINT64_FMTX__, 220, 290
__UINT64_FMTo__, 220, 290
__UINT64_FMTu__, 220, 290
__UINT64_FMTx__, 220, 291
__UINT64_MAX__, 220, 291
__UINT64_TYPE__, 221, 291
__UINT8_C_SUFFIX__, 221, 291
__UINT8_FMTX__, 221, 291
__UINT8_FMTo__, 221, 291
__UINT8_FMTu__, 221, 291
__UINT8_FMTx__, 221, 291
__UINT8_MAX__, 221, 291
__UINT8_TYPE__, 221, 291
__UINTMAX_C_SUFFIX__, 226, 296
__UINTMAX_FMTX__, 226, 297
__UINTMAX_FMTo__, 226, 296
__UINTMAX_FMTu__, 226, 297
__UINTMAX_FMTx__, 227, 297
__UINTMAX_MAX__, 227, 297
__UINTMAX_TYPE__, 227, 297
__UINTMAX_WIDTH__, 227, 297
__UINTPTR_FMTX__, 227, 297
__UINTPTR_FMTo__, 227, 297
__UINTPTR_FMTu__, 227, 297
__UINTPTR_FMTx__, 227, 297
__UINTPTR_MAX__, 227, 298
__UINTPTR_TYPE__, 227, 298
__UINTPTR_WIDTH__, 228, 298
__UINT_FAST16_FMTX__, 222, 292
__UINT_FAST16_FMTo__, 221, 292
__UINT_FAST16_FMTu__, 221, 292
__UINT_FAST16_FMTx__, 222, 292
__UINT_FAST16_MAX__, 222, 292
__UINT_FAST16_TYPE__, 222, 292
__UINT_FAST32_FMTX__, 222, 292
__UINT_FAST32_FMTo__, 222, 292
__UINT_FAST32_FMTu__, 222, 292
__UINT_FAST32_FMTx__, 222, 292
__UINT_FAST32_MAX__, 222, 293
__UINT_FAST32_TYPE__, 222, 293
__UINT_FAST64_FMTX__, 223, 293
__UINT_FAST64_FMTo__, 223, 293
__UINT_FAST64_FMTu__, 223, 293
__UINT_FAST64_FMTx__, 223, 293
__UINT_FAST64_MAX__, 223, 293
__UINT_FAST64_TYPE__, 223, 293
__UINT_FAST8_FMTX__, 223, 294
__UINT_FAST8_FMTo__, 223, 293
__UINT_FAST8_FMTu__, 223, 293
__UINT_FAST8_FMTx__, 223, 294
__UINT_FAST8_MAX__, 224, 294
__UINT_FAST8_TYPE__, 224, 294
__UINT_LEAST16_FMTX__, 224, 294
__UINT_LEAST16_FMTo__, 224, 294
__UINT_LEAST16_FMTu__, 224, 294
__UINT_LEAST16_FMTx__, 224, 294
__UINT_LEAST16_MAX__, 224, 294
__UINT_LEAST16_TYPE__, 224, 294
__UINT_LEAST32_FMTX__, 225, 295
__UINT_LEAST32_FMTo__, 224, 295
__UINT_LEAST32_FMTu__, 224, 295
__UINT_LEAST32_FMTx__, 225, 295
__UINT_LEAST32_MAX__, 225, 295
__UINT_LEAST32_TYPE__, 225, 295
__UINT_LEAST64_FMTX__, 225, 295
__UINT_LEAST64_FMTo__, 225, 295
__UINT_LEAST64_FMTu__, 225, 295
__UINT_LEAST64_FMTx__, 225, 295
__UINT_LEAST64_MAX__, 225, 296
__UINT_LEAST64_TYPE__, 225, 296
__UINT_LEAST8_FMTX__, 226, 296
__UINT_LEAST8_FMTo__, 226, 296
__UINT_LEAST8_FMTu__, 226, 296
__UINT_LEAST8_FMTx__, 226, 296
__UINT_LEAST8_MAX__, 226, 296
__UINT_LEAST8_TYPE__, 226, 296
__USER_LABEL_PREFIX__, 228, 298
__VERSION__, 228, 298
__WCHAR_MAX__, 228, 298
__WCHAR_TYPE__, 228, 298
__WCHAR_WIDTH__, 228, 298
__WINT_MAX__, 228, 299
__WINT_TYPE__, 228, 299
__WINT_WIDTH__, 229, 299
__aarch64__, 178, 248
__apple_build_version__, 179, 249

