HardwareQualification

chromeos-hwqual-x86-generic-0.7.45.0

Generated by Doxygen 1.6.3

Fri May 28 12:52:56 2010

Contents

1 Main Page **1**

1.1 Glossary . . . 1

1.2 Test Setup . . . 1

1.3 Automated and Semi-Automated Test Runs . . . 2

1.4 Manual Test Runs . . . 4

1.5 Reporting Results . . . 5

2 Namespace Index **7**

2.1 Package List . . . 7

3 Class Index **9**

3.1 Class List . . . 9

4 File Index **11**

4.1 File List . . . 11

5 Namespace Documentation **13**

5.1 Package audiovideo_FFMPEG . . . 13

5.1.1 Detailed Description . . . 13

5.2 Package audiovideo_V4L2 . . . 15

5.2.1 Detailed Description . . . 15

5.3 Package compilebench . . . 16

5.3.1 Detailed Description . . . 16

5.3.2 Variable Documentation . . . 16

5.3.2.1 test_name . . . 16

5.4 Package desktopui_SunSpiderBench . . . 17

5.4.1 Detailed Description . . . 17

5.5 Package desktopui_V8Bench . . . 18

5.5.1 Detailed Description . . . 18

5.6 Package disktest . . . 19

5.6.1 Detailed Description . . . 19
5.7 Package firmware_RomSize . . . 20
5.7.1 Detailed Description . . . 20
5.8 Package firmware_VbootCrypto . . . 21
5.8.1 Detailed Description . . . 21
5.9 Package gl_Bench . . . 22
5.9.1 Detailed Description . . . 22
5.10 Package graphics_GLAPICheck . . . 23
5.10.1 Detailed Description . . . 23
5.11 Package graphics_SanAngeles . . . 24
5.11.1 Detailed Description . . . 24
5.12 Package graphics_TearTest . . . 25
5.12.1 Detailed Description . . . 25
5.12.2 Function Documentation . . . 25
5.12.2.1 html_button . . . 25
5.12.3 Variable Documentation . . . 26
5.12.3.1 TEMPLATE . . . 26
5.13 Package hardware_Backlight . . . 27
5.13.1 Detailed Description . . . 27
5.13.2 Function Documentation . . . 27
5.13.2.1 backlight_tool . . . 27
5.14 Package hardware_BluetoothSemiAuto . . . 28
5.14.1 Detailed Description . . . 28
5.14.2 Variable Documentation . . . 28
5.14.2.1 _HREF_END . . . 28
5.14.2.2 _HREF_START . . . 29
5.14.2.3 _QUESTION_START . . . 29
5.15 Package hardware_Components . . . 30
5.15.1 Detailed Description . . . 30
5.16 Package hardware_DiskSize . . . 31
5.16.1 Detailed Description . . . 31
5.17 Package hardware_KeyboardAssembly . . . 32
5.17.1 Detailed Description . . . 32
5.18 Package hardware_MemoryThroughput . . . 33
5.18.1 Detailed Description . . . 33
5.19 Package hardware_MemoryTotalSize . . . 34

5.19.1 Detailed Description . . . 34
5.20 Package hardware_Resolution . . . 35
5.20.1 Detailed Description . . . 35
5.20.2 Variable Documentation . . . 35
5.20.2.1 __author__ . . . 35
5.21 Package hardware_SAT . . . 36
5.21.1 Detailed Description . . . 36
5.22 Package hardware_SsdDetection . . . 37
5.22.1 Detailed Description . . . 37
5.23 Package hardware_StorageFio . . . 38
5.23.1 Detailed Description . . . 38
5.24 Package hardware_Touchpad . . . 39
5.24.1 Detailed Description . . . 39
5.25 Package hardware_VideoOutSemiAuto . . . 40
5.25.1 Detailed Description . . . 40
5.26 Package ltp . . . 41
5.26.1 Detailed Description . . . 41
5.27 Package network_DisableInterface . . . 42
5.27.1 Detailed Description . . . 42
5.28 Package network_WiFiCaps . . . 43
5.28.1 Detailed Description . . . 43
5.29 Package platform_AesThroughput . . . 44
5.29.1 Detailed Description . . . 44
5.30 Package platform_BootPerf . . . 45
5.30.1 Detailed Description . . . 45
5.31 Package platform_KernelVersion . . . 46
5.31.1 Detailed Description . . . 46
5.32 Package power_BatteryCharge . . . 47
5.32.1 Detailed Description . . . 47
5.33 Package power_CPUFreq . . . 48
5.33.1 Detailed Description . . . 48
5.34 Package power_CPUIdle . . . 49
5.34.1 Detailed Description . . . 49
5.35 Package power_Draw . . . 50
5.35.1 Detailed Description . . . 50
5.36 Package power_LoadTest . . . 51

5.36.1 Detailed Description . . . 51
5.36.2 Variable Documentation . . . 52
5.36.2.1 params_dict . . . 52
5.37 Package power_Resume . . . 53
5.37.1 Detailed Description . . . 53
5.38 Package realtimecomm_GTalkAudioPlayground . . . 54
5.38.1 Detailed Description . . . 54
5.38.2 Variable Documentation . . . 55
5.38.2.1 SLEEP_DURATION . . . 55
5.38.2.2 WARMUP_TIME . . . 55
5.39 Package realtimecomm_GTalkPlayground . . . 56
5.39.1 Detailed Description . . . 56
5.39.2 Variable Documentation . . . 57
5.39.2.1 SLEEP_DURATION . . . 57
5.39.2.2 WARMUP_TIME . . . 57
5.40 Package unixbench . . . 58
5.40.1 Detailed Description . . . 58

6 Class Documentation 59

6.1 hardware_BluetoothSemiAuto.Agent Class Reference . . . 59
6.1.1 Detailed Description . . . 59
6.1.2 Member Function Documentation . . . 59
6.1.2.1 Cancel . . . 59
6.1.2.2 Release . . . 59
6.1.2.3 RequestPinCode . . . 60
6.1.3 Member Data Documentation . . . 60
6.1.3.1 in_signature . . . 60
6.2 audiovideo_FFMPEG.audiovideo_FFMPEG Class Reference . . . 61
6.2.1 Detailed Description . . . 61
6.2.2 Member Function Documentation . . . 61
6.2.2.1 run_once . . . 61
6.2.2.2 run_testcase . . . 62
6.2.2.3 setup . . . 63
6.2.3 Member Data Documentation . . . 63
6.2.3.1 max_tpf_audio . . . 63
6.2.3.2 min_fps_video . . . 63
6.2.3.3 performance_results . . . 63

6.2.3.4 version . . . 63
6.3 audiovideo_V4L2.audiovideo_V4L2 Class Reference . . . 64
6.3.1 Detailed Description . . . 64
6.3.2 Member Function Documentation . . . 64
6.3.2.1 find_video_capture_devices . . . 64
6.3.2.2 run_once . . . 65
6.3.2.3 run_v4l2_capture_test . . . 65
6.3.2.4 run_v4l2_capture_tests . . . 65
6.3.2.5 run_v4l2_default_capture_test . . . 66
6.3.2.6 run_v4l2_unittests . . . 67
6.3.2.7 setup . . . 68
6.3.2.8 unittest_passed . . . 68
6.3.3 Member Data Documentation . . . 68
6.3.3.1 assert_mandatory_controls . . . 68
6.3.3.2 executable . . . 68
6.3.3.3 preserve_srcdir . . . 68
6.3.3.4 support_readwrite . . . 68
6.3.3.5 support_streaming . . . 69
6.3.3.6 supported_controls . . . 69
6.3.3.7 supported_formats . . . 69
6.3.3.8 v4l2_devices . . . 69
6.3.3.9 v4l2_major_dev_num . . . 69
6.3.3.10 v4l2_minor_dev_num_max . . . 69
6.3.3.11 v4l2_minor_dev_num_min . . . 69
6.3.3.12 version . . . 69
6.4 compilebench.compilebench Class Reference . . . 70
6.4.1 Detailed Description . . . 70
6.4.2 Member Function Documentation . . . 70
6.4.2.1 __format_results . . . 70
6.4.2.2 run_once . . . 71
6.4.2.3 setup . . . 71
6.4.3 Member Data Documentation . . . 71
6.4.3.1 tarball . . . 71
6.4.3.2 version . . . 72
6.5 power_CPUFreq.cpufreq Class Reference . . . 73
6.5.1 Detailed Description . . . 73

6.5.2 Member Function Documentation . . . 73
6.5.2.1 __init__ . . . 73
6.5.2.2 __read_file . . . 73
6.5.2.3 __write_file . . . 74
6.5.2.4 get_available_frequencies . . . 74
6.5.2.5 get_available_governors . . . 74
6.5.2.6 get_current_frequency . . . 74
6.5.2.7 get_current_governor . . . 75
6.5.2.8 restore_state . . . 75
6.5.2.9 save_state . . . 75
6.5.2.10 set_frequency . . . 75
6.5.2.11 set_governor . . . 76
6.5.3 Member Data Documentation . . . 76
6.5.3.1 __base_path . . . 76
6.5.3.2 __save_files_list . . . 76
6.6 power_CPUIdle.cpuidle Class Reference . . . 77
6.6.1 Detailed Description . . . 77
6.6.2 Member Function Documentation . . . 77
6.6.2.1 __init__ . . . 77
6.6.2.2 idle_time . . . 77
6.6.3 Member Data Documentation . . . 78
6.6.3.1 __base_path . . . 78
6.6.3.2 __states . . . 78
6.7 power_CPUIdle.cpuidle_state Class Reference . . . 79
6.7.1 Detailed Description . . . 79
6.7.2 Member Function Documentation . . . 79
6.7.2.1 __init__ . . . 79
6.7.2.2 __read_file . . . 79
6.7.2.3 idle_time . . . 80
6.7.3 Member Data Documentation . . . 80
6.7.3.1 __base_path . . . 80
6.7.3.2 __name . . . 80
6.8 power_CPUIdle.cpus Class Reference . . . 81
6.8.1 Detailed Description . . . 81
6.8.2 Member Function Documentation . . . 81
6.8.2.1 __init__ . . . 81

6.8.2.2 idle_time . . . 81
6.8.3 Member Data Documentation . . . 81
6.8.3.1 __base_path . . . 81
6.8.3.2 __cpus . . . 82
6.9 desktopui_SunSpiderBench.desktopui_SunSpiderBench Class Reference . . . 83
6.9.1 Detailed Description . . . 83
6.9.2 Member Function Documentation . . . 83
6.9.2.1 cleanup . . . 83
6.9.2.2 initialize . . . 83
6.9.2.3 run_once . . . 84
6.9.2.4 setup . . . 84
6.9.3 Member Data Documentation . . . 84
6.9.3.1 _test_url . . . 84
6.9.3.2 _testServer . . . 84
6.9.3.3 version . . . 84
6.10 desktopui_V8Bench.desktopui_V8Bench Class Reference . . . 85
6.10.1 Detailed Description . . . 85
6.10.2 Member Function Documentation . . . 85
6.10.2.1 cleanup . . . 85
6.10.2.2 initialize . . . 85
6.10.2.3 run_once . . . 86
6.10.2.4 setup . . . 86
6.10.3 Member Data Documentation . . . 86
6.10.3.1 _test_url . . . 86
6.10.3.2 _testServer . . . 86
6.10.3.3 version . . . 86
6.11 disktest.disktest Class Reference . . . 87
6.11.1 Detailed Description . . . 87
6.11.2 Member Function Documentation . . . 87
6.11.2.1 execute . . . 87
6.11.2.2 initialize . . . 88
6.11.2.3 setup . . . 88
6.11.2.4 test_one_disk_chunk . . . 88
6.11.3 Member Data Documentation . . . 88
6.11.3.1 chunk_mb . . . 88
6.11.3.2 memory_mb . . . 88

6.11.3.3 preserve_srcdir . . . 89
6.11.3.4 version . . . 89
6.12 firmware_RomSize.firmware_RomSize Class Reference . . . 90
6.12.1 Detailed Description . . . 90
6.12.2 Member Function Documentation . . . 90
6.12.2.1 run_once . . . 90
6.12.3 Member Data Documentation . . . 90
6.12.3.1 version . . . 90
6.13 firmware_VbootCrypto.firmware_VbootCrypto Class Reference . . . 91
6.13.1 Detailed Description . . . 91
6.13.2 Member Function Documentation . . . 92
6.13.2.1 __generate_test_cases . . . 92
6.13.2.2 __image_verification_test . . . 92
6.13.2.3 __output_result_keyvals . . . 92
6.13.2.4 __rollback_tests . . . 92
6.13.2.5 __rsa_benchmark . . . 93
6.13.2.6 __rsa_test . . . 93
6.13.2.7 __sha_benchmark . . . 93
6.13.2.8 __sha_test . . . 94
6.13.2.9 __splicing_tests . . . 94
6.13.2.10 __verify_image_benchmark . . . 94
6.13.2.11 run_benchmarks . . . 95
6.13.2.12 run_crypto . . . 95
6.13.2.13 run_once . . . 95
6.13.2.14 run_rollback . . . 95
6.13.2.15 run_splicing . . . 96
6.13.2.16 run_verification . . . 96
6.13.2.17 setup . . . 96
6.13.3 Member Data Documentation . . . 96
6.13.3.1 keyvals . . . 96
6.13.3.2 preserve_srcdir . . . 96
6.13.3.3 results . . . 96
6.13.3.4 version . . . 96
6.14 gl_Bench.gl_Bench Class Reference . . . 98
6.14.1 Detailed Description . . . 98
6.14.2 Member Function Documentation . . . 98

6.14.2.1 run_once . . . 98
6.14.2.2 setup . . . 98
6.14.3 Member Data Documentation . . . 99
6.14.3.1 preserve_srcdir . . . 99
6.14.3.2 results . . . 99
6.14.3.3 version . . . 99
6.15 graphics_GLAPICheck.graphics_GLAPICheck Class Reference . . . 100
6.15.1 Detailed Description . . . 100
6.15.2 Member Function Documentation . . . 100
6.15.2.1 __check_extensions . . . 100
6.15.2.2 __check_gl . . . 101
6.15.2.3 __check_gl_extensions_1x . . . 101
6.15.2.4 __check_gl_extensions_2x . . . 101
6.15.2.5 __check_gles . . . 102
6.15.2.6 __check_gles_extensions . . . 102
6.15.2.7 __check_x_extensions . . . 102
6.15.2.8 __run_x_cmd . . . 103
6.15.2.9 run_once . . . 103
6.15.2.10 setup . . . 104
6.15.3 Member Data Documentation . . . 104
6.15.3.1 error_message . . . 104
6.15.3.2 error_message . . . 104
6.15.3.3 preserve_srcdir . . . 104
6.15.3.4 version . . . 104
6.16 graphics_SanAngeles.graphics_SanAngeles Class Reference . . . 105
6.16.1 Detailed Description . . . 105
6.16.2 Member Function Documentation . . . 105
6.16.2.1 run_once . . . 105
6.16.2.2 setup . . . 105
6.16.3 Member Data Documentation . . . 106
6.16.3.1 preserve_srcdir . . . 106
6.16.3.2 version . . . 106
6.17 graphics_TearTest.graphics_TearTest Class Reference . . . 107
6.17.1 Detailed Description . . . 107
6.17.2 Member Function Documentation . . . 107
6.17.2.1 run_once . . . 107

6.17.2.2 setup . . . 108
6.17.3 Member Data Documentation . . . 108
6.17.3.1 version . . . 108
6.18 hardware_Backlight.hardware_Backlight Class Reference . . . 109
6.18.1 Detailed Description . . . 109
6.18.2 Member Function Documentation . . . 109
6.18.2.1 run_once . . . 109
6.18.3 Member Data Documentation . . . 109
6.18.3.1 version . . . 109
6.19 hardware_BluetoothSemiAuto.hardware_BluetoothSemiAuto Class Reference . . . 110
6.19.1 Detailed Description . . . 110
6.19.2 Member Function Documentation . . . 110
6.19.2.1 cleanup . . . 110
6.19.2.2 do_connect . . . 110
6.19.2.3 handle_error . . . 111
6.19.2.4 handle_reply . . . 111
6.19.2.5 initialize . . . 111
6.19.2.6 run_once . . . 112
6.19.3 Member Data Documentation . . . 112
6.19.3.1 mainloop . . . 112
6.19.3.2 version . . . 112
6.20 hardware_Components.hardware_Components Class Reference . . . 114
6.20.1 Detailed Description . . . 114
6.20.2 Member Function Documentation . . . 115
6.20.2.1 check_approved_part_id_existence . . . 115
6.20.2.2 check_component . . . 115
6.20.2.3 get_part_id_audio_codec . . . 115
6.20.2.4 get_part_id_bios . . . 116
6.20.2.5 get_part_id_cpu . . . 116
6.20.2.6 get_part_id_embedded_controller . . . 116
6.20.2.7 get_part_id_ethernet . . . 117
6.20.2.8 get_part_id_flash_chip . . . 117
6.20.2.9 get_part_id_storage . . . 117
6.20.2.10 get_part_id_wireless . . . 117
6.20.2.11 get_vendor_id_touchpad . . . 118
6.20.2.12 get_vendor_id_webcam . . . 118

6.20.2.13 initialize . . . 118
6.20.2.14 pformat . . . 118
6.20.2.15 run_once . . . 119
6.20.3 Member Data Documentation . . . 119
6.20.3.1 _approved . . . 119
6.20.3.2 _cids . . . 119
6.20.3.3 _failures . . . 120
6.20.3.4 _not_present . . . 120
6.20.3.5 _pci_cids . . . 120
6.20.3.6 _pp . . . 120
6.20.3.7 _system . . . 120
6.20.3.8 _usb_cids . . . 120
6.20.3.9 version . . . 120
6.21 hardware_DiskSize.hardware_DiskSize Class Reference . . . 121
6.21.1 Detailed Description . . . 121
6.21.2 Member Function Documentation . . . 121
6.21.2.1 run_once . . . 121
6.21.3 Member Data Documentation . . . 121
6.21.3.1 version . . . 121
6.22 hardware_KeyboardAssembly.hardware_KeyboardAssembly Class Reference . . . 122
6.22.1 Detailed Description . . . 122
6.22.2 Member Function Documentation . . . 122
6.22.2.1 run_once . . . 122
6.22.3 Member Data Documentation . . . 122
6.22.3.1 preserve_srcdir . . . 122
6.22.3.2 version . . . 122
6.23 hardware_MemoryThroughput.hardware_MemoryThroughput Class Reference . . . 124
6.23.1 Detailed Description . . . 124
6.23.2 Member Function Documentation . . . 124
6.23.2.1 run_once . . . 124
6.23.2.2 setup . . . 125
6.23.3 Member Data Documentation . . . 125
6.23.3.1 preserve_srcdir . . . 125
6.23.3.2 results . . . 125
6.23.3.3 version . . . 125
6.24 hardware_MemoryTotalSize.hardware_MemoryTotalSize Class Reference . . . 126

6.24.1 Detailed Description . . . 126
6.24.2 Member Function Documentation . . . 126
6.24.2.1 run_once . . . 126
6.24.3 Member Data Documentation . . . 126
6.24.3.1 version . . . 126
6.25 hardware_Resolution.hardware_Resolution Class Reference . . . 127
6.25.1 Detailed Description . . . 127
6.25.2 Member Function Documentation . . . 127
6.25.2.1 get_resolution . . . 127
6.25.2.2 run_once . . . 128
6.25.3 Member Data Documentation . . . 128
6.25.3.1 version . . . 128
6.26 hardware_SAT.hardware_SAT Class Reference . . . 129
6.26.1 Detailed Description . . . 129
6.26.2 Member Function Documentation . . . 129
6.26.2.1 run_once . . . 129
6.26.2.2 setup . . . 130
6.26.3 Member Data Documentation . . . 130
6.26.3.1 version . . . 130
6.27 hardware_SsdDetection.hardware_SsdDetection Class Reference . . . 131
6.27.1 Detailed Description . . . 131
6.27.2 Member Function Documentation . . . 131
6.27.2.1 run_once . . . 131
6.27.2.2 setup . . . 131
6.27.3 Member Data Documentation . . . 132
6.27.3.1 version . . . 132
6.28 hardware_StorageFio.hardware_StorageFio Class Reference . . . 133
6.28.1 Detailed Description . . . 133
6.28.2 Member Function Documentation . . . 133
6.28.2.1 __find_free_root_partition . . . 133
6.28.2.2 __get_device_description . . . 134
6.28.2.3 __get_file_size . . . 134
6.28.2.4 __parse_fio . . . 134
6.28.2.5 __RunFio . . . 135
6.28.2.6 initialize . . . 136
6.28.2.7 run_once . . . 136