__arm64, 179, 249
__arm64__, 179, 249
__block, 184, 255
__clang__, 185, 255
__clang_literal_encoding__, 186, 256
__clang_major__, 186, 257
__clang_minor__, 186, 257
__clang_patchlevel__, 187, 257
__clang_version__, 187, 257
__clang_wide_literal_encoding__, 187, 257
__cplusplus, 187, 257
__cpp_aggregate_bases, 187, 257
__cpp_aggregate_nsdmi, 187, 257
__cpp_alias_templates, 187, 257
__cpp_aligned_new, 187, 258
__cpp_attributes, 187, 258
__cpp_binary_literals, 188, 258
__cpp_capture_star_this, 188, 258
__cpp_constexpr, 188, 258
__cpp_constexpr_in_decltype, 188, 258
__cpp_decltype, 188, 258
__cpp_decltype_auto, 188, 258
__cpp_deduction_guides, 188, 258
__cpp_delegating_constructors, 188, 258
__cpp_deleted_function, 188, 259
__cpp_digit_separators, 188, 259
__cpp_enumerator_attributes, 189, 259
__cpp_exceptions, 189, 259
__cpp_fold_expressions, 189, 259
__cpp_generic_lambdas, 189, 259
__cpp_guaranteed_copy_elision, 189, 259
__cpp_hex_float, 189, 259
__cpp_if_constexpr, 189, 259
__cpp_impl_destroying_delete, 189, 259
__cpp_inheriting_constructors, 189, 260
__cpp_init_captures, 189, 260
__cpp_initializer_lists, 190, 260
__cpp_inline_variables, 190, 260
__cpp_lambdas, 190, 260
__cpp_named_character_escapes, 190, 260
__cpp_namespace_attributes, 190, 260
__cpp_nested_namespace_definitions, 190, 260
__cpp_noexcept_function_type, 190, 260
__cpp_nontype_template_args, 190, 260
__cpp_nontype_template_parameter_auto, 190, 261
__cpp_nsdmi, 190, 261
__cpp_pack_indexing, 191, 261
__cpp_placeholder_variables, 191, 261
__cpp_range_based_for, 191, 261
__cpp_raw_strings, 191, 261
__cpp_ref_qualifiers, 191, 261
__cpp_return_type_deduction, 191, 261
__cpp_rtti, 191, 261
__cpp_rvalue_references, 191, 261
__cpp_static_assert, 191, 262
__cpp_static_call_operator, 191, 262
__cpp_structured_bindings, 192, 262
__cpp_template_auto, 192, 262
__cpp_template_template_args, 192, 262
__cpp_threadsafe_static_init, 192, 262
__cpp_unicode_characters, 192, 262
__cpp_unicode_literals, 192, 262
__cpp_user_defined_literals, 192, 262
__cpp_variable_templates, 192, 262
__cpp_variadic_templates, 192, 263
__cpp_variadic_using, 192, 263
__llvm__, 211, 281
__nonnull, 213, 283
__null_unspecified, 213, 283
__nullable, 213, 283
__pic__, 214, 284
__private_extern__, 214, 284
__strong, 218, 289
__unsafe_unretained, 228, 298
__weak, 228, 298
QT_CHARTS_LIB, 229, 299
QT_CHARTS_USE_NAMESPACE, 229
QT_CONCURRENT_LIB, 229
QT_CORE_LIB, 229, 299
QT_GUI_LIB, 229, 299
QT_NETWORK_LIB, 229, 299
QT_NO_DEBUG, 229, 299
QT_OPENGL_LIB, 229, 299
QT_OPENGLWIDGETS_LIB, 230, 300
QT_WIDGETS_LIB, 230, 300
SIZEOF_DPTR, 230, 300
TARGET_IPHONE_SIMULATOR, 230, 300
TARGET_OS_ARROW, 230, 300
TARGET_OS_BRIDGE, 230, 300
TARGET_OS_DRIVERKIT, 230, 300
TARGET_OS_EMBEDDED, 230, 300
TARGET_OS_IOS, 230, 300
TARGET_OS_IOSMAC, 230, 300
TARGET_OS_IPHONE, 231, 301
TARGET_OS_LINUX, 231, 301
TARGET_OS_MAC, 231, 301
TARGET_OS_MACCATALYST, 231, 301
TARGET_OS_NANO, 231, 301
TARGET_OS_OSX, 231, 301
TARGET_OS_SIMULATOR, 231, 301
TARGET_OS_TV, 231, 301
TARGET_OS_UIKITFORMAC, 231, 301
TARGET_OS_UNIX, 231, 301
TARGET_OS_VISION, 232, 302
TARGET_OS_WATCH, 232, 302
TARGET_OS_WIN32, 232, 302
TARGET_OS_WINDOWS, 232, 302
TARGET_OS_XR, 232, 302
moc_scanhistory.cpp
Q_CONSTINIT, 164, 166
mouseMoveEvent
DeviceNode, 36
mousePressEvent
DeviceNode, 36
mouseReleaseEvent
DeviceNode, 36