6.28.2.8 setup . . . 137
6.28.3 Member Data Documentation . . . 137
6.28.3.1 __description . . . 137
6.28.3.2 __filename . . . 137
6.28.3.3 __filesize . . . 137
6.28.3.4 version . . . 137
6.29 hardware_Touchpad.hardware_Touchpad Class Reference . . . 139
6.29.1 Detailed Description . . . 139
6.29.2 Member Function Documentation . . . 139
6.29.2.1 run_once . . . 139
6.29.3 Member Data Documentation . . . 139
6.29.3.1 preserve_srcdir . . . 139
6.29.3.2 version . . . 139
6.30 hardware_VideoOutSemiAuto.hardware_VideoOutSemiAuto Class Reference . . . 140
6.30.1 Detailed Description . . . 140
6.30.2 Member Function Documentation . . . 140
6.30.2.1 __configure_and_check_output . . . 140
6.30.2.2 __output_connected . . . 141
6.30.2.3 __output_is_set . . . 141
6.30.2.4 __query_for_output . . . 141
6.30.2.5 run_once . . . 142
6.30.3 Member Data Documentation . . . 142
6.30.3.1 HDMI_ID . . . 142
6.30.3.2 RECONFIG_PATH . . . 142
6.30.3.3 version . . . 143
6.30.3.4 VGA_ID . . . 143
6.30.3.5 XRANDR_PATH . . . 143
6.31 ltp.ltp Class Reference . . . 144
6.31.1 Detailed Description . . . 144
6.31.2 Member Function Documentation . . . 144
6.31.2.1 _import_site_config . . . 144
6.31.2.2 initialize . . . 144
6.31.2.3 run_once . . . 145
6.31.2.4 setup . . . 145
6.31.3 Member Data Documentation . . . 146
6.31.3.1 site_ignore_tests . . . 146

6.31.3.2 version . . . 146
6.32 network_DisableInterface.network_DisableInterface Class Reference . . . 147
6.32.1 Detailed Description . . . 147
6.32.2 Member Function Documentation . . . 147
6.32.2.1 is_iface_up . . . 147
6.32.2.2 run_once . . . 147
6.32.3 Member Data Documentation . . . 148
6.32.3.1 _ifconfig . . . 148
6.32.3.2 version . . . 148
6.33 network_WiFiCaps.network_WiFiCaps Class Reference . . . 149
6.33.1 Detailed Description . . . 149
6.33.2 Member Function Documentation . . . 149
6.33.2.1 __parse_iwcap . . . 149
6.33.2.2 __run_iwcap . . . 149
6.33.2.3 run_once . . . 150
6.33.2.4 setup . . . 150
6.33.3 Member Data Documentation . . . 150
6.33.3.1 version . . . 150
6.34 platform_AesThroughput.platform_AesThroughput Class Reference . . . 151
6.34.1 Detailed Description . . . 151
6.34.2 Member Function Documentation . . . 151
6.34.2.1 export_stats . . . 151
6.34.2.2 openssl_speed . . . 151
6.34.2.3 parse_results . . . 152
6.34.2.4 run_once . . . 152
6.34.2.5 setup . . . 152
6.34.2.6 update_stats . . . 152
6.34.3 Member Data Documentation . . . 153
6.34.3.1 results . . . 153
6.34.3.2 version . . . 153
6.35 platform_BootPerf.platform_BootPerf Class Reference . . . 154
6.35.1 Detailed Description . . . 154
6.35.2 Member Function Documentation . . . 154
6.35.2.1 __parse_disk_login_prompt_ready . . . 154
6.35.2.2 __parse_firmware_boot_time . . . 154
6.35.2.3 __parse_syslog . . . 155

6.35.2.4 __parse_uptime . . . 156
6.35.2.5 run_once . . . 156
6.35.3 Member Data Documentation . . . 156
6.35.3.1 version . . . 156
6.36 platform_KernelVersion.platform_KernelVersion Class Reference . . . 157
6.36.1 Detailed Description . . . 157
6.36.2 Member Function Documentation . . . 157
6.36.2.1 run_once . . . 157
6.36.3 Member Data Documentation . . . 157
6.36.3.1 version . . . 157
6.37 power_BatteryCharge.power_BatteryCharge Class Reference . . . 158
6.37.1 Detailed Description . . . 158
6.37.2 Member Function Documentation . . . 158
6.37.2.1 initialize . . . 158
6.37.2.2 on_ac . . . 158
6.37.2.3 postprocess_iteration . . . 159
6.37.2.4 run_once . . . 159
6.37.3 Member Data Documentation . . . 160
6.37.3.1 charge_full_design . . . 160
6.37.3.2 initial_charge . . . 160
6.37.3.3 max_run_time . . . 160
6.37.3.4 remaining_time . . . 160
6.37.3.5 status . . . 160
6.37.3.6 version . . . 160
6.38 power_CPUFreq.power_CPUFreq Class Reference . . . 162
6.38.1 Detailed Description . . . 162
6.38.2 Member Function Documentation . . . 162
6.38.2.1 run_once . . . 162
6.38.3 Member Data Documentation . . . 163
6.38.3.1 version . . . 163
6.39 power_CPUIdle.power_CPUIdle Class Reference . . . 164
6.39.1 Detailed Description . . . 164
6.39.2 Member Function Documentation . . . 164
6.39.2.1 run_once . . . 164
6.39.3 Member Data Documentation . . . 164
6.39.3.1 version . . . 164

6.40 power_Draw.power_Draw Class Reference . . . 165
6.40.1 Detailed Description . . . 165
6.40.2 Member Function Documentation . . . 165
6.40.2.1 run_once . . . 165
6.40.3 Member Data Documentation . . . 165
6.40.3.1 version . . . 165
6.41 power_LoadTest.power_LoadTest Class Reference . . . 166
6.41.1 Detailed Description . . . 166
6.41.2 Member Function Documentation . . . 167
6.41.2.1 _do_wait . . . 167
6.41.2.2 _is_network_iface_running . . . 167
6.41.2.3 _percent_current_charge . . . 168
6.41.2.4 _write_ext_params . . . 168
6.41.2.5 cleanup . . . 168
6.41.2.6 postprocess_iteration . . . 168
6.41.2.7 run_once . . . 169
6.41.2.8 setup . . . 171
6.41.3 Member Data Documentation . . . 172
6.41.3.1 _ah_charge_start . . . 172
6.41.3.2 _cpufreq_stats . . . 172
6.41.3.3 _cpuidle_stats . . . 172
6.41.3.4 _ext_path . . . 172
6.41.3.5 _loop_count . . . 172
6.41.3.6 _loop_time . . . 172
6.41.3.7 _low_battery_threshold . . . 172
6.41.3.8 _mseconds . . . 172
6.41.3.9 _power_status . . . 172
6.41.3.10 _scroll_by_pixels . . . 172
6.41.3.11 _scroll_interval_ms . . . 172
6.41.3.12 _scroll_loop . . . 173
6.41.3.13 _should_scroll . . . 173
6.41.3.14 _should_scroll_up . . . 173
6.41.3.15 _testServer . . . 173
6.41.3.16 _tmp_keyvals . . . 173
6.41.3.17 _usb_stats . . . 173
6.41.3.18 _verbose . . . 173

6.41.3.19 _wh_energy_start . . . 173
6.41.3.20 version . . . 173
6.42 power_Resume.power_Resume Class Reference . . . 174
6.42.1 Detailed Description . . . 174
6.42.2 Member Function Documentation . . . 174
6.42.2.1 __get_end_suspend_time . . . 174
6.42.2.2 __get_last_msg_time . . . 174
6.42.2.3 __get_start_suspend_time . . . 175
6.42.2.4 __is_iface_up . . . 175
6.42.2.5 __sanity_check_system . . . 175
6.42.2.6 run_once . . . 175
6.42.3 Member Data Documentation . . . 176
6.42.3.1 preserve_srcdir . . . 176
6.42.3.2 version . . . 176
6.43 realtimecomm_GTalkAudioPlayground.realtimecomm_GTalkAudioPlayground Class Reference . . . 177
6.43.1 Detailed Description . . . 177
6.43.2 Member Function Documentation . . . 177
6.43.2.1 run_once . . . 177
6.43.2.2 run_verification . . . 178
6.43.2.3 setup . . . 178
6.43.3 Member Data Documentation . . . 178
6.43.3.1 dep . . . 178
6.43.3.2 dep_dir . . . 179
6.43.3.3 performance_results . . . 179
6.43.3.4 playground . . . 179
6.43.3.5 version . . . 179
6.44 realtimecomm_GTalkPlayground.realtimecomm_GTalkPlayground Class Reference . . . 180
6.44.1 Detailed Description . . . 180
6.44.2 Member Function Documentation . . . 180
6.44.2.1 get_framerate . . . 180
6.44.2.2 run_once . . . 181
6.44.2.3 run_verification . . . 182
6.44.2.4 setup . . . 182
6.44.3 Member Data Documentation . . . 182
6.44.3.1 dep . . . 182
6.44.3.2 dep_dir . . . 182

6.44.3.3 performance_results . . . 182
6.44.3.4 playground . . . 183
6.44.3.5 version . . . 183
6.45 unixbench.unixbench Class Reference . . . 184
6.45.1 Detailed Description . . . 184
6.45.2 Member Function Documentation . . . 184
6.45.2.1 check_for_error . . . 184
6.45.2.2 cleanup . . . 184
6.45.2.3 initialize . . . 185
6.45.2.4 postprocess_iteration . . . 185
6.45.2.5 run_once . . . 186
6.45.2.6 setup . . . 187
6.45.3 Member Data Documentation . . . 187
6.45.3.1 err . . . 187
6.45.3.2 report_data . . . 187
6.45.3.3 version . . . 187

7 File Documentation **189**

7.1 testsource/audiovideo_FFMPEG.py File Reference . . . 189
7.2 testsource/audiovideo_V4L2.py File Reference . . . 190
7.3 testsource/compilebench.py File Reference . . . 191
7.4 testsource/desktopui_SunSpiderBench.py File Reference . . . 192
7.5 testsource/desktopui_V8Bench.py File Reference . . . 193
7.6 testsource/disktest.py File Reference . . . 194
7.7 testsource/firmware_RomSize.py File Reference . . . 195
7.8 testsource/firmware_VbootCrypto.py File Reference . . . 196
7.9 testsource/gl_Bench.py File Reference . . . 197
7.10 testsource/graphics_GLAPICheck.py File Reference . . . 198
7.11 testsource/graphics_SanAngeles.py File Reference . . . 199
7.12 testsource/graphics_TearTest.py File Reference . . . 200
7.13 testsource/hardware_Backlight.py File Reference . . . 201
7.14 testsource/hardware_BluetoothSemiAuto.py File Reference . . . 202
7.15 testsource/hardware_Components.py File Reference . . . 203
7.16 testsource/hardware_DiskSize.py File Reference . . . 204
7.17 testsource/hardware_KeyboardAssembly.py File Reference . . . 205
7.18 testsource/hardware_MemoryThroughput.py File Reference . . . 206
7.19 testsource/hardware_MemoryTotalSize.py File Reference . . . 207

7.20 testsource/hardware_Resolution.py File Reference 208
7.21 testsource/hardware_SAT.py File Reference 209
7.22 testsource/hardware_SsdDetection.py File Reference 210
7.23 testsource/hardware_StorageFio.py File Reference 211
7.24 testsource/hardware_Touchpad.py File Reference 212
7.25 testsource/hardware_VideoOutSemiAuto.py File Reference 213
7.26 testsource/ltp.py File Reference 214
7.27 testsource/mainpage.txt File Reference 215
7.28 testsource/network_DisableInterface.py File Reference 216
7.29 testsource/network_WiFiCaps.py File Reference 217
7.30 testsource/platform_AesThroughput.py File Reference 218
7.31 testsource/platform_BootPerf.py File Reference 219
7.32 testsource/platform_KernelVersion.py File Reference 220
7.33 testsource/power_BatteryCharge.py File Reference 221
7.34 testsource/power_CPUFreq.py File Reference 222
7.35 testsource/power_CPUIdle.py File Reference 223
7.36 testsource/power_Draw.py File Reference 224
7.37 testsource/power_LoadTest.py File Reference 225
7.38 testsource/power_Resume.py File Reference 226
7.39 testsource/realtimecomm_GTalkAudioPlayground.py File Reference 227
7.40 testsource/realtimecomm_GTalkPlayground.py File Reference 228
7.41 testsource/unixbench.py File Reference 229

Chapter 1

Main Page

1.1 Glossary

- $: command line prompt
- $HOME: home directory of current user.
- AC: alternating current, implies device is not powered by battery.
- DUT: device under test
- Semi-Automated Test: test that runs with automation but requires manual intervention.

1.2 Test Setup

- Setup a Linux machine to serve as the Autotest server. The Autotest server requires Python, Wireless access to the DUT and basic Linux shell utilities. The setup has been tested on Ubuntu 9.10 available for download at http://www.ubuntu.com/getubuntu/download/.
- Create an installation directory on the Autotest server for the Chrome OS hardware qualification package. The rest of the instructions assume that you're installing the package in the current user home directory ($HOME/).
- Contact your Google technical support person and download the Chrome OS hardware qualification package chromeos-hwqual-TAG.tar.bz2 for your device in $HOME/.
- Install the package on the server:

```
$ cd $HOME/ && tar xjf chromeos-hwqual-TAG.tar.bz2
```

- Install the Chrome OS test image on the DUT. The USB test image is available in:

$HOME/chromeos-hwqual-TAG/chromeos-hwqual-usb.img

Here are sample steps to install the test image.

- Plug a USB storage device into the Autotest server. Note that all data on your USB stick will be destroyed.

- Unmount any mounts on the USB device.
- Copy the USB image to a USB storage device by executing:

```
$ sudo dd if=$HOME/chromeos-hwqual-TAG/chromeos-hwqual-usb.img of=/dev/sdX
```

- where /dev/sdX is your USB device.
- Plug the USB device into the DUT and boot from it.
- Install Chrome OS on the DUT: switch to VT2 by pressing Ctrl-Alt-F2, login as "chronos", password "test0000", and run

```
$ /usr/sbin/chromeos-install
```

- Cold boot the DUT -- turn the DUT off and then back on. This ensures a consistent starting point for the qualification tests and allows the system to collect cold boot performance metrics. Make sure you don't boot from USB.
- Connect the DUT to the network and note its IP address <DUT_IP>. The IP address is displayed at the bottom of the network selection menu. Unless specified explicitly, the test setup works correctly through either wireless or wired network connections.
- Add the DUT root private key to ssh-agent on the Autotest server:

```
$ ssh-add $HOME/chromeos-hwqual-TAG/testing_rsa
```

- If ssh-add fails saying that it cannot connect to your authentication agent, retry the command after running:

```
$ eval `ssh-agent -s`
```

- These commands allow the Autotest server to connect and login as root on the DUT.
- Make sure you can ssh as root to the DUT from the Autotest server. The command below should print 0.

```
$ ssh root@<DUT_IP> true; echo $?
```

1.3 Automated and Semi-Automated Test Runs

- Unless otherwise noted, all tests can be performed on an AC-powered DUT.
- Go to the Autotest server directory:

```
$ cd $HOME/chromeos-hwqual-TAG/autotest/
```

- Autotest logs progress and performance data in results.* directories specified through the '-r' autoserv option (see below). A quick way to review the test results is to use the 'generate_test_report' script installed under $HOME/chromeos-hwqual-TAG/:

```
$ ../generate_test_report results.*
```

- This will display a table with test status and perfomance data for all result directories.
- Alternatively, you can setup the Autotest's web server and database components and use them to view test results and maintain history.
- If deeper investigation into a failure is required, you can review the debug information stored in results:

```
$ */*/debug/.
```

- Before running the tests, cleanup previous test results:

```
$ rm -rf results.*
```

- Run the fully automated client-side tests:

```
$ ./server/autoserv -r results.auto -m <DUT_IP> -c client/site_tests/suite_HWQual/control.auto
```

- Plug high-speed high-capacity storage devices in all USB and SD Card slots and run the external storage test:

```
$ ./server/autoserv -r results.external_devices -m <DUT_IP> -c client/site_tests/suite_HWQual/control.exte
```

- Run the approved components test by first following the manual instructions specified in the control file (control.components) and then executing:

```
$ ./server/autoserv -r results.components -m <DUT_IP> -c client/site_tests/suite_HWQual/control.components
```

- Run the system suspend/resume stability test:

```
$ ./server/autoserv -r results.suspend_resume -m <DUT_IP> -c client/site_tests/suite_HWQual/control.suspen
```

- Run the keyboard semi-automated test by first reading the instructions specified in the control file (control.keyboard) and then executing:

```
$ ./server/autoserv -r results.keyboard -m <DUT_IP> -a hwqual -c client/site_tests/suite_HWQual/control.ke
```

- Run the touchpad semi-automated test by first reading the instructions specified in the control file (control.touchpad) and then executing:

```
$ ./server/autoserv -r results.touchpad -m <DUT_IP> -a hwqual -c client/site_tests/suite_HWQual/control.to
```

- If the DUT has a Bluetooth adapter, run the Bluetooth semi-automated tests by following the instructions specified in the control file (control.bluetooth) and then executing:

```
$ ./server/autoserv -r results.bluetooth -m <DUT_IP> -c client/site_tests/suite_HWQual/control.bluetooth
```

- If the DUT has video out ports, run the Video Out semi-automated test by following the instructions specified in the control file (control.video_out) and then executing:

```
$ ./server/autoserv -r results.video_out.${PORT} -m <DUT_IP> -c client/site_tests/suite_HWQual/control.vid
```

- Where PORT is the name of each video port you are testing. For example, if the DUT has one HDMI and one VGA out port, run:

```
$ ./server/autoserv -r results.video_out.hdmi1 -m <DUT_IP> -c client/site_tests/suite_HWQual/control.video
```

```
$ ./server/autoserv -r results.video_out.vga1 -m <DUT_IP> -c client/site_tests/suite_HWQual/control.video_
```

- Run the graphics tearing test:

```
$ ./server/autoserv -r results.teartest -m <DUT_IP> -c client/site_tests/suite_HWQual/control.teartest
```

- Run the device on AC. Plug a power draw USB dongle in each USB port. Run the max power draw test:

```
$ ./server/autoserv -r results.max_power_draw.ac -m <DUT_IP> -c client/site_tests/suite_HWQual/control.max
```

- Run the device on battery. Plug a power draw USB dongle in each USB port. Run the max power draw test:

```
$ ./server/autoserv -r results.max_power_draw.batt -m <DUT_IP> -c client/site_tests/suite_HWQual/control.m
```

- Make sure the remaining battery charge is less than 5%. Run the DUT on AC. Run the battery charge test:

```
$ ./server/autoserv -r results.battery_charge_time -m <DUT_IP> -c client/site_tests/suite_HWQual/control.b
```

- Note that the test will check and fail quickly if the initial battery charge is more than 5%.

- Make sure that the battery is fully charged. Run the DUT on battery. Run the battery load test by first following the manual instructions specified in the control file (control.battery_load) and then executing:

```
$ ./server/autoserv -r results.battery_load -m <DUT_IP> -c client/site_tests/suite_HWQual/control.battery_
```

- Note that the test will not check if the battery is fully charged before running.

1.4 Manual Test Runs

- Perform the manual tests specified in $HOME/chromeos-hwqual-TAG/manual/testcases.csv.

- Please note that some tests cannot be tested as they rely on features not yet implemented. They are being included as a preview for manual tests that will be required. Such tests will have "NotImplemented" in their "LABELS" column.

1.5 Reporting Results

- Make sure that there are no test failures in automatic, semi-automatic, or manual test categories.
- Once all tests pass or if you need technical assistance with the hardware qualification results, package the result directories:

```
$ tar cjf chromeos-hwqual-TAG-DATE.tar.bz2 results.*
```

- Send the tarball to your Google technical support contact for review or analysis.