NetScanner_autogen/EWIEGA46WW/moc_deviceanalyzer.cpp, 130, 131
NetScanner_autogen/EWIEGA46WW/moc_deviceanalyzer.cpp.d, 135
NetScanner_autogen/EWIEGA46WW/moc_mainwindow.cpp, 139, 141
NetScanner_autogen/EWIEGA46WW/moc_mainwindow.cpp.d, 148
NetScanner_autogen/JRIAJ772TK/moc_deviceanalyzer.cpp, 132, 134
NetScanner_autogen/JRIAJ772TK/moc_deviceanalyzer.cpp.d, 135
NetScanner_autogen/JRIAJ772TK/moc_mainwindow.cpp, 144, 145
NetScanner_autogen/JRIAJ772TK/moc_mainwindow.cpp.d, 148
NetScanner_autogen/JRIAJ772TK/moc_networkscanner.cpp, 152, 153
NetScanner_autogen/JRIAJ772TK/moc_networkscanner.cpp.d, 155
NetScanner_autogen/JRIAJ772TK/moc_networktopology.cpp, 159, 160
NetScanner_autogen/JRIAJ772TK/moc_networktopology.cpp.d, 163
NetScanner_autogen/JRIAJ772TK/moc_scanhistory.cpp, 165, 167
NetScanner_autogen/JRIAJ772TK/moc_scanhistory.cpp.d, 168
NetScanner_autogen/moc_predefs.h, 238, 302
NetScanner_autogen/mocs_compilation.cpp, 309
Network Scanner 项目文档, 1
networkLayer
  DeviceNode, 36
NetworkScanner, 69
  ~NetworkScanner, 73
  executeProcess, 74
  generatePseudoMACFromIP, 74
  getAddressesToScan, 74
  getLocalNetworkAddresses, 75
  getMacAddressFromSystemCalls, 75
  getScannedHosts, 75
  hostFound, 76
  isHostReachable, 76
  isReachableOnPorts, 76
  isScanning, 77
  lookupHostName, 77
  lookupMacAddress, 77
  lookupMacVendor, 78
  m_activeHosts, 88
  m_endIPRange, 88
  m_isScanning, 88
  m_macAddressCache, 88
  m_mutex, 88
  m_portsToScan, 88
  m_scanFutures, 88
  m_scannedHosts, 88
  m_scannedHostsList, 89
  m_scanStrategy, 89
  m_scanTimeout, 89
  m_startIPRange, 89
  m_threadPool, 89
  m_totalHosts, 89
  m_useCustomRange, 89
  NetworkScanner, 73
  normalizeMacAddress, 78
  onHostNameLookedUp, 79
  onScanTaskFinished, 79
  performARPScan, 80
  pingTargetWithTimeout, 80
  processScanResults, 81
  quickPingScan, 81
  saveResultsToFile, 82
  scanError, 82
  scanFinished, 82
  scanHost, 83
  scanHostPorts, 84
  scanProgress, 84
  scanStarted, 84
  setCustomIPRange, 85
  setCustomPortsToScan, 85
  setScanStrategy, 86
  setScanTimeout, 86
  startScan, 86
  stopScan, 86
  updateScanProgress, 87
networkscanner.cpp, 316
networkscanner.h, 317
NetworkTopology, 90
  clear, 93
  deviceSelected, 93
  LAYOUT_AUTO, 92
  LAYOUT_GROUPED, 92
  LAYOUT_HIERARCHICAL, 92
  LayoutMode, 92
  m_controlPanel, 94
  m_currentHosts, 94
  m_layoutMode, 94
  m_topologyView, 94
  NetworkTopology, 92
  resetView, 93
  scale, 93
  setLayoutMode, 93
  updateTopology, 93
networktopology.cpp, 320
networktopology.h, 320
  CONNECTION_DIRECT, 322
  CONNECTION_ROUTED, 322
  CONNECTION_UNKNOWN, 322
  CONNECTION_VPN, 322
  CONNECTION_WIRELESS, 322
  ConnectionType, 321
  DEVICE_IOT, 322
  DEVICE_MOBILE, 322
  DEVICE_PC, 322
  DEVICE_PRINTER, 322
  DEVICE_ROUTER, 322