Chapter 2

Namespace Index

2.1 Package List

Here are the packages with brief descriptions (if available):

audiovideo_FFMPEG (This test exercises the ffmpeg-based software video decoder) 13
audiovideo_V4L2 (Exercises v4l2 camera devices to verify required operations) 15
compilebench (Benchmark the filesystem performance) 16
desktopui_SunSpiderBench (Measure the performance of Chrome's JavaScript) 17
desktopui_V8Bench (Benchmark javascript operations of a web browser) 18
disktest (Verify the integrity of the disk and disk controller) 19
firmware_RomSize (Ensure the firmware size is large enough) 20
firmware_VbootCrypto (Verifies Firmware Verified Boot Reference Implementation, it's components, and crypto performance) 21
gl_Bench (Benchmark the graphics library performance) 22
graphics_GLAPICheck (Verify correctness of OpenGL/GLES and X11 versions/extensions) . . 23
graphics_SanAngeles (Benchmark OpenGL object rendering) 24
graphics_TearTest (Verify Chrome does not tear with veritcal synchronization) 25
hardware_Backlight (Verify that the backlight can be adjusted in software) 27
hardware_BluetoothSemiAuto (Verify the basic functionality of the Bluetooth adapter) 28
hardware_Components (Ensure system components are in the approved components database) . 30
hardware_DiskSize (Ensure the hard disk is large enough) 31
hardware_KeyboardAssembly (Verify that keyboard keys function properly) 32
hardware_MemoryThroughput (Benchmark sequential and random access mode memory throughput) 33
hardware_MemoryTotalSize (Verify there is enough memory to run Chrome OS) 34
hardware_Resolution (Determine if the current screen resolution is supported) 35
hardware_SAT (Stress test hardware devices) 36
hardware_SsdDetection (Determine if main disk is a solid state device) 37
hardware_StorageFio (Benchmark storage performance using an unmounted root partition) . . . 38
hardware_Touchpad (Verify all touchpad functions) 39
hardware_VideoOutSemiAuto (Verify external video ports are configurable with Chrome OS) . 40
ltp (Verify kernel system calls are operating correctly) 41
network_DisableInterface (Verify a network interface can be disabled) 42
network_WiFiCaps (Verify that WiFi devices have the required capabilities) 43
platform_AesThroughput (Benchmark processor performance using OpenSSL using AES options) 44
platform_BootPerf (Collect boot performance metrics from the last system reboot) 45

platform_KernelVersion (Ensure the running kernel is supported) 46
power_BatteryCharge (Measure the time required to charge the battery) 47
power_CPUFreq (Verify that supported CPU frequencies can be set) 48
power_CPUIdle (Ensure the processor drops into idle state when it is idle) 49
power_Draw (Measure how much power is drawn over a given amount of time) 50
power_LoadTest (Measure power draw when system is under load) 51
power_Resume (Measure the amount of time it takes to resume from suspend) 53
realtimecomm_GTalkAudioPlayground (Verify that Google Talk Plugin Audio works with Chrome) 54
realtimecomm_GTalkPlayground (Verify that Google Talk Plugin executes) 56
unixbench (Measure system level performance) 58

Chapter 3

Class Index

3.1 Class List

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

hardware_BluetoothSemiAuto.Agent . . . 59
audiovideo_FFMPEG.audiovideo_FFMPEG . . . 61
audiovideo_V4L2.audiovideo_V4L2 . . . 64
compilebench.compilebench . . . 70
power_CPUFreq.cpufreq . . . 73
power_CPUIdle.cpuidle . . . 77
power_CPUIdle.cpuidle_state . . . 79
power_CPUIdle.cpus . . . 81
desktopui_SunSpiderBench.desktopui_SunSpiderBench . . . 83
desktopui_V8Bench.desktopui_V8Bench . . . 85
disktest.disktest . . . 87
firmware_RomSize.firmware_RomSize . . . 90
firmware_VbootCrypto.firmware_VbootCrypto . . . 91
gl_Bench.gl_Bench . . . 98
graphics_GLAPICheck.graphics_GLAPICheck . . . 100
graphics_SanAngeles.graphics_SanAngeles . . . 105
graphics_TearTest.graphics_TearTest . . . 107
hardware_Backlight.hardware_Backlight . . . 109
hardware_BluetoothSemiAuto.hardware_BluetoothSemiAuto . . . 110
hardware_Components.hardware_Components . . . 114
hardware_DiskSize.hardware_DiskSize . . . 121
hardware_KeyboardAssembly.hardware_KeyboardAssembly . . . 122
hardware_MemoryThroughput.hardware_MemoryThroughput . . . 124
hardware_MemoryTotalSize.hardware_MemoryTotalSize . . . 126
hardware_Resolution.hardware_Resolution . . . 127
hardware_SAT.hardware_SAT . . . 129
hardware_SsdDetection.hardware_SsdDetection . . . 131
hardware_StorageFio.hardware_StorageFio . . . 133
hardware_Touchpad.hardware_Touchpad . . . 139
hardware_VideoOutSemiAuto.hardware_VideoOutSemiAuto . . . 140
ltp.ltp . . . 144
network_DisableInterface.network_DisableInterface . . . 147
network_WiFiCaps.network_WiFiCaps . . . 149

platform_AesThroughput.platform_AesThroughput 151
platform_BootPerf.platform_BootPerf 154
platform_KernelVersion.platform_KernelVersion 157
power_BatteryCharge.power_BatteryCharge 158
power_CPUFreq.power_CPUFreq 162
power_CPUIdle.power_CPUIdle 164
power_Draw.power_Draw 165
power_LoadTest.power_LoadTest 166
power_Resume.power_Resume 174
realtimecomm_GTalkAudioPlayground.realtimecomm_GTalkAudioPlayground 177
realtimecomm_GTalkPlayground.realtimecomm_GTalkPlayground 180
unixbench.unixbench 184

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

testsource/audiovideo_FFMPEG.py . . . 189
testsource/audiovideo_V4L2.py . . . 190
testsource/compilebench.py . . . 191
testsource/desktopui_SunSpiderBench.py . . . 192
testsource/desktopui_V8Bench.py . . . 193
testsource/disktest.py . . . 194
testsource/firmware_RomSize.py . . . 195
testsource/firmware_VbootCrypto.py . . . 196
testsource/gl_Bench.py . . . 197
testsource/graphics_GLAPICheck.py . . . 198
testsource/graphics_SanAngeles.py . . . 199
testsource/graphics_TearTest.py . . . 200
testsource/hardware_Backlight.py . . . 201
testsource/hardware_BluetoothSemiAuto.py . . . 202
testsource/hardware_Components.py . . . 203
testsource/hardware_DiskSize.py . . . 204
testsource/hardware_KeyboardAssembly.py . . . 205
testsource/hardware_MemoryThroughput.py . . . 206
testsource/hardware_MemoryTotalSize.py . . . 207
testsource/hardware_Resolution.py . . . 208
testsource/hardware_SAT.py . . . 209
testsource/hardware_SsdDetection.py . . . 210
testsource/hardware_StorageFio.py . . . 211
testsource/hardware_Touchpad.py . . . 212
testsource/hardware_VideoOutSemiAuto.py . . . 213
testsource/ltp.py . . . 214
testsource/network_DisableInterface.py . . . 216
testsource/network_WiFiCaps.py . . . 217
testsource/platform_AesThroughput.py . . . 218
testsource/platform_BootPerf.py . . . 219
testsource/platform_KernelVersion.py . . . 220
testsource/power_BatteryCharge.py . . . 221
testsource/power_CPUFreq.py . . . 222

testsource/power_CPUIdle.py 223
testsource/power_Draw.py 224
testsource/power_LoadTest.py 225
testsource/power_Resume.py 226
testsource/realtimecomm_GTalkAudioPlayground.py 227
testsource/realtimecomm_GTalkPlayground.py 228
testsource/unixbench.py 229

Chapter 5

Namespace Documentation

5.1 Package audiovideo_FFMPEG

This test exercises the ffmpeg-based software video decoder.

Classes

- class audiovideo_FFMPEG

5.1.1 Detailed Description

This test exercises the ffmpeg-based software video decoder.

Pass/Fail Criteria:

This test is a benchmark. Failure criteria may be added by constraints in the suite control file.

Arguments:

- fps_video_min >= 40.0
- tpf_audio_max <= 4.0

Test Duration

LONG

Category

FFMPEG

Test Type

client

Test Class

audiovideo

5.2 Package audiovideo_V4L2

Exercises v4l2 camera devices to verify required operations.

Classes

- class audiovideo_V4L2

5.2.1 Detailed Description

Exercises v4l2 camera devices to verify required operations.

Pass/Fail Criteria:

This is a complex text and tests many different functions. This test will fail if any of the following conditions occur:

- No v4L2 device is found
- If a mandatory control is not supported
- If streaming is not supported
- If a required resolution is not supported when capturing a stream

Arguments:

- time: run_time
- run_capture_tests: False
- run_default_capture_test: True
- assert_mandatory_controls: False
- tag: i

Test Duration

MEDIUM

Category

V4L2

Test Type

client

Test Class

audiovideo

5.3 Package compilebench

Benchmark the filesystem performance.

Classes

- class compilebench

Variables

- string test_name = 'compilebench'

5.3.1 Detailed Description

Benchmark the filesystem performance.

Pass/Fail Criteria:

Compilebench is a filesystem performance test.

Test Duration

SHORT

Category

Functional

Test Type

client

Test Class

General

5.3.2 Variable Documentation

5.3.2.1 string compilebench.test_name = 'compilebench'

Definition at line 5 of file compilebench.py.

5.4 Package desktopui_SunSpiderBench

Measure the performance of Chrome's JavaScript.

Classes

- class desktopui_SunSpiderBench

5.4.1 Detailed Description

Measure the performance of Chrome's JavaScript.

Pass/Fail Criteria:

This test is a benchmark.

Test Duration

SHORT

Category

Benchmark

Test Type

client

Test Class

desktopui

5.5 Package desktopui_V8Bench

Benchmark javascript operations of a web browser.

Classes

- class desktopui_V8Bench

5.5.1 Detailed Description

Benchmark javascript operations of a web browser.

Pass/Fail Criteria:

The test will fail if it does not complete the benchmark. Additional contraints are needed if a minimum score is required.

Test Duration

SHORT

Category

Benchmark

Test Type

client

Test Class

desktopui

5.6 Package disktest

Verify the integrity of the disk and disk controller.

Classes

- class disktest

5.6.1 Detailed Description

Verify the integrity of the disk and disk controller.

Pass/Fail Criteria:

This test reads back all data written during the test, and if the data it reads does not match the data it wrote, the test fails.

Test Duration

MEDIUM

Category

Kernel

Test Type

client

Test Class

Hardware

5.7 Package firmware_RomSize

Ensure the firmware size is large enough.

Classes

- class firmware_RomSize

5.7.1 Detailed Description

Ensure the firmware size is large enough.

Pass/Fail Criteria:

Firmware size must be >= 2MB.

Arguments:

- kb_system_rom_size >= 2048

Test Duration

SHORT

Category

Functional

Test Type

client

Test Class

firmware

5.8 Package firmware_VbootCrypto

Verifies Firmware Verified Boot Reference Implementation, it's components, and crypto performance.

Classes

- class firmware_VbootCrypto

5.8.1 Detailed Description

Verifies Firmware Verified Boot Reference Implementation, it's components, and crypto performance.

Pass/Fail Criteria:

This test is a benchmark.

Errors in any of the following tests will cause a failure:

- _sha_test()
- _rsa_test()
- _image_verification_test()
- _rollbackA_tests()
- _splicing_tests()

Arguments:

- suite: benchmarks
- tag: benchmarks

Test Duration

LONG

Category

Functional

Test Type

client

Test Class

firmware

5.9 Package gl_Bench

Benchmark the graphics library performance.

Classes

- class gl_Bench

5.9.1 Detailed Description

Benchmark the graphics library performance.

Pass/Fail Criteria:

Various performance metrics must be met or exceeded or the test will fail. See constraints for triangle setup, solid fill, tex nearest fill, and bilinear tex fill.

Arguments:

- mtri_sec_triangle_setup >= 10
- mpixels_sec_fill_solid >= 190
- mpixels_sec_fill_tex_nearest >= 190
- mpixels_sec_fill_tex_bilinear >= 190
- options: - -t fill -t triangle -d d
- tag: tag

Test Duration

MEDIUM

Category

Performance

Test Type

client

Test Class

gl

5.10 Package graphics_GLAPICheck

Verify correctness of OpenGL/GLES and X11 versions/extensions.

Classes

- class graphics_GLAPICheck

5.10.1 Detailed Description

Verify correctness of OpenGL/GLES and X11 versions/extensions.

Pass/Fail Criteria:

This test will fail if:

- GL version is less than 1.4
- GL extension is less than 2
- GLES version is less than 2
- EGL version is less than 1.3
- If GL extensions don't include needed extensions
- If X extensions don't include DAMAGE and Composite

Test Duration

SHORT

Category

Performance

Test Type

client

Test Class

graphics

5.11 Package graphics_SanAngeles

Benchmark OpenGL object rendering.

Classes

- class graphics_SanAngeles

5.11.1 Detailed Description

Benchmark OpenGL object rendering.

Pass/Fail Criteria:

This test is a benchmark. It will fail if it fails to complete.

Arguments:

- iterations: 2
- tag: _tag

Test Duration

MEDIUM

Category

Performance

Test Type

client

Test Class

graphics

5.12 Package graphics_TearTest

Verify Chrome does not tear with veritcal synchronization.

Classes

- class graphics_TearTest

Functions

- def html_button

Variables

- string TEMPLATE

5.12.1 Detailed Description

Verify Chrome does not tear with veritcal synchronization.

Pass/Fail Criteria:

This test will fail if there is tearing in the two vertical lines that are scrolling horizontally.

Test Duration

MEDIUM

Category

Functional

Test Type

client

Test Class

graphics

5.12.2 Function Documentation

5.12.2.1 def graphics_TearTest.html_button (*label*, *onclick* = `None`)

Definition at line 11 of file graphics_TearTest.py.

```
12                                      :
13   return ('''<input type="button" value="%s" onclick="do_submit('%s')"/>''' %
14           (label, onclick if onclick else label))
```

5.12.3 Variable Documentation

5.12.3.1 string graphics_TearTest.TEMPLATE

Initial value:

```
'''
<h5>{0}</h5>
<table>
<tr> <td>{1[0][desc]}</td> <td>{1[0][result]}</td> </tr>
<tr> <td>{1[1][desc]}</td> <td>{1[1][result]}</td> </tr>
<tr> <td>{1[2][desc]}</td> <td>{1[2][result]}</td> </tr>
</table>
'''
```

Definition at line 15 of file graphics_TearTest.py.

5.13 Package hardware_Backlight

Verify that the backlight can be adjusted in software.

Classes

- class hardware_Backlight

Functions

- def backlight_tool

5.13.1 Detailed Description

Verify that the backlight can be adjusted in software.

Pass/Fail Criteria:

If backlight-tool fails to adjust the backlight brightness this test will fail.

Test Duration

SHORT

Category

Functional

Test Type

client

Test Class

hardware

5.13.2 Function Documentation

5.13.2.1 def hardware_Backlight.backlight_tool (*args*)

Definition at line 9 of file hardware_Backlight.py.

```
10                          :
11     cmd = 'backlight-tool %s' % args
12     return utils.system_output(cmd)
```

5.14 Package hardware_BluetoothSemiAuto

Verify the basic functionality of the Bluetooth adapter.

Classes

- class Agent
- class hardware_BluetoothSemiAuto

Variables

- string _QUESTION_START
- string _HREF_START = "'"'
- string _HREF_END = "'"'

5.14.1 Detailed Description

Verify the basic functionality of the Bluetooth adapter.

Pass/Fail Criteria:

If no Bluetooth devices are found, or the user does not input an answer to the dialog box this test will fail.

Test Duration

SHORT

Category

Functional

Test Type

client

Test Class

Hardware

5.14.2 Variable Documentation

5.14.2.1 string hardware_BluetoothSemiAuto._HREF_END = "'"'

Definition at line 21 of file hardware_BluetoothSemiAuto.py.

5.14.2.2 string hardware_BluetoothSemiAuto._HREF_START = '''<a href="#" onclick="do_submit('%s')">'''

Definition at line 20 of file hardware_BluetoothSemiAuto.py.

5.14.2.3 string hardware_BluetoothSemiAuto._QUESTION_START

Initial value:

```
'''
<h5>
The Bluetooth scan discovered the following devices.<br>
If a device is not on the list, switch it into pairing mode and rescan.<br>
<br>
You can click on an input device (e.g., mouse) to pair with it.<br>
</h5>
<table border="1"><tr><td>Address</td><td>Name</td></tr>
'''
```

Definition at line 10 of file hardware_BluetoothSemiAuto.py.

5.15 Package hardware_Components

Ensure system components are in the approved components database.

Classes

- class hardware_Components

5.15.1 Detailed Description

Ensure system components are in the approved components database.

Pass/Fail Criteria:

If system components don't match those from the approved component database this test will fail.

Test Duration

SHORT

Category

Functional

Test Type

client

Test Class

hardware

5.16 Package hardware_DiskSize

Ensure the hard disk is large enough.

Classes

- class hardware_DiskSize

5.16.1 Detailed Description

Ensure the hard disk is large enough.

Pass/Fail Criteria:

Fails if the main disk is less than gb_main_disk_size.

Arguments:

- gb_main_disk_size >= 8

Test Duration

SHORT

Category

Functional

Test Type

client

Test Class

hardware

5.17 Package hardware_KeyboardAssembly

Verify that keyboard keys function properly.

Classes

- class hardware_KeyboardAssembly

5.17.1 Detailed Description

Verify that keyboard keys function properly.

Pass/Fail Criteria:

This test will fail if keys fail to respond when they are pressed and released. This test will also fail if all keys have not been used within 50 seconds.

Arguments:

- restart_ui: restart_ui

Test Duration

SHORT

Category

Factory

Test Type

client

Test Class

hardware

5.18 Package hardware_MemoryThroughput

Benchmark sequential and random access mode memory throughput.

Classes

- class hardware_MemoryThroughput

5.18.1 Detailed Description

Benchmark sequential and random access mode memory throughput.

Pass/Fail Criteria:

This test is a benchmark.

Arguments:

- num_iteration: 2500
- test_list: 21

Test Duration

MEDIUM

Category

Performance

Test Type

client

Test Class

hardware

5.19 Package hardware_MemoryTotalSize

Verify there is enough memory to run Chrome OS.

Classes

- class hardware_MemoryTotalSize

5.19.1 Detailed Description

Verify there is enough memory to run Chrome OS.

Pass/Fail Criteria:

Fails if memory size >= 1G.

Test Duration

SHORT

Category

Performance

Test Type

client

Test Class

hardware

5.20 Package hardware_Resolution

Determine if the current screen resolution is supported.

Classes

- class hardware_Resolution

Variables

- string __author__ = 'kdlucas@chromium.org (Kelly Lucas)'

5.20.1 Detailed Description

Determine if the current screen resolution is supported.

Pass/Fail Criteria:

Screen resolutions supported:

- 1280x800
- 1366x768

Test Duration

SHORT

Category

Functional

Test Type

client

Test Class

hardware

5.20.2 Variable Documentation

5.20.2.1 string hardware_Resolution.__author__ = 'kdlucas@chromium.org (Kelly Lucas)'

Definition at line 7 of file hardware_Resolution.py.

5.21 Package hardware_SAT

Stress test hardware devices.

Classes

- class hardware_SAT

5.21.1 Detailed Description

Stress test hardware devices.

Pass/Fail Criteria:

Fails if memory pages do not match the original fill pattern.

Arguments:

- seconds: 600
- seconds: _seconds
- tag: _tag
- seconds: seconds
- tag: tag

Test Duration

MEDIUM

Category

Stress

Test Type

client

Test Class

hardware

5.22 Package hardware_SsdDetection

Determine if main disk is a solid state device.

Classes

- class hardware_SsdDetection

5.22.1 Detailed Description

Determine if main disk is a solid state device.

Pass/Fail Criteria:

Fails if the main disk is not a solid state device.

Test Duration

SHORT

Category

Functional

Test Type

client

Test Class

hardware

5.23 Package hardware_StorageFio

Benchmark storage performance using an unmounted root partition.

Classes

- class hardware_StorageFio

5.23.1 Detailed Description

Benchmark storage performance using an unmounted root partition.

Pass/Fail Criteria:

This test is a benchmark.

Arguments:

- bytes_per_sec_seq_read >= 20 * 1024 * 1024
- bytes_per_sec_seq_write >= 15 * 1024 * 1024
- iops_4k_write >= 10
- dev: devpath
- tag: device

Test Duration

MEDIUM

Category

Performance

Test Type

client

Test Class

hardware

5.24 Package hardware_Touchpad

Verify all touchpad functions.

Classes

- class hardware_Touchpad

5.24.1 Detailed Description

Verify all touchpad functions.

Pass/Fail Criteria:

Test will fail if not all features are used in the alloted time.

Arguments:

- restart_ui: restart_ui

Test Duration

SHORT

Category

Factory

Test Type

client

Test Class

hardware

5.25 Package hardware_VideoOutSemiAuto

Verify external video ports are configurable with Chrome OS.

Classes

- class hardware_VideoOutSemiAuto

5.25.1 Detailed Description

Verify external video ports are configurable with Chrome OS.

Pass/Fail Criteria:

This test will fail under the following conditions:

- No external HDMI or VGA device is detected.
- monitor_reconfigure tool is not found
- xrandr is not found
- if the external video port is present but unable to be configured

Test Duration

SHORT

Category

Functional

Test Type

client

Test Class

hardware

5.26 Package ltp

Verify kernel system calls are operating correctly.

Classes

- class ltp

5.26.1 Detailed Description

Verify kernel system calls are operating correctly.

Pass/Fail Criteria:

There are over 1,000 tests and each one has it's own pass/failure criteria. Each individual test case is written from the system call's manpage.

Test Duration

MEDIUM

Category

FUNCTIONAL

Test Type

CLIENT

Test Class

KERNEL

5.27 Package network_DisableInterface

Verify a network interface can be disabled.

Classes

- class network_DisableInterface

5.27.1 Detailed Description

Verify a network interface can be disabled.

Pass/Fail Criteria:

Fails if the interface can not be disabled/reenabled.

Arguments:

- iface_name: wlan0
- tag: wlan0
- iface_name: eth0
- tag: eth0
- iface_name: hci0
- tag: hci0

Test Duration

SHORT

Category

Functional

Test Type

client

Test Class

network

5.28 Package network_WiFiCaps

Verify that WiFi devices have the required capabilities.

Classes

- class network_WiFiCaps

5.28.1 Detailed Description

Verify that WiFi devices have the required capabilities.

Pass/Fail Criteria:

If the following requirements are not present the test case will fail:

- station mode
- 2.4GHz band
- 5 GHz band
- 802.11n
- HT40
- Short GI in HT40

Test Duration

SHORT

Category

Functional

Test Type

client

Test Class

network

5.29 Package platform_AesThroughput

Benchmark processor performance using OpenSSL using AES options.

Classes

- class platform_AesThroughput

5.29.1 Detailed Description

Benchmark processor performance using OpenSSL using AES options.

Pass/Fail Criteria:

This test is a benchmark.

This test will fail if the output of openssl cannot be parsed.

Test Duration

MEDIUM

Category

Performance

Test Type

client

Test Class

platform

5.30 Package platform_BootPerf

Collect boot performance metrics from the last system reboot.

Classes

- class platform_BootPerf

5.30.1 Detailed Description

Collect boot performance metrics from the last system reboot.

Pass/Fail Criteria:

The test will fail if startup times or shut down times are exceeded. These values are set in the suite control file.

Arguments:

- seconds_firmware_boot <= 1.0
- seconds_power_on_to_login <= 5.0

Test Duration

SHORT

Category

Benchmark

Test Type

client

Test Class

platform

5.31 Package platform_KernelVersion

Ensure the running kernel is supported.

Classes

- class platform_KernelVersion

5.31.1 Detailed Description

Ensure the running kernel is supported.

Pass/Fail Criteria:

Fails if the running kernel version is older than kernel_version.

Test Duration

SHORT

Category

Functional

Test Type

client

Test Class

platform

5.32 Package power_BatteryCharge

Measure the time required to charge the battery.

Classes

- class power_BatteryCharge

5.32.1 Detailed Description

Measure the time required to charge the battery.

Pass/Fail Criteria:

This test is a benchmark.

Arguments:

- max_run_time: time_limit
- percent_charge_to_add: 100
- percent_initial_charge_max: 5

Test Duration

LONG

Category

Benchmark

Test Type

client

Test Class

power

5.33 Package power_CPUFreq

Verify that supported CPU frequencies can be set.

Classes

- class power_CPUFreq
- class cpufreq

5.33.1 Detailed Description

Verify that supported CPU frequencies can be set.

Pass/Fail Criteria:

This test will fail under the following conditions:

- cpu frequencies are not supported
- no more than 1 frequency is supported
- if a supported frequency cannot be set

Test Duration

SHORT

Category

Functional

Test Type

client

Test Class

power

5.34 Package power_CPUIdle

Ensure the processor drops into idle state when it is idle.

Classes

- class power_CPUIdle
- class cpus
- class cpuidle
- class cpuidle_state

5.34.1 Detailed Description

Ensure the processor drops into idle state when it is idle.

Pass/Fail Criteria:

Fails if the cpu did not have any idle cycles during this test.

Test Duration

SHORT

Category

Functional

Test Type

client

Test Class

power

5.35 Package power_Draw

Measure how much power is drawn over a given amount of time.

Classes

- class power_Draw

5.35.1 Detailed Description

Measure how much power is drawn over a given amount of time.

Pass/Fail Criteria:

This test is a benchmark.

Arguments:

- seconds: seconds
- tag: tag

Test Duration

MEDIUM

Category

Functional

Test Type

client

Test Class

power

5.36 Package power_LoadTest

Measure power draw when system is under load.

Classes

- class power_LoadTest

Variables

- dictionary params_dict

5.36.1 Detailed Description

Measure power draw when system is under load.

Pass/Fail Criteria:

This test is a benchmark.

Arguments:

- loop_time: loop_time
- loop_count: loop_count
- low_battery_threshold: 3

Test Duration

LONG

Category

Benchmark

Test Type

client

Test Class

power

5.36.2 Variable Documentation

5.36.2.1 dictionary power_LoadTest.params_dict

Initial value:

```
{
    'test_time_ms': '_mseconds',
    'should_scroll': '_should_scroll',
    'should_scroll_up': '_should_scroll_up',
    'scroll_loop': '_scroll_loop',
    'scroll_interval_ms': '_scroll_interval_ms',
    'scroll_by_pixels': '_scroll_by_pixels',
}
```

Definition at line 10 of file power_LoadTest.py.

5.37 Package power_Resume

Measure the amount of time it takes to resume from suspend.

Classes

- class power_Resume

5.37.1 Detailed Description

Measure the amount of time it takes to resume from suspend.

Pass/Fail Criteria:

This test is a benchmark.

Arguments:

- seconds_system_resume <= 1.0
- iterations: 10
- tag: _tag

Test Duration

SHORT

Category

Logging

Test Type

client

Test Class

power

5.38 Package realtimecomm_GTalkAudioPlayground

Verify that Google Talk Plugin Audio works with Chrome.

Classes

- class realtimecomm_GTalkAudioPlayground

Variables

- int WARMUP_TIME = 30
- int SLEEP_DURATION = 90

5.38.1 Detailed Description

Verify that Google Talk Plugin Audio works with Chrome.

Pass/Fail Criteria:

Fails if there is an error sending or receiving audio.

Arguments:

- ctime_gtalk + stime_gtalk $<$ 20
- ctime_pulse + stime_pulse $<$ 5

Test Duration

MEDIUM

Category

GTalk

Test Type

client

Test Class

realtimecomm

5.38.2 Variable Documentation

5.38.2.1 int realtimecomm_GTalkAudioPlayground.SLEEP_DURATION = 90

Definition at line 11 of file realtimecomm_GTalkAudioPlayground.py.

5.38.2.2 int realtimecomm_GTalkAudioPlayground.WARMUP_TIME = 30

Definition at line 10 of file realtimecomm_GTalkAudioPlayground.py.

5.39 Package realtimecomm_GTalkPlayground

Verify that Google Talk Plugin executes.

Classes

- class realtimecomm_GTalkPlayground

Variables

- int WARMUP_TIME = 60
- int SLEEP_DURATION = 260

5.39.1 Detailed Description

Verify that Google Talk Plugin executes.

Pass/Fail Criteria:

This test will fail if one of the conditions occurs:

- the runtime is less than 5 minutes
- the number of video streams is not 2
- the video framerate is less than 5 fps

Arguments:

- fps_gtalk_down >= 28.0
- fps_gtalk_up >= 29.0
- ctime_gtalk + stime_gtalk < 65
- ctime_chrome + stime_chrome < 20
- ctime_pulse + stime_pulse < 5

Test Duration

MEDIUM

Category

GTalk

Test Type

client

Test Class

realtimecomm

5.39.2 Variable Documentation

5.39.2.1 int realtimecomm_GTalkPlayground.SLEEP_DURATION = 260

Definition at line 11 of file realtimecomm_GTalkPlayground.py.

5.39.2.2 int realtimecomm_GTalkPlayground.WARMUP_TIME = 60

Definition at line 10 of file realtimecomm_GTalkPlayground.py.

5.40 Package unixbench

Measure system level performance.

Classes

- class unixbench

5.40.1 Detailed Description

Measure system level performance.

Pass/Fail Criteria:

This test is a benchmark.

Test Duration

MEDIUM

Category

Benchmark

Test Type

client

Test Class

Kernel

Chapter 6

Class Documentation

6.1 hardware_BluetoothSemiAuto.Agent Class Reference

Public Member Functions

- def Release
- def RequestPinCode
- def Cancel

Static Public Attributes

- string in_signature = ""

6.1.1 Detailed Description

Definition at line 24 of file hardware_BluetoothSemiAuto.py.

6.1.2 Member Function Documentation

6.1.2.1 def hardware_BluetoothSemiAuto.Agent.Cancel (*self*)

Definition at line 42 of file hardware_BluetoothSemiAuto.py.

```
43                     :
44         logging.debug('Agent: Cancel')
45
```

6.1.2.2 def hardware_BluetoothSemiAuto.Agent.Release (*self*)

Definition at line 28 of file hardware_BluetoothSemiAuto.py.

```
29                     :
30         logging.debug("Agent: Release")
```

```
31
32
    @dbus.service.method("org.bluez.Agent",
```

6.1.2.3 def hardware_BluetoothSemiAuto.Agent.RequestPinCode (*self*, *device*)

Definition at line 34 of file hardware_BluetoothSemiAuto.py.

```
35                                      :
36         pin = '0000'
37         logging.debug('Agent: RequestPinCode (%s), sending %s.', device, pin)
38         return pin
39
40
    @dbus.service.method("org.bluez.Agent",
```

6.1.3 Member Data Documentation

6.1.3.1 string hardware_BluetoothSemiAuto.Agent::in_signature = "" `[static]`

Definition at line 27 of file hardware_BluetoothSemiAuto.py.

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

- testsource/hardware_BluetoothSemiAuto.py

6.2 audiovideo_FFMPEG.audiovideo_FFMPEG Class Reference

Public Member Functions

- def setup
- def run_once
- def run_testcase

Public Attributes

- performance_results
- min_fps_video
- max_tpf_audio

Static Public Attributes

- int version = 1

6.2.1 Detailed Description

Definition at line 38 of file audiovideo_FFMPEG.py.

6.2.2 Member Function Documentation

6.2.2.1 def audiovideo_FFMPEG.audiovideo_FFMPEG.run_once (*self*)

```
Run FFMPEG performance test!
```

Definition at line 52 of file audiovideo_FFMPEG.py.

```
53                      :
54         """ Run FFMPEG performance test! """
55         # fetch all the test cases from file.
56         testcases = os.path.join(self.bindir, "testcases")
57         self.performance_results = {}
58         self.min_fps_video = 100
59         self.max_tpf_audio = 0
60
61         for line in open(testcases, "rt"):
62             # skip comment line and blank line
63             line = line.rstrip()
64             if len(line) == 0: continue
65             if line[0] == "#": continue
66             # run each test cases
67             testcase = line.split()
68             self.run_testcase(testcase)
69         self.performance_results['fps_video_min'] = self.min_fps_video
70         self.performance_results['tpf_audio_max'] = self.max_tpf_audio
71         self.write_perf_keyval(self.performance_results)
72
```

6.2.2.2 def audiovideo_FFMPEG.audiovideo_FFMPEG.run_testcase (*self*, *testcase*)

Definition at line 73 of file audiovideo_FFMPEG.py.

```
74                                  :
75          if utils.get_arch() == 'i386':
76              executable = os.path.join(self.bindir, "ffmpeg_tests.i686")
77          else:  # TODO(jiesun): we only have ARM and i386.
78              executable = os.path.join(self.bindir, "ffmpeg_tests.arm")
79          file_url = testcase[0]
80
81          # TODO(jiesun): if url is not local, grab it from internet.
82          if file_url.startswith("http"):
83              file_name = file_url.split('/')[-1]
84              file_path = os.path.join(self.bindir, file_name)
85              logging.info("Retrieving %s" % file_url)
86              urllib.urlretrieve(file_url, file_path)
87              logging.info("Done.")
88          else:
89              # if url is local, we assume it is in the same directory.
90              file_name = file_url;
91              file_path = os.path.join(self.bindir, file_name)
92
93          if not os.path.exists(file_path):
94              raise error.TestError("ffmpeg_tests: test media missing %s!"
95                                    % file_url)
96
97          command_line = ("LD_LIBRARY_PATH=/opt/google/chrome/ %s %s"
98                          % (executable, file_path))
99          logging.info("Running %s" % command_line)
100
101          cpu_usage, stdout = utils.get_cpu_percentage(
102                                    utils.system_output,
103                                    command_line,
104                                    retain_output=True)
105
106          cpu_usage *= 100.0  # in percentage.
107
108          # what's the fps we measures for video.
109          fps_pattern = re.search(r"FPS:\s+([\d\.]+)", stdout)
110          # what's the time per frame for audio.
111          tpf_pattern = re.search(r"TIME PER FRAME \(MS\):\s+([\d\.]+)", stdout)
112          if fps_pattern:
113              fps = float(fps_pattern.group(1))
114              logging.info("CPU Usage %s%%; FPS: %s" % (cpu_usage, fps))
115              self.min_fps_video = min(self.min_fps_video, fps);
116              # record the performance data for future analysis.
117              namekey = file_name.lower().replace('.', '_')
118              self.performance_results['fps_' + namekey] = fps
119              self.performance_results['cpuusage_' + namekey] = cpu_usage
120          elif tpf_pattern:
121              tpf = float(tpf_pattern.group(1))
122              self.max_tpf_audio = max(self.max_tpf_audio, tpf);
123              logging.info("CPU Usage %s%%; TimePerFrame: %s" % (cpu_usage, tpf))
124              # record the performance data for future analysis.
125              namekey = file_name.lower().replace('.', '_')
126              self.performance_results['timeperframe_' + namekey] = tpf
127              self.performance_results['cpuusage_' + namekey] = cpu_usage
128          else:
129              raise error.TestFail("ffmpeg_tests failed to exit normally!")
130
131          # TODO(jiesun/fbarchard): what else need to be checked?
132
133          # remove file after test to save diskspace.
134          os.remove(file_path);
135
136
```

6.2.2.3 def audiovideo_FFMPEG.audiovideo_FFMPEG.setup (*self*)

```
copy test asset to bindir.
```

Definition at line 41 of file audiovideo_FFMPEG.py.

```
42                     :
43         """ copy test asset to bindir. """
44         if not os.path.exists(self.srcdir):
45             os.mkdir(self.srcdir)
46         # sysroot = os.environ["SYSROOT"]
47         # testdir = os.path.join(sysroot, "usr/local/autotest-chrome")
48         # testbin = os.path.join(testdir, "ffmpeg_tests")
49     # TODO(jiesun): retrieve chrome test asset from build.
50         # shutil.copy(testbin, self.bindir)
51
```

6.2.3 Member Data Documentation

6.2.3.1 audiovideo_FFMPEG.audiovideo_FFMPEG.max_tpf_audio

Definition at line 58 of file audiovideo_FFMPEG.py.

6.2.3.2 audiovideo_FFMPEG.audiovideo_FFMPEG.min_fps_video

Definition at line 57 of file audiovideo_FFMPEG.py.

6.2.3.3 audiovideo_FFMPEG.audiovideo_FFMPEG.performance_results

Definition at line 56 of file audiovideo_FFMPEG.py.

6.2.3.4 int audiovideo_FFMPEG.audiovideo_FFMPEG.version = 1 `[static]`

Definition at line 39 of file audiovideo_FFMPEG.py.

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

- testsource/audiovideo_FFMPEG.py

6.3 audiovideo_V4L2.audiovideo_V4L2 Class Reference

Public Member Functions

- def setup
- def run_once
- def find_video_capture_devices
- def unittest_passed
- def run_v4l2_unittests
- def run_v4l2_capture_test
- def run_v4l2_default_capture_test
- def run_v4l2_capture_tests

Public Attributes

- assert_mandatory_controls
- v4l2_devices
- executable
- supported_controls
- support_streaming
- support_readwrite
- supported_formats

Static Public Attributes

- int version = 1
- preserve_srcdir = True
- int v4l2_major_dev_num = 81
- int v4l2_minor_dev_num_min = 0
- int v4l2_minor_dev_num_max = 64

6.3.1 Detailed Description

Definition at line 45 of file audiovideo_V4L2.py.

6.3.2 Member Function Documentation

6.3.2.1 def audiovideo_V4L2.audiovideo_V4L2.find_video_capture_devices (*self*)

Definition at line 77 of file audiovideo_V4L2.py.

```
78                                      :
79         self.v4l2_devices = []
80         # TODO(jiesun): is this correct way to find video capture devices?
81         for device in glob.glob("/dev/video*"):
82             statinfo = os.stat(device)
83             if (stat.S_ISCHR(statinfo.st_mode) and
84                 os.major(statinfo.st_rdev) == self.v4l2_major_dev_num and
85                 os.minor(statinfo.st_rdev) >= self.v4l2_minor_dev_num_min and
86                 os.minor(statinfo.st_rdev) < self.v4l2_minor_dev_num_max):
87                 self.v4l2_devices.append(device)
```

```
88          logging.info("Detected devices: %s\n" % self.v4l2_devices)
89          if not self.v4l2_devices:
90              raise error.TestFail("No V4L2 devices found!")
91
```

6.3.2.2 def audiovideo_V4L2.audiovideo_V4L2.run_once (*self*, *run_unit_tests* = `True`, *run_capture_tests* = `True`, *run_default_capture_test* = `False`, *time* = `0`, *assert_mandatory_controls* = `True`)

Definition at line 60 of file audiovideo_V4L2.py.

```
63                                                  :
64
65          self.assert_mandatory_controls = assert_mandatory_controls
66          self.find_video_capture_devices()
67          time = time / len(self.v4l2_devices)
68
69          for device in self.v4l2_devices:
70              if run_unit_tests:
71                  self.run_v4l2_unittests(device)
72              if run_capture_tests:
73                  self.run_v4l2_capture_tests(device)
74              if run_default_capture_test:
75                  self.run_v4l2_default_capture_test(device, time)
76
```

6.3.2.3 def audiovideo_V4L2.audiovideo_V4L2.run_v4l2_capture_test (*self*, *fail_okay*, *options*)

Definition at line 173 of file audiovideo_V4L2.py.

```
174                                                         :
175          executable = os.path.join(self.bindir, "media_v4l2_test")
176          try:
177              cmd = "%s %s" % (executable, " ".join(options))
178              cmd = site_ui.xcommand(cmd)
179              logging.info("Running %s" % cmd)
180              stdout = utils.system_output(cmd, retain_output=True)
181          except:
182              if fail_okay:
183                  stdout = ""
184                  return (False, stdout)
185              else:
186                  raise
187          else:
188              return (True, stdout)
189
```

6.3.2.4 def audiovideo_V4L2.audiovideo_V4L2.run_v4l2_capture_tests (*self*, *device*)

Definition at line 198 of file audiovideo_V4L2.py.

```
199                                         :
200          default_options = ["--device=%s" % device ]
```

```
201
202         # If the device claims to support read/write i/o.
203         if self.support_readwrite:
204             option = default_options + ["--read"]
205             okay, stdout = self.run_v4l2_capture_test(False, option)
206
207         # If the device claims to support stream i/o.
208         # This could mean either mmap stream i/o or user pointer stream i/o.
209         if self.support_streaming:
210             option = default_options + ["--mmap"]
211             mmap_okay, stdout = self.run_v4l2_capture_test(True, option)
212
213             option = default_options + ["--userp"]
214             userp_okay, stdout = self.run_v4l2_capture_test(True, option)
215
216             if not userp_okay and not mmap_okay:
217                 raise error.TestFail("Stream i/o failed!")
218
219
220         # TODO(jiesun): test with different mandatory resultions that
221         # the capture device must support without scaling by ourselves.
222         required_resolutions = [
223             (320, 240, 30),  # SIF
224             (352, 288, 30),  # CIF
225             (640, 480, 30),  # VGA
226             (176, 144, 30)]  # QCIF
227         for (width, height, minfps) in required_resolutions:
228             # Note use default mmap i/o here.
229             option = default_options[:]
230             # Note use first supported pixel format.
231             option.append("--pixel-format=%s" % self.supported_formats[0])
232             option.append("--width=%s" % width)
233             option.append("--height=%s" % height)
234             okay, stdout = self.run_v4l2_capture_test(False, option)
235             # Check if the actual format is desired.
236             pattern = (r"actual format for capture (\d+)x(\d+)"
237                        r" (....) picture at (\d+) fps")
238             match = re.search(pattern, stdout)
239             if (not match or
240                 int(match.group(1)) != width or
241                 int(match.group(2)) != height or
242                 match.group(3) != self.supported_formats[0] or
243                 int(match.group(4)) < minfps):
244                 raise error.TestError("capture test failed")
245
246             # Run the test again with X display.
247             option.append("--display")
248             okay, stdout = self.run_v4l2_capture_test(False, option)
```

6.3.2.5 def audiovideo_V4L2.audiovideo_V4L2.run_v4l2_default_capture_test (*self*, *device*, *time*)

Definition at line 190 of file audiovideo_V4L2.py.

```
191                                                         :
192         options = ["--device=%s" % device ]
193         options.append("--display")
194         if time:
195             options.append("--time=%d" % time)
196         okay, stdout = self.run_v4l2_capture_test(False, options)
197
```