DEVICE_SERVER, 322
DEVICE_UNKNOWN, 322
DeviceType, 322
NetworkTopologyView, 95
autoLayout, 97
clear, 97
createConnection, 98
determineDeviceType, 98
groupedLayout, 98
hierarchicalLayout, 99
m_analyzer, 100
m_connections, 100
m_nodes, 100
m_scene, 100
NetworkTopologyView, 97
nodeSelected, 99
setHosts, 99
nodeSelected
NetworkTopologyView, 99
normalizeMacAddress
NetworkScanner, 78

onHostFound
MainWindow, 53
onHostNameLookedUp
NetworkScanner, 79
onScanError
MainWindow, 53
onScanFinished
MainWindow, 54
onScanProgress
MainWindow, 54
onScanStarted
MainWindow, 54
onScanTaskFinished
NetworkScanner, 79
onThemeChanged
MainWindow, 55
openPorts
HostInfo, 40

paint
ConnectionLine, 23
DeviceNode, 36
performARPScan
NetworkScanner, 80
performTraceRoute
TopologyAnalyzer, 122
pingTargetWithTimeout
NetworkScanner, 80
PLATFORM_ID
CMakeCXXCompilerId.cpp, 125
portDistribution
ScanSession, 109
processScanResults
NetworkScanner, 81

Q_CONSTINIT
moc_deviceanalyzer.cpp, 128, 131, 133
moc_mainwindow.cpp, 136, 140, 144
moc_networkscanner.cpp, 149, 152
moc_networktopology.cpp, 156, 160
moc_scanhistory.cpp, 164, 166
QT_CHARTS_LIB
moc_predefs.h, 229, 299
QT_CHARTS_USE_NAMESPACE
moc_predefs.h, 229
QT_CONCURRENT_LIB
moc_predefs.h, 229
QT_CORE_LIB
moc_predefs.h, 229, 299
QT_GUI_LIB
moc_predefs.h, 229, 299
QT_NETWORK_LIB
moc_predefs.h, 229, 299
QT_NO_DEBUG
moc_predefs.h, 229, 299
QT_OPENGL_LIB
moc_predefs.h, 229, 299
QT_OPENGLWIDGETS_LIB
moc_predefs.h, 230, 300
QT_WARNING_DISABLE_DEPRECATED, 19
QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN10MainWindowE
101
QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN11ScanHistoryE
101
QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN14DeviceAnalyz
102
QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN14NetworkScan
102
QT_WARNING_DISABLE_DEPRECATED::qt_meta_tag_ZN19NetworkTopol
103
QT_WIDGETS_LIB
moc_predefs.h, 230, 300
QUICK_SCAN
ScanStrategy, 112
quickPingScan
NetworkScanner, 81

reachableHosts
ScanSession, 109
README.dox, 325
README.md, 325
refreshTopology
MainWindow, 55
removeSession
ScanHistory, 107
resetView
NetworkTopology, 93
run
ScanTask, 117

saveHistoryToFile
MainWindow, 55
saveResults
MainWindow, 55
saveResultsToFile
NetworkScanner, 82