6.3.2.6 def audiovideo_V4L2.audiovideo_V4L2.run_v4l2_unittests (*self*, *device*)

Definition at line 96 of file audiovideo_V4L2.py.

```
97                                      :
98          self.executable = os.path.join(self.bindir, "media_v4l2_unittest")
99          cmd = "%s --device=%s" % (self.executable, device)
100          logging.info("Running %s" % cmd)
101          stdout = utils.system_output(cmd, retain_output=True)
102
103          # Check the result of unittests.
104          # We had exercise all the optional ioctls in unittest which maybe
105          # optional by V4L2 Specification.  Therefore we need to check those
106          # tests that we thought are mandatory.
107          # 1. Multiple open should be supported for panel application.
108          if not self.unittest_passed("MultipleOpen", stdout):
109              raise error.TestError(device + " does not support multiple open!")
110
111          # 2. Need to make sure this is really support or just driver error.
112          if not self.unittest_passed("MultipleInit", stdout):
113              raise error.TestError(device + " does support multiple init!")
114
115          # 3. EnumInput and EnumStandard is optional.
116
117          # 4. EnumControl is mandatory.
118          if not self.unittest_passed("EnumControl", stdout):
119              raise error.TestError(device + " does support enum controls!")
120          pattern = re.compile(r"Control (\w+) is enabled\((\d+)-(\d+):(\d+)\)")
121          control_info = pattern.findall(stdout)
122          self.supported_controls = [ x[0] for x in control_info ]
123          logging.info("Supported Controls: %s\n" % self.supported_controls)
124
125          # TODO(jiesun): what is required?
126          mandatory_controls = [
127              "Brightness",
128              "Contrast",
129              "Saturation",
130              "Hue",
131              "Gamma"]
132          for control in mandatory_controls:
133              if self.assert_mandatory_controls and \
134                  control not in self.supported_controls:
135                  raise error.TestError(device + " does not support " + control)
136
137          # 5. SetControl is mandatory.
138          if not self.unittest_passed("SetControl", stdout):
139              raise error.TestError(device + " does not support set controls!")
140
141          # 6. 7. Set/GetCrop are both optional.
142
143          # 8. ProbeCaps is mandatory.
144          if not self.unittest_passed("ProbeCaps", stdout):
145              raise error.TestError(device + " does not support probe caps!")
146
147          if not re.search(r"support video capture interface.>>>", stdout):
148              raise error.TestFail(device + " does not support video capture!")
149
150          pattern = r"support streaming i/o interface.>>>"
151          self.support_streaming = True if re.search(pattern, stdout) else False
152
153          pattern = r"support streaming read/write interface.>>>"
154          self.support_readwrite = True if re.search(pattern, stdout) else False
155
156          # Currently I assume streaming (mmap) is mandatroy.
157          if not self.support_streaming:
158              raise error.TestFail(device + " does not support streaming!")
159
```

```
160         # 9. EnumFormats is always mandatory.
161         if not self.unittest_passed("EnumFormats", stdout):
162             raise error.TestError(device + " does not support enum formats!")
163
164         pattern = re.compile(r"supported format #\d+: .* \((....)\)")
165         format_info = pattern.findall(stdout)
166         # Remove duplicated pixel formats from list.
167         self.supported_formats = list(set(format_info))
168         logging.info("Supported pixel format: %s\n", self.supported_formats)
169
170         # 10. Get/SetParam for framerate is optional.
171         # 11. EnumFrameSize is optional on some kernel/v4l2 version.
172
```

6.3.2.7 def audiovideo_V4L2.audiovideo_V4L2.setup (*self*)

Definition at line 53 of file audiovideo_V4L2.py.

```
54                 :
55         # TODO(jiesun): make binary here when cross compile issue is resolved.
56         os.chdir(self.srcdir)
57         utils.system("make clean")
58         utils.system("make")
59
```

6.3.2.8 def audiovideo_V4L2.audiovideo_V4L2.unittest_passed (*self*, *testname*, *stdout*)

Definition at line 92 of file audiovideo_V4L2.py.

```
93                                                 :
94         return re.search(r"OK \] V4L2DeviceTest\." + testname, stdout);
95
```

6.3.3 Member Data Documentation

6.3.3.1 audiovideo_V4L2.audiovideo_V4L2.assert_mandatory_controls

Definition at line 62 of file audiovideo_V4L2.py.

6.3.3.2 audiovideo_V4L2.audiovideo_V4L2.executable

Definition at line 97 of file audiovideo_V4L2.py.

6.3.3.3 audiovideo_V4L2.audiovideo_V4L2.preserve_srcdir = True [static]

Definition at line 47 of file audiovideo_V4L2.py.

6.3.3.4 audiovideo_V4L2.audiovideo_V4L2.support_readwrite

Definition at line 153 of file audiovideo_V4L2.py.

6.3.3.5 audiovideo_V4L2.audiovideo_V4L2.support_streaming

Definition at line 150 of file audiovideo_V4L2.py.

6.3.3.6 audiovideo_V4L2.audiovideo_V4L2.supported_controls

Definition at line 121 of file audiovideo_V4L2.py.

6.3.3.7 audiovideo_V4L2.audiovideo_V4L2.supported_formats

Definition at line 166 of file audiovideo_V4L2.py.

6.3.3.8 audiovideo_V4L2.audiovideo_V4L2.v4l2_devices

Definition at line 78 of file audiovideo_V4L2.py.

6.3.3.9 int audiovideo_V4L2.audiovideo_V4L2.v4l2_major_dev_num = 81 [static]

Definition at line 48 of file audiovideo_V4L2.py.

6.3.3.10 int audiovideo_V4L2.audiovideo_V4L2.v4l2_minor_dev_num_max = 64 [static]

Definition at line 50 of file audiovideo_V4L2.py.

6.3.3.11 int audiovideo_V4L2.audiovideo_V4L2.v4l2_minor_dev_num_min = 0 [static]

Definition at line 49 of file audiovideo_V4L2.py.

6.3.3.12 int audiovideo_V4L2.audiovideo_V4L2.version = 1 [static]

Definition at line 46 of file audiovideo_V4L2.py.

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

- testsource/audiovideo_V4L2.py

6.4 compilebench.compilebench Class Reference

Public Member Functions

- def setup
- def run_once

Public Attributes

- tarball

Static Public Attributes

- int version = 1

Private Member Functions

- def __format_results

6.4.1 Detailed Description

Definition at line 28 of file compilebench.py.

6.4.2 Member Function Documentation

6.4.2.1 def compilebench.compilebench.__format_results (*self*, *output*) [private]

Definition at line 54 of file compilebench.py.

```
55                                      :
56         keylist = {}
57
58         THROUGHPUT = "MB_s"
59         TIME       = "secs"
60
61         run_type_list = (
62             ('intial create', THROUGHPUT, 6, 'initial_create'),
63             ('create', THROUGHPUT, 5, 'new_create'),
64             ('patch', THROUGHPUT, 5, 'patch'),
65             ('compile', THROUGHPUT, 5, 'compile'),
66             ('clean', THROUGHPUT, 5, 'clean'),
67             ('read tree', THROUGHPUT, 6, 'read_tree'),
68             ('read compiled tree', THROUGHPUT, 7, 'read_compiled_tree'),
69             ('delete tree', TIME, 6, 'delete_tree'),
70             ('delete compiled tree', TIME, 6, 'delete_compiled_tree'),
71             ('stat tree', TIME, 6, 'stat_tree'),
72             ('stat compiled tree', TIME, 7, 'stat_compiled_tree'),
73         )
74
75 # intial create total runs 10 avg 149.82 MB/s (user 0.63s sys 0.85s)
76 # create total runs 5 avg 27.50 MB/s (user 0.62s sys 0.83s)
77 # patch total runs 4 avg 15.01 MB/s (user 0.33s sys 0.63s)
78 # compile total runs 7 avg 41.47 MB/s (user 0.14s sys 0.75s)
79 # clean total runs 4 avg 697.77 MB/s (user 0.02s sys 0.08s)
```

```
80 # read tree total runs 2 avg 23.68 MB/s (user 0.85s sys 1.59s)
81 # read compiled tree total runs 1 avg 25.27 MB/s (user 0.98s sys 2.84s)
82 # delete tree total runs 2 avg 1.48 seconds (user 0.35s sys 0.45s)
83 # no runs for delete compiled tree
84 # stat tree total runs 4 avg 1.46 seconds (user 0.35s sys 0.26s)
85 # stat compiled tree total runs 1 avg 1.49 seconds (user 0.37s sys 0.29s)
86
87         for line in output.splitlines():
88             for pattern, result_type, position, tag in run_type_list:
89                 if re.search('^%s' % pattern, line):
90                     l = line.split()
91                     value = l[position]
92
93                     s = "%s_%s" % (tag, result_type)
94
95                     keylist[s] = value
96                     break
97
98         self.write_perf_keyval(keylist)
```

6.4.2.2 def compilebench.compilebench.run_once (*self*, *dir* = `None`, *num_kernel_trees* = `10`, *num_random_runs* = `30`)

Definition at line 38 of file compilebench.py.

```
39                                                                              :
40         if not dir:
41             dir = self.tmpdir
42
43         cmd = "%s -D %s -s %s -i %d -r %d" % (
44                         os.path.join(self.srcdir, test_name),
45                         dir,
46                         self.srcdir,
47                         num_kernel_trees,
48                         num_random_runs)
49
50         output = utils.system_output(cmd)
51
52         self.__format_results(output)
53
```

6.4.2.3 def compilebench.compilebench.setup (*self*, *tarball* = `'compilebench-0.6.tar.gz'`)

Definition at line 31 of file compilebench.py.

```
32                                                          :
33         self.tarball = utils.unmap_url(self.bindir, tarball, self.tmpdir)
34         utils.extract_tarball_to_dir(self.tarball, self.srcdir)
35         os.chdir(self.srcdir)
36         utils.system('patch -p1 < ../compilebench.patch')
37
```

6.4.3 Member Data Documentation

6.4.3.1 compilebench.compilebench.tarball

Definition at line 32 of file compilebench.py.

6.4.3.2 int compilebench.compilebench.version = 1 [static]

Definition at line 29 of file compilebench.py.

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

- testsource/compilebench.py

6.5 power_CPUFreq.cpufreq Class Reference

Public Member Functions

- def __init__
- def save_state
- def restore_state
- def get_available_governors
- def get_current_governor
- def set_governor
- def get_available_frequencies
- def get_current_frequency
- def set_frequency

Private Member Functions

- def __write_file
- def __read_file

Private Attributes

- __base_path
- __save_files_list

6.5.1 Detailed Description

Definition at line 73 of file power_CPUFreq.py.

6.5.2 Member Function Documentation

6.5.2.1 def power_CPUFreq.cpufreq.__init__ (*self*, *path*)

Definition at line 74 of file power_CPUFreq.py.

```
75                              :
76         self.__base_path = path
77         self.__save_files_list = ['scaling_max_freq', 'scaling_min_freq',
78                                   'scaling_governor']
79
```

6.5.2.2 def power_CPUFreq.cpufreq.__read_file (*self*, *file_name*) [private]

Definition at line 85 of file power_CPUFreq.py.

```
86                                      :
87         path = os.path.join(self.__base_path, file_name)
88         f = open(path, 'r')
89         data = f.read()
90         f.close()
```

```
91         return data
92
```

6.5.2.3 def power_CPUFreq.cpufreq.__write_file (*self*, *file_name*, *data*) [private]

Definition at line 80 of file power_CPUFreq.py.

```
81                                          :
82         path = os.path.join(self.__base_path, file_name)
83         utils.open_write_close(path, data)
84
```

6.5.2.4 def power_CPUFreq.cpufreq.get_available_frequencies (*self*)

Definition at line 126 of file power_CPUFreq.py.

```
127                                     :
128         frequencies = self.__read_file('scaling_available_frequencies')
129         logging.info('available frequencies: %s' % frequencies)
130         return [int(i) for i in frequencies.split()]
131
```

6.5.2.5 def power_CPUFreq.cpufreq.get_available_governors (*self*)

Definition at line 109 of file power_CPUFreq.py.

```
110                                    :
111         governors = self.__read_file('scaling_available_governors')
112         logging.info('available governors: %s' % governors)
113         return governors.split()
114
```

6.5.2.6 def power_CPUFreq.cpufreq.get_current_frequency (*self*)

Definition at line 132 of file power_CPUFreq.py.

```
133                                   :
134         freq = int(self.__read_file('scaling_cur_freq'))
135         logging.info('current frequency: %s' % freq)
136         return freq
137
```

6.5.2.7 def power_CPUFreq.cpufreq.get_current_governor (*self*)

Definition at line 115 of file power_CPUFreq.py.

```
116                                     :
117         governor = self.__read_file('scaling_governor')
118         logging.info('current governor: %s' % governor)
119         return governor.split()[0]
120
```

6.5.2.8 def power_CPUFreq.cpufreq.restore_state (*self*)

Definition at line 101 of file power_CPUFreq.py.

```
102                             :
103         logging.info('restoring state:')
104         for file in self.__save_files_list:
105             data = getattr(self, file)
106             logging.info(file + ': '  + data)
107             self.__write_file(file, data)
108
```

6.5.2.9 def power_CPUFreq.cpufreq.save_state (*self*)

Definition at line 93 of file power_CPUFreq.py.

```
94                         :
95         logging.info('saving state:')
96         for file in self.__save_files_list:
97             data = self.__read_file(file)
98             setattr(self, file, data)
99             logging.info(file + ': '  + data)
100
```

6.5.2.10 def power_CPUFreq.cpufreq.set_frequency (*self*, *frequency*)

Definition at line 138 of file power_CPUFreq.py.

```
139                                       :
140         logging.info('setting frequency to %d' % frequency)
141         if frequency >= self.get_current_frequency():
142             file_list = ['scaling_max_freq', 'scaling_min_freq',
143                          'scaling_setspeed']
144         else:
145             file_list = ['scaling_min_freq', 'scaling_max_freq',
146                          'scaling_setspeed']
147
148         for file in file_list:
149             self.__write_file(file, str(frequency))
```

6.5.2.11 def power_CPUFreq.cpufreq.set_governor (*self*, *governor*)

Definition at line 121 of file power_CPUFreq.py.

```
122                                     :
123         logging.info('setting governor to %s' % governor)
124         self.__write_file('scaling_governor', governor)
125
```

6.5.3 Member Data Documentation

6.5.3.1 power_CPUFreq.cpufreq.__base_path [private]

Definition at line 75 of file power_CPUFreq.py.

6.5.3.2 power_CPUFreq.cpufreq.__save_files_list [private]

Definition at line 76 of file power_CPUFreq.py.

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

- testsource/power_CPUFreq.py

6.6 power_CPUIdle.cpuidle Class Reference

Public Member Functions

- def __init__
- def idle_time

Private Attributes

- __base_path
- __states

6.6.1 Detailed Description

Definition at line 76 of file power_CPUIdle.py.

6.6.2 Member Function Documentation

6.6.2.1 def power_CPUIdle.cpuidle.__init__ (*self*, *path*)

Definition at line 77 of file power_CPUIdle.py.

```
78                              :
79         self.__base_path = path
80         self.__states = []
81 
82         dirs = glob.glob(os.path.join(self.__base_path, 'state*'))
83         if not dirs:
84             raise error.TestError('cpuidle states missing')
85 
86         for dir in dirs:
87             state = cpuidle_state(dir)
88             self.__states.append(state)
89 
```

6.6.2.2 def power_CPUIdle.cpuidle.idle_time (*self*)

Definition at line 90 of file power_CPUIdle.py.

```
91                         :
92         total_idle_time = 0
93         for state in self.__states:
94             total_idle_time += state.idle_time()
95 
96         return total_idle_time
97 
98 
```

6.6.3 Member Data Documentation

6.6.3.1 power_CPUIdle.cpuidle.__base_path [private]

Definition at line 78 of file power_CPUIdle.py.

6.6.3.2 power_CPUIdle.cpuidle.__states [private]

Definition at line 79 of file power_CPUIdle.py.

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

- testsource/power_CPUIdle.py

6.7 power_CPUIdle.cpuidle_state Class Reference

Public Member Functions

- def __init__
- def idle_time

Private Member Functions

- def __read_file

Private Attributes

- __base_path
- __name

6.7.1 Detailed Description

Definition at line 99 of file power_CPUIdle.py.

6.7.2 Member Function Documentation

6.7.2.1 def power_CPUIdle.cpuidle_state.__init__ (*self*, *path*)

Definition at line 100 of file power_CPUIdle.py.

```
101                                 :
102         self.__base_path = path
103         self.__name = self.__read_file('name').split()[0]
104
```

6.7.2.2 def power_CPUIdle.cpuidle_state.__read_file (*self*, *file_name*) [private]

Definition at line 105 of file power_CPUIdle.py.

```
106                                         :
107         path = os.path.join(self.__base_path, file_name)
108         f = open(path, 'r')
109         data = f.read()
110         f.close()
111         return data
112
```

6.7.2.3 def power_CPUIdle.cpuidle_state.idle_time (*self*)

Definition at line 113 of file power_CPUIdle.py.

```
114                         :
115         time = int(self.__read_file('time'))
116         logging.info('idle_time(%s): %s' % (self.__name, time))
117         return time
```

6.7.3 Member Data Documentation

6.7.3.1 power_CPUIdle.cpuidle_state.__base_path [private]

Definition at line 101 of file power_CPUIdle.py.

6.7.3.2 power_CPUIdle.cpuidle_state.__name [private]

Definition at line 102 of file power_CPUIdle.py.

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

- testsource/power_CPUIdle.py

6.8 power_CPUIdle.cpus Class Reference

Public Member Functions

- def __init__
- def idle_time

Private Attributes

- __base_path
- __cpus

6.8.1 Detailed Description

Definition at line 54 of file power_CPUIdle.py.

6.8.2 Member Function Documentation

6.8.2.1 def power_CPUIdle.cpus.__init__ (*self*)

Definition at line 55 of file power_CPUIdle.py.

```
56                         :
57         self.__base_path = '/sys/devices/system/cpu/cpu*/cpuidle'
58         self.__cpus = []
59
60         dirs = glob.glob(self.__base_path)
61         if not dirs:
62             raise error.TestError('cpuidle not supported')
63
64         for dir in dirs:
65             cpu = cpuidle(dir)
66             self.__cpus.append(cpu)
67
```

6.8.2.2 def power_CPUIdle.cpus.idle_time (*self*)

Definition at line 68 of file power_CPUIdle.py.

```
69                         :
70         total_idle_time = 0
71         for cpu in self.__cpus:
72             total_idle_time += cpu.idle_time()
73         return total_idle_time
74
75
```

6.8.3 Member Data Documentation

6.8.3.1 power_CPUIdle.cpus.__base_path [private]

Definition at line 56 of file power_CPUIdle.py.

6.8.3.2 power_CPUIdle.cpus.__cpus [private]

Definition at line 57 of file power_CPUIdle.py.

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

- testsource/power_CPUIdle.py

6.9 desktopui_SunSpiderBench.desktopui_SunSpiderBench Class Reference

Public Member Functions

- def initialize
- def setup
- def cleanup
- def run_once

Static Public Attributes

- int version = 1

Private Attributes

- _test_url
- _testServer

6.9.1 Detailed Description

Definition at line 31 of file desktopui_SunSpiderBench.py.

6.9.2 Member Function Documentation

6.9.2.1 def desktopui_SunSpiderBench.desktopui_SunSpiderBench.cleanup (*self*)

Definition at line 49 of file desktopui_SunSpiderBench.py.

```
50                     :
51         self._testServer.stop()
52         site_ui_test.UITest.cleanup(self)
53
```

6.9.2.2 def desktopui_SunSpiderBench.desktopui_SunSpiderBench.initialize (*self*, *creds* = '$default')

Definition at line 34 of file desktopui_SunSpiderBench.py.

```
35                                          :
36         self._test_url = 'http://localhost:8000/sunspider-driver.html'
37         self._testServer = site_httpd.HTTPListener(8000, docroot=self.srcdir)
38         self._testServer.run()
39         site_ui_test.UITest.initialize(self, creds)
40
```

6.9.2.3 def desktopui_SunSpiderBench.desktopui_SunSpiderBench.run_once (*self*, *timeout* = 180)

Definition at line 54 of file desktopui_SunSpiderBench.py.

```
55                                      :
56         latch = self._testServer.add_wait_url('/sunspider/scores')
57
58         session = site_ui.ChromeSession(self._test_url)
59         logging.debug('Chrome session started.')
60         latch.wait(timeout)
61         session.close()
62
63         if not latch.is_set():
64             raise error.TestFail('Never received callback from browser.')
65
66         self.write_perf_keyval(self._testServer.get_form_entries())
```

6.9.2.4 def desktopui_SunSpiderBench.desktopui_SunSpiderBench.setup (*self*, *tarball* = 'sunspider-0.9.tar.bz2')

Definition at line 41 of file desktopui_SunSpiderBench.py.

```
42                                                      :
43         shutil.rmtree(self.srcdir, ignore_errors=True)
44         tarball = utils.unmap_url(self.bindir, tarball, self.tmpdir)
45         utils.extract_tarball_to_dir(tarball, self.srcdir)
46         os.chdir(self.srcdir)
47         utils.system('patch -p1 < ../sunspider.patch')
48
```

6.9.3 Member Data Documentation

6.9.3.1 desktopui_SunSpiderBench.desktopui_SunSpiderBench._test_url [private]

Definition at line 35 of file desktopui_SunSpiderBench.py.