saveSettings
MainWindow, 56
saveToFile
ScanHistory, 108
saveTopologyImage
MainWindow, 56
scale
NetworkTopology, 93
scanError
NetworkScanner, 82
scanFinished
NetworkScanner, 82
ScanHistory, 103
addSession, 106
clearHistory, 106
compareScans, 106
getSession, 107
getSessions, 107
historyChanged, 107
loadFromFile, 107
m_sessions, 108
removeSession, 107
saveToFile, 108
ScanHistory, 106
sessionCount, 108
scanhistory.cpp, 325
scanhistory.h, 325
scanHost
NetworkScanner, 83
scanHostPorts
NetworkScanner, 84
ScanMode
ScanStrategy, 112
scanProgress
NetworkScanner, 84
ScanSession, 109
description, 110
hosts, 110
portDistribution, 109
reachableHosts, 109
scanTime, 111
totalHosts, 110
unreachableHosts, 110
scanStarted
NetworkScanner, 84
ScanStrategy, 111
DEEP_SCAN, 112
getMaxParallelTasks, 113
getMode, 113
getPortsToScan, 113
getScanTimeout, 113
m_hostResponseTimes, 114
m_mode, 114
QUICK_SCAN, 112
ScanMode, 112
ScanStrategy, 113
setMode, 114
STANDARD_SCAN, 112
updateHostResponseTime, 114
ScanTask, 115
m_address, 117
m_parent, 117
m_ports, 118
m_timeout, 118
run, 117
ScanTask, 117
scanTime
HostInfo, 40
ScanSession, 111
scheduleScan
MainWindow, 56
sessionCount
ScanHistory, 108
setConnectionType
ConnectionLine, 23
setCustomIPRange
NetworkScanner, 85
setCustomPortsToScan
NetworkScanner, 85
setDeviceType
DeviceNode, 37
setHosts
NetworkTopologyView, 99
setLayoutMode
NetworkTopology, 93
setMode
ScanStrategy, 114
setNetworkLayer
DeviceNode, 37
setPosition
DeviceNode, 37
setScanStrategy
NetworkScanner, 86
setScanTimeout
NetworkScanner, 86
setSubnetGroup
DeviceNode, 37
setupConnections
MainWindow, 57
showAbout
MainWindow, 58
showHistoryView
MainWindow, 59
showHostDetails
MainWindow, 59
showSettings
MainWindow, 59
showStatisticsView
MainWindow, 59
showTopologyView
MainWindow, 59
SIZEOF_DPTR
moc_predefs.h, 230, 300
STANDARD_SCAN
ScanStrategy, 112
startScan

MainWindow, 60
NetworkScanner, 86
stopScan
MainWindow, 60
NetworkScanner, 86
STRINGIFY
CMakeCXXCompilerId.cpp, 125
STRINGIFY_HELPER
CMakeCXXCompilerId.cpp, 125
subnetGroup
DeviceNode, 37

TARGET_IPHONE_SIMULATOR
moc_predefs.h, 230, 300
TARGET_OS_ARROW
moc_predefs.h, 230, 300
TARGET_OS_BRIDGE
moc_predefs.h, 230, 300
TARGET_OS_DRIVERKIT
moc_predefs.h, 230, 300
TARGET_OS_EMBEDDED
moc_predefs.h, 230, 300
TARGET_OS_IOS
moc_predefs.h, 230, 300
TARGET_OS_IOSMAC
moc_predefs.h, 230, 300
TARGET_OS_IPHONE
moc_predefs.h, 231, 301
TARGET_OS_LINUX
moc_predefs.h, 231, 301
TARGET_OS_MAC
moc_predefs.h, 231, 301
TARGET_OS_MACCATALYST
moc_predefs.h, 231, 301
TARGET_OS_NANO
moc_predefs.h, 231, 301
TARGET_OS_OSX
moc_predefs.h, 231, 301
TARGET_OS_SIMULATOR
moc_predefs.h, 231, 301
TARGET_OS_TV
moc_predefs.h, 231, 301
TARGET_OS_UIKITFORMAC
moc_predefs.h, 231, 301
TARGET_OS_UNIX
moc_predefs.h, 231, 301
TARGET_OS_VISION
moc_predefs.h, 232, 302
TARGET_OS_WATCH
moc_predefs.h, 232, 302
TARGET_OS_WIN32
moc_predefs.h, 232, 302
TARGET_OS_WINDOWS
moc_predefs.h, 232, 302
TARGET_OS_XR
moc_predefs.h, 232, 302
toggleDarkMode
MainWindow, 60
togglePortScanOptions
MainWindow, 61
toggleRangeOptions
MainWindow, 61
TopologyAnalyzer, 118
analyzeSubnets, 120
analyzeTTLLayers, 120
calculateSubnet, 121
clusterDevicesByResponseTime, 121
getTTLValue, 121
inferDeviceConnections, 121
inSameSubnet, 122
performTraceRoute, 122
TopologyAnalyzer, 119
totalHosts
ScanSession, 110

unreachableHosts
ScanSession, 110
updateHostResponseTime
ScanStrategy, 114
updateNetworkTopology
MainWindow, 61
updatePortsList
MainWindow, 62
updatePosition
ConnectionLine, 23
updateScanProgress
NetworkScanner, 87
updateStatistics
MainWindow, 62
updateTopology
NetworkTopology, 93

网络扫描器, 3