6.9.3.2 desktopui_SunSpiderBench.desktopui_SunSpiderBench._testServer [private]

Definition at line 36 of file desktopui_SunSpiderBench.py.

6.9.3.3 int desktopui_SunSpiderBench.desktopui_SunSpiderBench.version = 1 [static]

Definition at line 32 of file desktopui_SunSpiderBench.py.

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

- testsource/desktopui_SunSpiderBench.py

6.10 desktopui_V8Bench.desktopui_V8Bench Class Reference

Public Member Functions

- def initialize
- def setup
- def cleanup
- def run_once

Static Public Attributes

- int version = 1

Private Attributes

- _test_url
- _testServer

6.10.1 Detailed Description

Definition at line 34 of file desktopui_V8Bench.py.

6.10.2 Member Function Documentation

6.10.2.1 def desktopui_V8Bench.desktopui_V8Bench.cleanup (*self*)

Definition at line 52 of file desktopui_V8Bench.py.

```
53                     :
54         self._testServer.stop()
55         site_ui_test.UITest.cleanup(self)
56
```

6.10.2.2 def desktopui_V8Bench.desktopui_V8Bench.initialize (*self*, *creds* = `'$default'`)

Definition at line 37 of file desktopui_V8Bench.py.

```
38                                             :
39         self._test_url = 'http://localhost:8000/run.html'
40         self._testServer = site_httpd.HTTPListener(8000, docroot=self.srcdir)
41         self._testServer.run()
42         site_ui_test.UITest.initialize(self, creds)
43
```

6.10.2.3 def desktopui_V8Bench.desktopui_V8Bench.run_once (*self*, *timeout* = 60)

Definition at line 57 of file desktopui_V8Bench.py.

```
58                                      :
59         latch = self._testServer.add_wait_url('/v8/scores')
60
61         session = site_ui.ChromeSession(self._test_url)
62         logging.debug('Chrome session started.')
63         latch.wait(timeout)
64         session.close()
65
66         if not latch.is_set():
67             raise error.TestFail('Never received callback from browser.')
68
69         self.write_perf_keyval(self._testServer.get_form_entries())
```

6.10.2.4 def desktopui_V8Bench.desktopui_V8Bench.setup (*self*, *tarball* = 'v8_v5.tar.bz2')

Definition at line 44 of file desktopui_V8Bench.py.

```
45                                              :
46         shutil.rmtree(self.srcdir, ignore_errors=True)
47         tarball = utils.unmap_url(self.bindir, tarball, self.tmpdir)
48         utils.extract_tarball_to_dir(tarball, self.srcdir)
49         os.chdir(self.srcdir)
50         utils.system('patch -p1 < ../v8.patch')
51
```

6.10.3 Member Data Documentation

6.10.3.1 desktopui_V8Bench.desktopui_V8Bench._test_url [private]

Definition at line 38 of file desktopui_V8Bench.py.

6.10.3.2 desktopui_V8Bench.desktopui_V8Bench._testServer [private]

Definition at line 39 of file desktopui_V8Bench.py.

6.10.3.3 int desktopui_V8Bench.desktopui_V8Bench.version = 1 [static]

Definition at line 35 of file desktopui_V8Bench.py.

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

- testsource/desktopui_V8Bench.py

6.11 disktest.disktest Class Reference

Public Member Functions

- def setup
- def initialize
- def test_one_disk_chunk
- def execute

Public Attributes

- chunk_mb
- memory_mb

Static Public Attributes

- int version = 1
- preserve_srcdir = True

6.11.1 Detailed Description

Definition at line 30 of file disktest.py.

6.11.2 Member Function Documentation

6.11.2.1 def disktest.disktest.execute (*self*, *disks* = `None`, *gigabytes* = `None`, *chunk_mb* = `utils.memtotal() / 1024`)

Definition at line 53 of file disktest.py.

```
55                                                  :
56         os.chdir(self.srcdir)
57
58         if not disks:
59             disks = [self.tmpdir]
60         if not gigabytes:
61             free = 100       # cap it at 100GB by default
62             for disk in disks:
63                 free = min(utils.freespace(disk) / 1024**3, free)
64             gigabytes = free
65             logging.info("resizing to %s GB", gigabytes)
66             sys.stdout.flush()
67
68         self.chunk_mb = chunk_mb
69         self.memory_mb = utils.memtotal()/1024
70         if self.memory_mb > chunk_mb:
71             e_msg = "Too much RAM (%dMB) for this test to work" % self.memory_mb
72             raise error.TestError(e_msg)
73
74         chunks = (1024 * gigabytes) / chunk_mb
75
76         for i in range(chunks):
77             pids = []
78             for disk in disks:
```

```
79                 pid = self.test_one_disk_chunk(disk, i)
80                 pids.append(pid)
81             errors = []
82             for pid in pids:
83                 (junk, retval) = os.waitpid(pid, 0)
84                 if (retval != 0):
85                     errors.append(retval)
86             if errors:
87                 raise error.TestError("Errors from children: %s" % errors)
```

6.11.2.2 def disktest.disktest.initialize (*self*)

Definition at line 40 of file disktest.py.

```
41                         :
42         self.job.require_gcc()
43
```

6.11.2.3 def disktest.disktest.setup (*self*)

Definition at line 34 of file disktest.py.

```
35                 :
36         os.chdir(self.srcdir)
37         utils.system('make clean')
38         utils.system('make')
39
```

6.11.2.4 def disktest.disktest.test_one_disk_chunk (*self*, *disk*, *chunk*)

Definition at line 44 of file disktest.py.

```
45                                             :
46         logging.info("testing %d MB files on %s in %d MB memory",
47                      self.chunk_mb, disk, self.memory_mb)
48         cmd = "%s/disktest -m %d -f %s/testfile.%d -i -S" % \
49                                 (self.srcdir, self.chunk_mb, disk, chunk)
50         p = subprocess.Popen(cmd, shell=True)
51         return(p.pid)
52
```

6.11.3 Member Data Documentation

6.11.3.1 disktest.disktest.chunk_mb

Definition at line 66 of file disktest.py.

6.11.3.2 disktest.disktest.memory_mb

Definition at line 67 of file disktest.py.

6.11.3.3 disktest.disktest.preserve_srcdir = True [static]

Definition at line 32 of file disktest.py.

6.11.3.4 int disktest.disktest.version = 1 [static]

Definition at line 31 of file disktest.py.

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

- testsource/disktest.py

6.12 firmware_RomSize.firmware_RomSize Class Reference

Public Member Functions

- def run_once

Static Public Attributes

- int version = 1

6.12.1 Detailed Description

Definition at line 32 of file firmware_RomSize.py.

6.12.2 Member Function Documentation

6.12.2.1 def firmware_RomSize.firmware_RomSize.run_once (*self*)

Definition at line 35 of file firmware_RomSize.py.

```
36                      :
37         cmd = 'dmidecode | grep "ROM Size" | sed "s/.*: \([0-9]\+\) kB/\\1/"'
38         size = int(utils.system_output(cmd).strip())
39         self.write_perf_keyval({"kb_system_rom_size": size})
```

6.12.3 Member Data Documentation

6.12.3.1 int firmware_RomSize.firmware_RomSize.version = 1 `[static]`

Definition at line 33 of file firmware_RomSize.py.

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

- testsource/firmware_RomSize.py

6.13 firmware_VbootCrypto.firmware_VbootCrypto Class Reference

Public Member Functions

- def setup
- def run_crypto
- def run_verification
- def run_benchmarks
- def run_rollback
- def run_splicing
- def run_once

Public Attributes

- results
- keyvals

Static Public Attributes

- int version = 1
- preserve_srcdir = True

Private Member Functions

- def __output_result_keyvals
- def __generate_test_cases
- def __sha_test
- def __rsa_test
- def __image_verification_test
- def __sha_benchmark
- def __rsa_benchmark
- def __verify_image_benchmark
- def __rollback_tests
- def __splicing_tests

6.13.1 Detailed Description

```
Tests for correctness of verified boot reference crypto implementation.
```

Definition at line 47 of file firmware_VbootCrypto.py.

6.13.2 Member Function Documentation

6.13.2.1 def firmware_VbootCrypto.firmware_VbootCrypto.__generate_test_cases (*self*) [private]

Definition at line 68 of file firmware_VbootCrypto.py.

```
69                                  :
70         gen_test_case_cmd = os.path.join(self.srcdir, "tests",
71                                          "gen_test_cases.sh")
72         return_code = utils.system(gen_test_case_cmd, ignore_status = True)
73         if return_code == 255:
74             return False
75         if return_code == 1:
76             raise error.TestError("Couldn't generate test cases")
77         return True
78
```

6.13.2.2 def firmware_VbootCrypto.firmware_VbootCrypto.__image_verification_test (*self*) [private]

Definition at line 101 of file firmware_VbootCrypto.py.

```
102                                       :
103         image_verification_cmd = "cd %s && ./run_image_verification_tests.sh" \
104                                  % os.path.join(self.srcdir, "tests")
105         return_code = utils.system(image_verification_cmd,
106                                    ignore_status=True)
107         if return_code == 255:
108             return False
109         if return_code == 1:
110             raise error.TestError("Image Verification Test Error")
111         return True
112
```

6.13.2.3 def firmware_VbootCrypto.firmware_VbootCrypto.__output_result_keyvals (*self*, *results*) [private]

Definition at line 60 of file firmware_VbootCrypto.py.

```
61                                            :
62         for keyval in results.splitlines():
63             if keyval.strip().startswith('#'):
64                 continue
65             key, val = keyval.split(':')
66             self.keyvals[key.strip()] = float(val)
67
```

6.13.2.4 def firmware_VbootCrypto.firmware_VbootCrypto.__rollback_tests (*self*) [private]

Definition at line 142 of file firmware_VbootCrypto.py.

```
143                               :
144         firmware_rollback_test_cmd = "cd %s && ./firmware_rollback_tests" % \
145                                      os.path.join(self.srcdir, "tests")
146         kernel_rollback_test_cmd = "cd %s && ./kernel_rollback_tests" % \
147                                      os.path.join(self.srcdir, "tests")
148         return_code = utils.system(firmware_rollback_test_cmd,
149                                    ignore_status=True)
150         if return_code == 255:
151             return False
152         if return_code == 1:
153             raise error.TestError("Firmware Rollback Test Error")
154
155         return_code = utils.system(kernel_rollback_test_cmd,
156                                    ignore_status=True)
157         if return_code == 255:
158             return False
159         if return_code == 1:
160             raise error.TestError("KernelRollback Test Error")
161         return True
162
```

6.13.2.5 def firmware_VbootCrypto.firmware_VbootCrypto.__rsa_benchmark (*self*) [private]

Definition at line 121 of file firmware_VbootCrypto.py.

```
122                               :
123         rsa_benchmark_cmd = "cd %s && ./rsa_verify_benchmark" % \
124                             os.path.join(self.srcdir, "tests")
125         self.results = utils.system_output(rsa_benchmark_cmd,
126                                            retain_output=True)
127         self.__output_result_keyvals(self.results)
128
```

6.13.2.6 def firmware_VbootCrypto.firmware_VbootCrypto.__rsa_test (*self*) [private]

Definition at line 89 of file firmware_VbootCrypto.py.

```
90                           :
91         os.chdir(self.srcdir)
92         rsa_test_cmd = os.path.join(self.srcdir, "tests",
93                                     "run_rsa_tests.sh")
94         return_code = utils.system(rsa_test_cmd, ignore_status=True)
95         if return_code == 255:
96             return False
97         if return_code == 1:
98             raise error.TestError("RSA Test Error")
99         return True
100
```

6.13.2.7 def firmware_VbootCrypto.firmware_VbootCrypto.__sha_benchmark (*self*) [private]

Definition at line 113 of file firmware_VbootCrypto.py.

```
114                             :
115         sha_benchmark_cmd = os.path.join(self.srcdir, "tests",
116                                          "sha_benchmark")
117         self.results = utils.system_output(sha_benchmark_cmd,
118                                            retain_output=True)
119         self.__output_result_keyvals(self.results)
120
```

6.13.2.8 def firmware_VbootCrypto.firmware_VbootCrypto.__sha_test (*self*) [private]

Definition at line 79 of file firmware_VbootCrypto.py.

```
80                         :
81         sha_test_cmd = os.path.join(self.srcdir, "tests", "sha_tests")
82         return_code = utils.system(sha_test_cmd, ignore_status=True)
83         if return_code == 255:
84             return False
85         if return_code == 1:
86             raise error.TestError("SHA Test Error")
87         return True
88
```

6.13.2.9 def firmware_VbootCrypto.firmware_VbootCrypto.__splicing_tests (*self*) [private]

Definition at line 163 of file firmware_VbootCrypto.py.

```
164                             :
165         firmware_splicing_test_cmd = "cd %s && ./firmware_splicing_tests" % \
166                                      os.path.join(self.srcdir, "tests")
167         kernel_splicing_test_cmd = "cd %s && ./kernel_splicing_tests" % \
168                                      os.path.join(self.srcdir, "tests")
169         return_code = utils.system(firmware_splicing_test_cmd,
170                                    ignore_status=True)
171         if return_code == 255:
172             return False
173         if return_code == 1:
174             raise error.TestError("Firmware Splicing Test Error")
175
176         return_code = utils.system(kernel_splicing_test_cmd,
177                                    ignore_status=True)
178         if return_code == 255:
179             return False
180         if return_code == 1:
181             raise error.TestError("Kernel Splicing Test Error")
182         return True
183
```

6.13.2.10 def firmware_VbootCrypto.firmware_VbootCrypto.__verify_image_benchmark (*self*) [private]

Definition at line 129 of file firmware_VbootCrypto.py.

```
130                                     :
131         firmware_benchmark_cmd = "cd %s && ./firmware_verify_benchmark" % \
```

```
132                                os.path.join(self.srcdir, "tests")
133         kernel_benchmark_cmd = "cd %s && ./kernel_verify_benchmark" % \
134                                os.path.join(self.srcdir, "tests")
135         self.results = utils.system_output(firmware_benchmark_cmd,
136                                            retain_output=True)
137         self.__output_result_keyvals(self.results)
138         self.results = utils.system_output(kernel_benchmark_cmd,
139                                            retain_output=True)
140         self.__output_result_keyvals(self.results)
141
```

6.13.2.11 def firmware_VbootCrypto.firmware_VbootCrypto.run_benchmarks (*self*)

Definition at line 199 of file firmware_VbootCrypto.py.

```
200                              :
201         self.keyvals = {}
202         self.__sha_benchmark()
203         self.__rsa_benchmark()
204         self.__verify_image_benchmark()
205         self.write_perf_keyval(self.keyvals)
206
```

6.13.2.12 def firmware_VbootCrypto.firmware_VbootCrypto.run_crypto (*self*)

Definition at line 184 of file firmware_VbootCrypto.py.

```
185                           :
186         success = self.__sha_test()
187         if not success:
188             raise error.TestFail("SHA Test Failed")
189         success = self.__rsa_test()
190         if not success:
191             raise error.TestFail("RSA Test Failed")
192
```

6.13.2.13 def firmware_VbootCrypto.firmware_VbootCrypto.run_once (*self*, *suite* = `'crypto'`)

Definition at line 219 of file firmware_VbootCrypto.py.

```
220                                    :
221         self.__generate_test_cases()
222         getattr(self, 'run_' + suite)()
```

6.13.2.14 def firmware_VbootCrypto.firmware_VbootCrypto.run_rollback (*self*)

Definition at line 207 of file firmware_VbootCrypto.py.

```
208                             :
209         success = self.__rollback_tests()
210         if not success:
211             raise error.TestFail("Rollback Tests Failed")
212
```

6.13.2.15 def firmware_VbootCrypto.firmware_VbootCrypto.run_splicing (*self*)

Definition at line 213 of file firmware_VbootCrypto.py.

```
214                           :
215         success = self.__splicing_tests()
216         if not success:
217             raise error.TestFail("Splicing Tests Failed")
218
```

6.13.2.16 def firmware_VbootCrypto.firmware_VbootCrypto.run_verification (*self*)

Definition at line 193 of file firmware_VbootCrypto.py.

```
194                                 :
195         success = self.__image_verification_test()
196         if not success:
197             raise error.TestFail("Image Verification Test Failed")
198
```

6.13.2.17 def firmware_VbootCrypto.firmware_VbootCrypto.setup (*self*)

Definition at line 54 of file firmware_VbootCrypto.py.

```
55                     :
56         os.chdir(self.srcdir)
57         utils.system('make clean all')
58
59
    # Parses the [result] and output the key-value pairs.
```

6.13.3 Member Data Documentation

6.13.3.1 firmware_VbootCrypto.firmware_VbootCrypto.keyvals

Definition at line 200 of file firmware_VbootCrypto.py.

6.13.3.2 firmware_VbootCrypto.firmware_VbootCrypto.preserve_srcdir = True [static]

Definition at line 52 of file firmware_VbootCrypto.py.

6.13.3.3 firmware_VbootCrypto.firmware_VbootCrypto.results

Definition at line 116 of file firmware_VbootCrypto.py.

6.13.3.4 int firmware_VbootCrypto.firmware_VbootCrypto.version = 1 [static]

Definition at line 51 of file firmware_VbootCrypto.py.

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

- testsource/firmware_VbootCrypto.py

6.14 gl_Bench.gl_Bench Class Reference

Public Member Functions

- def setup
- def run_once

Public Attributes

- results

Static Public Attributes

- int version = 1
- preserve_srcdir = True

6.14.1 Detailed Description

Definition at line 43 of file gl_Bench.py.

6.14.2 Member Function Documentation

6.14.2.1 def gl_Bench.gl_Bench.run_once (*self*, *options* = *''*)

Definition at line 51 of file gl_Bench.py.

```
52                                :
53       dep = 'glbench'
54       dep_dir = os.path.join(self.autodir, 'deps', dep)
55       self.job.install_pkg(dep, 'dep', dep_dir)
56
57       exefile = os.path.join(self.autodir, 'deps/glbench/glbench')
58       cmd = "X :1 & sleep 1; DISPLAY=:1 %s %s; kill $!" % (exefile, options)
59       self.results = utils.system_output(cmd, retain_output=True)
60
61       keyvals = {}
62       for keyval in self.results.splitlines():
63           if keyval.strip().startswith('#'):
64               continue
65           key, val = keyval.split(':')
66           keyvals[key.strip()] = float(val)
67
68       self.write_perf_keyval(keyvals)
```

6.14.2.2 def gl_Bench.gl_Bench.setup (*self*)

Definition at line 47 of file gl_Bench.py.

```
48                  :
49       self.job.setup_dep(['glbench'])
50
```

6.14.3 Member Data Documentation

6.14.3.1 gl_Bench.gl_Bench.preserve_srcdir = True [static]

Definition at line 45 of file gl_Bench.py.

6.14.3.2 gl_Bench.gl_Bench.results

Definition at line 58 of file gl_Bench.py.

6.14.3.3 int gl_Bench.gl_Bench.version = 1 [static]

Definition at line 44 of file gl_Bench.py.

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

- testsource/gl_Bench.py

6.15 graphics_GLAPICheck.graphics_GLAPICheck Class Reference

Public Member Functions

- def setup
- def run_once

Public Attributes

- error_message

Static Public Attributes

- int version = 1
- preserve_srcdir = True
- string error_message = ""

Private Member Functions

- def __check_extensions
- def __check_gl_extensions_1x
- def __check_gl_extensions_2x
- def __check_gles_extensions
- def __check_gl
- def __check_gles
- def __check_x_extensions
- def __run_x_cmd

6.15.1 Detailed Description

Definition at line 38 of file graphics_GLAPICheck.py.

6.15.2 Member Function Documentation

6.15.2.1 def graphics_GLAPICheck.graphics_GLAPICheck.__check_extensions (*self*, *info*, *ext_entries*) [private]

Definition at line 50 of file graphics_GLAPICheck.py.

```
51                                                  :
52         info_split = info.split()
53         comply = True
54         for extension in ext_entries:
55             match = extension in info_split
56             if not match:
57                 self.error_message += " " + extension
58                 comply = False
59         return comply
60
```

6.15.2.2 def graphics_GLAPICheck.graphics_GLAPICheck.__check_gl (*self*, *result*) [private]

Definition at line 90 of file graphics_GLAPICheck.py.

```
91                                  :
92          version = re.findall(r"GL_VERSION = ([0-9]+).([0-9]+).+", result)
93          if version:
94              version_major = int(version[0][0])
95              version_minor = int(version[0][1])
96              version_info = (" GL_VERSION = %d.%d" %
97                              (version_major, version_minor))
98              if version_major == 1:
99                  if version_minor < 4:
100                     self.error_message = version_info
101                     return False
102                 return self.__check_gl_extensions_1x(result)
103             elif version_major >= 2:
104                 return self.__check_gl_extensions_2x(result)
105             else:
106                 self.error_message = version_info
107                 return False
108         # No GL version info found.
109         self.error_message = " missing GL version info"
110         return False
111
```

6.15.2.3 def graphics_GLAPICheck.graphics_GLAPICheck.__check_gl_extensions_1x (*self*, *info*) [private]

Definition at line 61 of file graphics_GLAPICheck.py.

```
62                                          :
63          extensions = [
64              'GL_ARB_vertex_buffer_object',
65              'GL_ARB_shader_objects',
66              'GL_ARB_texture_non_power_of_two',
67              'GL_ARB_point_sprite',
68              'GL_EXT_framebuffer_object',
69              'GLX_EXT_texture_from_pixmap'
70          ]
71          return self.__check_extensions(info, extensions)
72
```

6.15.2.4 def graphics_GLAPICheck.graphics_GLAPICheck.__check_gl_extensions_2x (*self*, *info*) [private]

Definition at line 73 of file graphics_GLAPICheck.py.

```
74                                          :
75          extensions = [
76              'GL_EXT_framebuffer_object',
77              'GLX_EXT_texture_from_pixmap'
78          ]
79          return self.__check_extensions(info, extensions)
80
```

6.15.2.5 def graphics_GLAPICheck.graphics_GLAPICheck.__check_gles (*self*, *result*) [private]

Definition at line 112 of file graphics_GLAPICheck.py.

```
113                                 :
114         version = re.findall(r"GLES_VERSION = OpenGL ES.* ([0-9]+).([0-9]+)",
115                              result)
116         if version:
117             # GLES version has to be 2.0 or above.
118             version_major = int(version[0][0])
119             version_minor = int(version[0][1])
120             version_info = (" GLES_VERSION = %d.%d" %
121                             (version_major, version_minor))
122             if version_major < 2:
123                 self.error_message = version_info
124                 return False;
125             # EGL version has to be 1.3 or above.
126             version = re.findall(r"EGL_VERSION = ([0-9]+).([0-9]+)", result)
127             if version:
128                 version_major = int(version[0][0])
129                 version_minor = int(version[0][1])
130                 version_info = ("EGL_VERSION = %d.%d" %
131                                 (version_major, version_minor))
132                 if (version_major == 1 and version_minor >= 3 or
133                     version_major > 1):
134                     return self.__check_gles_extensions(result)
135                 else:
136                     self.error_message = version_info
137                     return False
138             # No EGL version info found.
139             self.error_message = " missing EGL version info"
140             return False
141         # No GLES version info found.
142         self.error_message = " missing GLES version info"
143         return False
144
```

6.15.2.6 def graphics_GLAPICheck.graphics_GLAPICheck.__check_gles_extensions (*self*, *info*) [private]

Definition at line 81 of file graphics_GLAPICheck.py.

```
82                                          :
83         extensions = [
84             'EGL_KHR_image_pixmap',
85             'GL_OES_EGL_image',
86             'GL_OES_texture_npot'
87         ]
88         return self.__check_extensions(info, extensions)
89
```

6.15.2.7 def graphics_GLAPICheck.graphics_GLAPICheck.__check_x_extensions (*self*, *result*) [private]

Definition at line 145 of file graphics_GLAPICheck.py.

```
146                                         :
147         extensions = [
148             'DAMAGE',
149             'Composite'
150         ]
151         return self.__check_extensions(result, extensions)
152
```

6.15.2.8 def graphics_GLAPICheck.graphics_GLAPICheck.__run_x_cmd (*self*, *cmd*) [private]

Definition at line 153 of file graphics_GLAPICheck.py.

```
154                                 :
155         cmd = site_ui.xcommand(cmd)
156         result = utils.system_output(cmd, retain_output=True,
157                                      ignore_status=True)
158         return result
159
```

6.15.2.9 def graphics_GLAPICheck.graphics_GLAPICheck.run_once (*self*)

Definition at line 160 of file graphics_GLAPICheck.py.

```
161                       :
162         test_done = False
163         cmd_gl = os.path.join(self.bindir, 'gl_APICheck')
164         cmd_gles = os.path.join(self.bindir, 'gles_APICheck')
165         exist_gl = os.path.isfile(cmd_gl)
166         exist_gles = os.path.isfile(cmd_gles)
167         if not exist_gl and not exist_gles:
168             raise error.TestFail('Found neither gl_APICheck nor gles_APICheck. '
169                                  'Test setup error.')
170         elif exist_gl and exist_gles:
171             raise error.TestFail('Found both gl_APICheck and gles_APICheck. '
172                                  'Test setup error.')
173         elif exist_gl:
174             self.error_message = ""
175             result = self.__run_x_cmd(cmd_gl)
176             errors = re.findall(r"ERROR: ", result)
177             run_through = re.findall(r"SUCCEED: run to the end", result)
178             if not errors and run_through:
179                 check_result = self.__check_gl(result)
180                 if not check_result:
181                     raise error.TestFail('GL API insufficient:' +
182                                          self.error_message)
183             else:
184                 raise error.TestFail('gl_APICheck error: ' + result)
185         else:
186             self.error_message = ""
187             # TODO(zmo@): smarter mechanism with GLES & EGL library names.
188             result = self.__run_x_cmd(cmd_gles + ' libGLESv2.so libEGL.so')
189             errors = re.findall(r"ERROR: ", result)
190             run_through = re.findall(r"SUCCEED: run to the end", result)
191             if not errors and run_through:
192                 check_result = self.__check_gles(result)
193                 if not check_result:
194                     raise error.TestFail('GLES API insufficient:' +
```

```
195                                         self.error_message)
196             else:
197                 raise error.TestFail('gles_APICheck error: ' + result)
198
199         # Check X11 extensions.
200         self.error_message = ""
201         check_result = self.__check_x_extensions(result)
202         if not check_result:
203             raise error.TestFail('X extensions insufficient:' +
204                                  self.error_message)
```

6.15.2.10 def graphics_GLAPICheck.graphics_GLAPICheck.setup (*self*)

Definition at line 44 of file graphics_GLAPICheck.py.

```
45                 :
46         os.chdir(self.srcdir)
47         utils.system('make clean')
48         utils.system('make all')
49
```

6.15.3 Member Data Documentation

6.15.3.1 graphics_GLAPICheck.graphics_GLAPICheck.error_message

Definition at line 99 of file graphics_GLAPICheck.py.

6.15.3.2 string graphics_GLAPICheck.graphics_GLAPICheck.error_message = "" [static]

Definition at line 41 of file graphics_GLAPICheck.py.

6.15.3.3 graphics_GLAPICheck.graphics_GLAPICheck.preserve_srcdir = True [static]

Definition at line 40 of file graphics_GLAPICheck.py.

6.15.3.4 int graphics_GLAPICheck.graphics_GLAPICheck.version = 1 [static]

Definition at line 39 of file graphics_GLAPICheck.py.

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

- testsource/graphics_GLAPICheck.py

6.16 graphics_SanAngeles.graphics_SanAngeles Class Reference

Public Member Functions

- def setup
- def run_once

Static Public Attributes

- int version = 1
- preserve_srcdir = True

6.16.1 Detailed Description

Definition at line 35 of file graphics_SanAngeles.py.

6.16.2 Member Function Documentation

6.16.2.1 def graphics_SanAngeles.graphics_SanAngeles.run_once (*self*)

Definition at line 46 of file graphics_SanAngeles.py.

```
47                      :
48          cmd_gl = os.path.join(self.srcdir, 'SanOGL')
49          cmd_gles = os.path.join(self.srcdir, 'SanOGLES')
50          cmd_gles_s = os.path.join(self.srcdir, 'SanOGLES_S')
51          if os.path.isfile(cmd_gl):
52              cmd = cmd_gl
53          elif os.path.isfile(cmd_gles):
54              cmd = cmd_gles
55          elif os.path.isfile(cmd_gles_s):
56              cmd = cmd_gles_s
57          else:
58              raise error.TestFail('Fail to locate SanAngles Observation exe.'
59                                   'Test setup error.')
60
61          cmd = site_ui.xcommand(cmd)
62          result = utils.run(cmd, ignore_status = True)
63
64          report = re.findall(r"frame_rate = ([0-9.]+)", result.stdout)
65          if len(result.stderr) > 0 or not report:
66              raise error.TestFail('Fail to complete San Angeles Observation' +
67                                   result.stderr)
68          frame_rate = float(report[0])
69          logging.info('frame_rate = %.1f' % frame_rate)
70          self.write_perf_keyval(
71              {'frames_per_sec_rate_san_angeles': frame_rate})
```

6.16.2.2 def graphics_SanAngeles.graphics_SanAngeles.setup (*self*)

Definition at line 40 of file graphics_SanAngeles.py.

```
41                      :
42          os.chdir(self.srcdir)
```

```
43         utils.system('make clean')
44         utils.system('make all')
45
```

6.16.3 Member Data Documentation

6.16.3.1 graphics_SanAngeles.graphics_SanAngeles.preserve_srcdir = True [static]

Definition at line 37 of file graphics_SanAngeles.py.

6.16.3.2 int graphics_SanAngeles.graphics_SanAngeles.version = 1 [static]

Definition at line 36 of file graphics_SanAngeles.py.

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

- testsource/graphics_SanAngeles.py

6.17 graphics_TearTest.graphics_TearTest Class Reference

Public Member Functions

- def setup
- def run_once

Static Public Attributes

- int version = 1

6.17.1 Detailed Description

Definition at line 48 of file graphics_TearTest.py.

6.17.2 Member Function Documentation

6.17.2.1 def graphics_TearTest.graphics_TearTest.run_once (*self*)

Definition at line 56 of file graphics_TearTest.py.

```
57                      :
58         dep = 'glbench'
59         dep_dir = os.path.join(self.autodir, 'deps', dep)
60         self.job.install_pkg(dep, 'dep', dep_dir)
61 
62         exefile = os.path.join(self.autodir, 'deps/glbench/teartest')
63 
64         while True:
65             tests = [
66                 dict(cmd=exefile+' --tests uniform',
67                      desc='Uniform updates', result=''),
68                 dict(cmd=exefile+' --tests teximage2d',
69                      desc='glTexImage2D updates', result=''),
70                 dict(cmd=exefile+' --tests pixmap',
71                      desc='Pixmap to texture', result=''),
72             ]
73 
74             # First, present the starting screen with one Start button.
75             header = ("These tests check vertical synchronization. You will " +
76                     "see two vertical lines scrolling horizontally. The test " +
77                     "passes if lines stay straight with no tearing.<br/>" +
78                     html_button('Start'))
79             dialog = site_ui.Dialog(question=TEMPLATE.format(header, tests),
80                                     choices=[])
81             result = dialog.get_result()
82 
83             header = html_button('Restart')
84 
85             # Run testcases from tests array.
86             for test in tests:
87                 cmd = test['cmd']
88                 logging.info("command launched: %s" % cmd)
89                 ret = utils.system(site_ui.xcommand(cmd), ignore_status=True)
90 
91                 if ret == 0:
92                     test['result'] = html_button('Pass') + html_button('Fail')
93                     dialog = site_ui.Dialog(
```

```
94                          question=TEMPLATE.format(header, tests), choices=[])
95                      # Store user's response if the testcase passed or failed.
96                      result = dialog.get_result()
97                      test['result'] = result if result else 'Timeout'
98                  else:
99                      # If test return nonzero status, mark it as failed.
100                     test['result'] = 'Fail'
101
102             # Test passed if all testcases passed.
103             passed = all(test['result'] == 'Pass' for test in tests)
104             header = ("Test %s.<br/>" % ("passed" if passed else "failed") +
105                       html_button('Done') + html_button('Restart'))
106             # Show the summary screen.
107             dialog = site_ui.Dialog(question=TEMPLATE.format(header, tests),
108                                     choices=[])
109             result = dialog.get_result()
110
111             # If user chose 'Restart', run the whole thing again.
112             if result != 'Restart':
113                 break
114
115         if not passed:
116             raise error.TestFail('Failed: ' +
117                 ', '.join(test['desc'] for test in tests
118                           if test['result'] != 'Pass'))
```

6.17.2.2 def graphics_TearTest.graphics_TearTest.setup (*self*)

Definition at line 52 of file graphics_TearTest.py.

```
53                  :
54         self.job.setup_dep(['glbench'])
55
```

6.17.3 Member Data Documentation

6.17.3.1 int graphics_TearTest.graphics_TearTest.version = 1 `[static]`

Definition at line 49 of file graphics_TearTest.py.

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

- testsource/graphics_TearTest.py

6.18 hardware_Backlight.hardware_Backlight Class Reference

Public Member Functions

- def run_once

Static Public Attributes

- int version = 1

6.18.1 Detailed Description

Definition at line 36 of file hardware_Backlight.py.

6.18.2 Member Function Documentation

6.18.2.1 def hardware_Backlight.hardware_Backlight.run_once (*self*)

Definition at line 39 of file hardware_Backlight.py.

```
40                      :
41         try:
42             brightness = int(backlight_tool("--get_brightness").rstrip())
43         except error.CmdError, e:
44             raise error.TestFail('Cannot get brightness with backlight-tool')
45         max_brightness = int(backlight_tool("--get_max_brightness").rstrip())
46         try:
47             for i in range(max_brightness + 1):
48                 backlight_tool("--set_brightness %d" % i)
49                 result = int(backlight_tool("--get_brightness").rstrip())
50                 if i != result:
51                     raise error.TestFail('Adjusting backlight should change ' \
52                                          'actual brightness')
53         finally:
54             backlight_tool("--set_brightness %d" % brightness)
```

6.18.3 Member Data Documentation

6.18.3.1 int hardware_Backlight.hardware_Backlight.version = 1 [static]

Definition at line 37 of file hardware_Backlight.py.

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

- testsource/hardware_Backlight.py

6.19 hardware_BluetoothSemiAuto.hardware_BluetoothSemiAuto Class Reference

Public Member Functions

- def initialize
- def cleanup
- def handle_reply
- def handle_error
- def do_connect
- def run_once

Public Attributes

- mainloop

Static Public Attributes

- int version = 1

6.19.1 Detailed Description

Definition at line 70 of file hardware_BluetoothSemiAuto.py.

6.19.2 Member Function Documentation

6.19.2.1 def hardware_BluetoothSemiAuto.hardware_BluetoothSemiAuto.cleanup (*self*)

Definition at line 77 of file hardware_BluetoothSemiAuto.py.

```
78                     :
79         site_ui_test.UITest.cleanup(self)
80
```

6.19.2.2 def hardware_BluetoothSemiAuto.hardware_BluetoothSemiAuto.do_connect (*self*, *addr*)

Definition at line 91 of file hardware_BluetoothSemiAuto.py.

```
92                             :
93         logging.debug("do_connect: %s", addr)
94         dbus.mainloop.glib.DBusGMainLoop(set_as_default=True)
95
96         bus = dbus.SystemBus()
97         manager = dbus.Interface(bus.get_object("org.bluez", "/"),
98                                  "org.bluez.Manager")
99
100         adapter = dbus.Interface(bus.get_object("org.bluez",
101                                                 manager.DefaultAdapter()),
102                                  "org.bluez.Adapter")
```

```
103
104
105         logging.debug("Creating Agent")
106         agent_path = "/blueztestagent"
107         try:
108             agent = Agent(bus, agent_path)
109         except Exception, e:
110             logging.debug('Unable to create an agent: %s', e)
111
112         self.mainloop = gobject.MainLoop()
113
114         try:
115             device = adapter.FindDevice(addr)
116             adapter.RemoveDevice(device)
117         except Exception, e:
118             logging.debug('Unable to find/remove device %s: %s', addr, e)
119
120         adapter.CreatePairedDevice(addr, agent_path, "DisplayOnly",
121                                    reply_handler=self.handle_reply,
122                                    error_handler=self.handle_error)
123
124         logging.debug('Starting mainloop...')
125         self.mainloop.run()
126         logging.debug('... mainloop ended.')
127
128         device = adapter.FindDevice(addr)
129         input = dbus.Interface(bus.get_object("org.bluez", device),
130                                "org.bluez.Input")
131         input.Connect()
132         logging.debug('Connected to input:%s.', addr)
133
```

6.19.2.3 def hardware_BluetoothSemiAuto.hardware_BluetoothSemiAuto.handle_error (*self*, *error*)

Definition at line 86 of file hardware_BluetoothSemiAuto.py.

```
87                                    :
88         logging.debug('Unable to create device: %s', error)
89         self.mainloop.quit()
90
```

6.19.2.4 def hardware_BluetoothSemiAuto.hardware_BluetoothSemiAuto.handle_reply (*self*, *device*)

Definition at line 81 of file hardware_BluetoothSemiAuto.py.

```
82                                    :
83         logging.debug("Device created: %s", device)
84         self.mainloop.quit()
85
```

6.19.2.5 def hardware_BluetoothSemiAuto.hardware_BluetoothSemiAuto.initialize (*self*, *creds* = `'$default'`)

Definition at line 73 of file hardware_BluetoothSemiAuto.py.

```
74                                              :
75         site_ui_test.UITest.initialize(self, creds)
76
```

6.19.2.6 def hardware_BluetoothSemiAuto.hardware_BluetoothSemiAuto.run_once (*self*)

Definition at line 134 of file hardware_BluetoothSemiAuto.py.

```
135                     :
136         question_prepend = ''
137         while True:
138             question = question_prepend + _QUESTION_START
139             hciscan = utils.system_output('hcitool scan')
140             logging.debug(hciscan)
141             for line in hciscan.split('\n'):
142                 line = line.strip()
143                 match = re.search(r'^(..:..:..:..:..:..)\s+(.*)$', line)
144                 if match:
145                     addr = match.group(1)
146                     question += '<tr>'
147                     question += ('<td>' +
148                                  (_HREF_START % addr) + addr + _HREF_END +
149                                  '</td>')
150                     question += '<td>' + match.group(2) + '</td>'
151                     question += '</tr>'
152             question += '</table><br>'
153
154             dialog = site_ui.Dialog(question=question,
155                                     choices=['Pass', 'Fail', 'Rescan'])
156             result = dialog.get_result()
157             if result is None:
158                 raise error.TestFail('Timeout')
159             if result == 'Pass':
160                 return
161             if result == 'Fail':
162                 raise error.TestFail('Unable to find Bluetooth devices')
163             if result == 'Rescan':
164                 question_prepend = ''
165                 continue
166
167             logging.debug("Connecting to %s", result)
168             try:
169                 self.do_connect(result)
170                 question_prepend = 'Paired with device %s.<br>' % result
171             except Exception, e:
172                 logging.debug('Unable to connect: %s', e)
173                 question_prepend = 'Unable to pair with device %s.<br>' % result
```

6.19.3 Member Data Documentation

6.19.3.1 hardware_BluetoothSemiAuto.hardware_BluetoothSemiAuto.mainloop

Definition at line 111 of file hardware_BluetoothSemiAuto.py.

6.19.3.2 int hardware_BluetoothSemiAuto.hardware_BluetoothSemiAuto.version = 1 `[static]`

Definition at line 71 of file hardware_BluetoothSemiAuto.py.

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

- testsource/hardware_BluetoothSemiAuto.py

6.20 hardware_Components.hardware_Components Class Reference

Public Member Functions

- def check_component
- def get_part_id_audio_codec
- def get_part_id_bios
- def get_part_id_cpu
- def get_part_id_embedded_controller
- def get_part_id_ethernet
- def get_part_id_flash_chip
- def get_part_id_storage
- def get_part_id_wireless
- def check_approved_part_id_existence
- def get_vendor_id_touchpad
- def get_vendor_id_webcam
- def pformat
- def initialize
- def run_once

Static Public Attributes

- int version = 1

Private Attributes

- _pp
- _system
- _failures
- _approved

Static Private Attributes

- list _cids
- list _pci_cids
- list _usb_cids
- string _not_present = 'Not Present'

6.20.1 Detailed Description

Definition at line 34 of file hardware_Components.py.

6.20.2 Member Function Documentation

6.20.2.1 def hardware_Components.hardware_Components.check_approved_part_id_existence (*self*, *cid*, *type*)

```
Check if there are matching vendor_id:product_id pairs on the PCI or
USB. Parameter type should be either 'pci' or 'usb'.
```

Definition at line 160 of file hardware_Components.py.

```
161                                                     :
162         """
163         Check if there are matching vendor_id:product_id pairs on the PCI or
164         USB. Parameter type should be either 'pci' or 'usb'.
165         """
166         cmd = 'sudo /usr/sbin/ls' + type + ' -d %s'
167         if not self._approved.has_key(cid):
168             raise error.TestFail('%s missing from database' % cid)
169
170         approved_devices = self._approved[cid]
171         if '*' in approved_devices:
172             self._system[cid] = [ '*' ]
173             return
174
175         for device in approved_devices:
176             try:
177                 utils.system(cmd % device)
178                 self._system[cid] = [ device ]
179                 return
180             except:
181                 pass
182         self._failures[cid] = [ 'No match' ]
183
```

6.20.2.2 def hardware_Components.hardware_Components.check_component (*self*, *comp_key*, *comp_id*)

Definition at line 61 of file hardware_Components.py.

```
62                                              :
63         self._system[comp_key] = [ comp_id ]
64
65         if not self._approved.has_key(comp_key):
66             raise error.TestFail('%s missing from database' % comp_key)
67
68         app_cids = self._approved[comp_key]
69
70         if '*' in app_cids:
71             return
72
73         if not comp_id in app_cids:
74             self._failures[comp_key] = [ comp_id ]
75
```

6.20.2.3 def hardware_Components.hardware_Components.get_part_id_audio_codec (*self*)

Definition at line 76 of file hardware_Components.py.

```
77                                      :
78          cmd = 'grep -R Codec: /proc/asound/* | head -n 1 | sed s/.\*Codec://'
79          part_id = utils.system_output(cmd).strip()
80          return part_id
81
```

6.20.2.4 def hardware_Components.hardware_Components.get_part_id_bios (*self*)

Definition at line 82 of file hardware_Components.py.

```
83                                  :
84          cmd = ('dmidecode | grep -A 2 "BIOS Information" | tail -2 '
85                 '| sed "s/.*: //" | tr "\n" " "')
86          part_id = utils.system_output(cmd).strip()
87
88          cmd = ('dmidecode | grep "\(BIOS\|Firmware\) Revision" | sed "s/\t//" '
89                 '| sed "s/Revision/Rev/"')
90          rev_num = ', '.join(utils.system_output(cmd).split('\n'))
91
92          if rev_num:
93              part_id = part_id + ' (' + rev_num + ')'
94
95          return part_id
96
```

6.20.2.5 def hardware_Components.hardware_Components.get_part_id_cpu (*self*)

Definition at line 97 of file hardware_Components.py.

```
98                                  :
99          cmd = 'grep -m 1 \'model name\' /proc/cpuinfo | sed s/.\*://'
100         part_id = utils.system_output(cmd).strip()
101         return part_id
102
```

6.20.2.6 def hardware_Components.hardware_Components.get_part_id_embedded_controller (*self*)

Definition at line 103 of file hardware_Components.py.

```
104                                                 :
105         # example output:
106         #  Found Nuvoton WPCE775x (id=0x05, rev=0x02) at 0x2e
107         parts = []
108         res = utils.system_output('superiotool', ignore_status=True).split('\n')
109         for line in res:
110           match = re.search(r'Found (.*) at', line)
111           if match:
112             parts.append(match.group(1))
113         part_id = ", ".join(parts)
114         return part_id
115
```

6.20.2.7 def hardware_Components.hardware_Components.get_part_id_ethernet (*self*)

```
  Returns a colon delimited string where the first section
  is the vendor id and the second section is the device id.
```

Definition at line 116 of file hardware_Components.py.

```
117                                 :
118         """
119           Returns a colon delimited string where the first section
120           is the vendor id and the second section is the device id.
121         """
122         # Ethernet is optional so mark it as not present. A human
123         # operator needs to decide if this is acceptable or not.
124         if not os.path.exists('/sys/class/net/eth0'):
125             return self._not_present
126         part_id = utils.read_one_line('/sys/class/net/eth0/device/device')
127         vendor_id = utils.read_one_line('/sys/class/net/eth0/device/vendor')
128         return "%s:%s" % (vendor_id.replace('0x',''), part_id.replace('0x',''))
129
```

6.20.2.8 def hardware_Components.hardware_Components.get_part_id_flash_chip (*self*)

Definition at line 130 of file hardware_Components.py.

```
131                                   :
132         # example output:
133         #  Found chip "Winbond W25x16" (2048 KB, FWH) at physical address 0xfe
134         parts = []
135         lines = utils.system_output('flashrom', ignore_status=True).split('\n')
136         for line in lines:
137           match = re.search(r'Found chip "(.*)" .* at physical address ', line)
138           if match:
139             parts.append(match.group(1))
140         part_id = ", ".join(parts)
141         return part_id
142
```

6.20.2.9 def hardware_Components.hardware_Components.get_part_id_storage (*self*)

Definition at line 143 of file hardware_Components.py.

```
144                                 :
145         cmd = ('cd $(find /sys/devices -name sda)/../..; '
146                'cat vendor model | tr "\n" " " | sed "s/ \+/ /g"')
147         part_id = utils.system_output(cmd).strip()
148         return part_id
149
```

6.20.2.10 def hardware_Components.hardware_Components.get_part_id_wireless (*self*)

```
  Returns a colon delimited string where the first section
  is the vendor id and the second section is the device id.
```

Definition at line 150 of file hardware_Components.py.

```
151                                     :
152         """
153           Returns a colon delimited string where the first section
154           is the vendor id and the second section is the device id.
155         """
156         part_id = utils.read_one_line('/sys/class/net/wlan0/device/device')
157         vendor_id = utils.read_one_line('/sys/class/net/wlan0/device/vendor')
158         return "%s:%s" % (vendor_id.replace('0x',''), part_id.replace('0x',''))
159
```

6.20.2.11 def hardware_Components.hardware_Components.get_vendor_id_touchpad (*self*)

Definition at line 184 of file hardware_Components.py.

```
185                                     :
186         cmd = 'grep -i Touchpad /proc/bus/input/devices | sed s/.\*=//'
187         part_id = utils.system_output(cmd).strip('"')
188         return part_id
189
```

6.20.2.12 def hardware_Components.hardware_Components.get_vendor_id_webcam (*self*)

Definition at line 190 of file hardware_Components.py.

```
191                                     :
192         cmd = 'cat /sys/class/video4linux/video0/name'
193         part_id = utils.system_output(cmd).strip()
194         return part_id
195
```

6.20.2.13 def hardware_Components.hardware_Components.initialize (*self*)

Definition at line 200 of file hardware_Components.py.

```
201                         :
202         self._pp = pprint.PrettyPrinter()
203
```

6.20.2.14 def hardware_Components.hardware_Components.pformat (*self*, *obj*)

Definition at line 196 of file hardware_Components.py.

```
197                           :
198         return "\n" + self._pp.pformat(obj) + "\n"
199
```

6.20.2.15 def hardware_Components.hardware_Components.run_once (*self*, *approved_db* = 'approved_components')

Definition at line 204 of file hardware_Components.py.

```
205                                                         :
206         self._system = {}
207         self._failures = {}
208 
209         approved_db = os.path.join(self.bindir, approved_db)
210         if not os.path.exists(approved_db):
211             raise error.TestError('Unable to find approved_db: %s' %
212                                   approved_db)
213 
214         self._approved = eval(utils.read_file(approved_db))
215         logging.debug('Approved DB: %s', self.pformat(self._approved))
216 
217         for cid in self._cids:
218             self.check_component(cid, getattr(self, 'get_' + cid)())
219 
220         for cid in self._pci_cids:
221             self.check_approved_part_id_existence(cid, type='pci')
222 
223         for cid in self._usb_cids:
224             self.check_approved_part_id_existence(cid, type='usb')
225 
226         logging.debug('System: %s', self.pformat(self._system))
227 
228         outdb = os.path.join(self.resultsdir, 'system_components')
229         utils.open_write_close(outdb, self.pformat(self._system))
230 
231         if self._failures:
232             raise error.TestFail(self.pformat(self._failures))
```

6.20.3 Member Data Documentation

6.20.3.1 hardware_Components.hardware_Components._approved [private]

Definition at line 213 of file hardware_Components.py.

6.20.3.2 list hardware_Components.hardware_Components._cids [static, private]

Initial value:

```
[
        'part_id_audio_codec',
        'part_id_bios',
        'part_id_cpu',
        'part_id_embedded_controller',
        'part_id_ethernet',
        'part_id_flash_chip',
        'part_id_storage',
        'part_id_wireless',
        'vendor_id_touchpad',
    ]
```

Definition at line 36 of file hardware_Components.py.

6.20.3.3 hardware_Components.hardware_Components._failures [private]

Definition at line 206 of file hardware_Components.py.

6.20.3.4 string hardware_Components.hardware_Components._not_present = 'Not Present' [static, private]

Definition at line 58 of file hardware_Components.py.

6.20.3.5 list hardware_Components.hardware_Components._pci_cids [static, private]

Initial value:

```
[
        'part_id_chipset',
        'part_id_usb_hosts',
        'part_id_vga',
    ]
```

Definition at line 47 of file hardware_Components.py.

6.20.3.6 hardware_Components.hardware_Components._pp [private]

Definition at line 201 of file hardware_Components.py.

6.20.3.7 hardware_Components.hardware_Components._system [private]

Definition at line 205 of file hardware_Components.py.

6.20.3.8 list hardware_Components.hardware_Components._usb_cids [static, private]

Initial value:

```
[
        'part_id_bluetooth',
        'part_id_cardreader',
        'part_id_webcam',
        'part_id_3g',
    ]
```

Definition at line 52 of file hardware_Components.py.

6.20.3.9 int hardware_Components.hardware_Components.version = 1 [static]

Definition at line 35 of file hardware_Components.py.

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

- testsource/hardware_Components.py

6.21 hardware_DiskSize.hardware_DiskSize Class Reference

Public Member Functions

- def run_once

Static Public Attributes

- int version = 1

6.21.1 Detailed Description

Definition at line 35 of file hardware_DiskSize.py.

6.21.2 Member Function Documentation

6.21.2.1 def hardware_DiskSize.hardware_DiskSize.run_once (*self*)

Definition at line 38 of file hardware_DiskSize.py.

```
39                     :
40         cmdline = file('/proc/cmdline').read()
41         match = re.search(r'root=/dev/([^ ]+)', cmdline)
42         if not match:
43             raise error.TestError('Unable to find the root partition')
44         device = match.group(1)[:-1]
45
46         for line in file('/proc/partitions'):
47             try:
48                 major, minor, blocks, name = re.split(r' +', line.strip())
49             except ValueError:
50                 continue
51             # TODO(waihong@): Check if this works on ARM.
52             if name == device:
53                 blocks = int(blocks)
54                 break
55         else:
56             raise error.TestError('Unable to determine main disk size')
57
58         # Capacity of a hard disk is quoted with SI prefixes, incrementing by
59         # powers of 1000, instead of powers of 1024.
60         gb = blocks * 1024.0 / 1000.0 / 1000.0 / 1000.0
61         self.write_perf_keyval({"gb_main_disk_size": gb})
62         logging.info("DiskSize: %.3f GB" % gb)
```

6.21.3 Member Data Documentation

6.21.3.1 int hardware_DiskSize.hardware_DiskSize.version = 1 [static]

Definition at line 36 of file hardware_DiskSize.py.

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

- testsource/hardware_DiskSize.py

6.22 hardware_KeyboardAssembly.hardware_KeyboardAssembly Class Reference

Public Member Functions

- def run_once

Static Public Attributes

- int version = 1
- preserve_srcdir = True

6.22.1 Detailed Description

Definition at line 36 of file hardware_KeyboardAssembly.py.

6.22.2 Member Function Documentation

6.22.2.1 def hardware_KeyboardAssembly.hardware_KeyboardAssembly.run_once (*self*, *restart_ui* = `False`)

Definition at line 41 of file hardware_KeyboardAssembly.py.

```
42                                          :
43
44         # kill chrome
45         utils.system('/sbin/initctl stop ui', ignore_status=True)
46
47         os.chdir(self.srcdir)
48         args = ''
49         if restart_ui:
50             args += '--exit-on-error'
51         status = utils.system('./start_test.sh ' + args, ignore_status=True)
52
53         if restart_ui:
54             utils.system('/sbin/initctl start ui', ignore_status=True)
55
56         if status:
57             raise error.TestFail('Test failed.')
```

6.22.3 Member Data Documentation

6.22.3.1 hardware_KeyboardAssembly.hardware_KeyboardAssembly.preserve_srcdir = True `[static]`

Definition at line 38 of file hardware_KeyboardAssembly.py.

6.22.3.2 int hardware_KeyboardAssembly.hardware_KeyboardAssembly.version = 1 `[static]`

Definition at line 37 of file hardware_KeyboardAssembly.py.

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

- testsource/hardware_KeyboardAssembly.py

6.23 hardware_MemoryThroughput.hardware_-MemoryThroughput Class Reference

Public Member Functions

- def setup
- def run_once

Public Attributes

- results

Static Public Attributes

- int version = 1
- preserve_srcdir = True

6.23.1 Detailed Description

Definition at line 37 of file hardware_MemoryThroughput.py.

6.23.2 Member Function Documentation

6.23.2.1 def hardware_MemoryThroughput.hardware_MemoryThroughput.run_once (*self*, *num_iteration* = −1, *test_list* = ″)

Definition at line 46 of file hardware_MemoryThroughput.py.

```
47                                                          :
48         exefile = os.path.join(self.srcdir, 'hardware_MemoryThroughput')
49         cmd = '%s %d %s' % (exefile, num_iteration, test_list)
50         self.results = utils.system_output(cmd, retain_output = True)
51
52         # Resulting time in MicroSec / MegaBytes.
53         # Write out memory operation performance in MegaBytes / Second.
54         performance_pattern = re.compile(
55             r"Action = ([a-z0-9.]+), BlockSize = (\w+), " +
56             r"Method = (\w+), Time = ([0-9.]+)")
57         keyval_list = performance_pattern.findall(self.results)
58         keyvals = {}
59         for keyval in keyval_list:
60             key = ('mb_per_sec_memory_' +
61                    keyval[0] + '_' + keyval[1] + '_' + keyval[2])
62             keyvals[key] = 1000000.0 / float(keyval[3])
63         self.write_perf_keyval(keyvals)
64
65         # Detect if an error has occured during the tests.
66         # Do this after writing out the test results so even an error occurred,
67         # we still get the performance evaluation.
68         error_pattern = re.compile(r"ERROR: \[(.+)\]")
69         errors = error_pattern.findall(self.results)
70         if len(errors) > 0:
71             logging.debug(self.results)
72             raise error.TestFail('malfunctioning memory detected');
```

6.23.2.2 def hardware_MemoryThroughput.hardware_MemoryThroughput.setup (*self*)

Definition at line 41 of file hardware_MemoryThroughput.py.

```
42                     :
43           os.chdir(self.srcdir)
44           utils.system('make clean')
45           utils.system('make')
```

6.23.3 Member Data Documentation

6.23.3.1 hardware_MemoryThroughput.hardware_MemoryThroughput.preserve_srcdir = True [static]

Definition at line 39 of file hardware_MemoryThroughput.py.

6.23.3.2 hardware_MemoryThroughput.hardware_MemoryThroughput.results

Definition at line 49 of file hardware_MemoryThroughput.py.

6.23.3.3 int hardware_MemoryThroughput.hardware_MemoryThroughput.version = 1 [static]

Definition at line 38 of file hardware_MemoryThroughput.py.

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

- testsource/hardware_MemoryThroughput.py

6.24 hardware_MemoryTotalSize.hardware_MemoryTotalSize Class Reference

Public Member Functions

- def run_once

Static Public Attributes

- int version = 1

6.24.1 Detailed Description

Definition at line 31 of file hardware_MemoryTotalSize.py.

6.24.2 Member Function Documentation

6.24.2.1 def hardware_MemoryTotalSize.hardware_MemoryTotalSize.run_once (*self*)

Definition at line 34 of file hardware_MemoryTotalSize.py.

```
35                      :
36          # TODO(zmo@): this may not get total physical memory size on ARM
37          #             or some x86 machines.
38          mem_size = utils.memtotal()
39          gb = mem_size / 1024.0 / 1024.0
40          self.write_perf_keyval({"gb_memory_total": gb})
41          logging.info("MemTotal: %.3f GB" % gb)
42
43          # We intend to check if a machine has at least 1G memory.  However,
44          # taking into consideration that some machines reserve certain amount
45          # of memory and these won't show in '/proc/meminfo', we lower the
46          # threshold to 0.75 Gb.  Hopefully the reserved memory size is less
47          # than 256 Mb.
48          if gb <= 0.75:
49              raise error.TestFail("total system memory size < 1G");
```

6.24.3 Member Data Documentation

6.24.3.1 int hardware_MemoryTotalSize.hardware_MemoryTotalSize.version = 1 [static]

Definition at line 32 of file hardware_MemoryTotalSize.py.

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

- testsource/hardware_MemoryTotalSize.py

6.25 hardware_Resolution.hardware_Resolution Class Reference

Public Member Functions

- def get_resolution
- def run_once

Static Public Attributes

- int version = 1

6.25.1 Detailed Description

```
Verify the current screen resolution is supported.
```

Definition at line 40 of file hardware_Resolution.py.

6.25.2 Member Function Documentation

6.25.2.1 def hardware_Resolution.hardware_Resolution.get_resolution (*self*)

```
Get the current video resolution.
Returns:
    string: represents the video resolution.
```

Definition at line 46 of file hardware_Resolution.py.

```
47                              :
48         """
49         Get the current video resolution.
50         Returns:
51             string: represents the video resolution.
52         """
53         cmd = 'xrandr'
54         # TODO:remove oldxauth when slim is deprecated.
55         oldxauth = '/var/run/slim.auth'
56         newxauth = '/home/chronos/.Xauthority'
57         # The new login manager uses XAUTHORITY=/home/chronos/.Xauthority
58         # so we need to check which file to use.
59         if os.path.isfile(oldxauth):
60             xauth = oldxauth
61         else:
62             xauth = newxauth
63 
64         environment = 'DISPLAY=:0.0 XAUTHORITY=%s' % xauth
65         output = utils.system_output('%s %s' % (environment, cmd))
66 
67         linesout = output.split('\n')
68         for line in linesout:
69             if 'Screen 0' in line:
70                 sections = line.split(',')
71                 for item in sections:
72                     if 'current' in item:
73                         res = item.split()
74                         return '%s%s%s' % (res[1], res[2], res[3])
75 
76         return None
```

```
77
```

6.25.2.2 def hardware_Resolution.hardware_Resolution.run_once (*self*)

Definition at line 78 of file hardware_Resolution.py.

```
79                      :
80
81         supported_resolutions = ['1280x800', '1366x768']
82         res = self.get_resolution()
83
84         if res not in supported_resolutions:
85             raise error.TestFail('%s is not a supported resoltuion' % res)
```

6.25.3 Member Data Documentation

6.25.3.1 int hardware_Resolution.hardware_Resolution.version = 1 `[static]`

Definition at line 44 of file hardware_Resolution.py.

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

- testsource/hardware_Resolution.py

6.26 hardware_SAT.hardware_SAT Class Reference

Public Member Functions

- def setup
- def run_once

Static Public Attributes

- int version = 1

6.26.1 Detailed Description

Definition at line 38 of file hardware_SAT.py.

6.26.2 Member Function Documentation

6.26.2.1 def hardware_SAT.hardware_SAT.run_once (*self*, *seconds* = 60)

Definition at line 65 of file hardware_SAT.py.

```
66                                      :
67          # Allow shmem access to all of memory. This is used for 32 bit
68          # access to > 1.4G. Virtual address space limitation prevents
69          # directly mapping the memory.
70          utils.run('mount -o remount,size=100% /dev/shm')
71          cpus = max(utils.count_cpus(), 1)
72          mbytes = max(int(utils.freememtotal() * .95 / 1024), 512)
73          # SAT should use as much memory as possible, while still
74          # avoiding OOMs and allowing the kernel to run, so that
75          # the maximum amoun tof memory can be tested.
76          args = ' -M %d' % mbytes  # megabytes to test
77          # The number of seconds under test can be chosen to fit into
78          # manufacturing or test flow. 60 seconds gives several
79          # passes and several patterns over each memory location
80          # and should catch clearly fautly memeory. 4 hours
81          # is an effective runin test, to catch lower frequency errors.
82          args += ' -s %d' % seconds  # seconds to run
83          # One memory copy thread per CPU should keep the memory bus
84          # as saturated as possible, while keeping each CPU busy as well.
85          args += ' -m %d' % cpus  # memory copy threads.
86          # SSE copy and checksum increases the rate at which the CPUs
87          # can drive memory, as well as stressing the CPU.
88          args += ' -W'  # Use SSE optimizatin in memory threads.
89          # File IO threads allow stressful transactions over the
90          # south bridge and SATA, as well as potentially finding SSD
91          # or disk cache problems. Two threads ensure multiple
92          # outstanding transactions to the disk, if supported.
93          args += ' -f sat.diskthread.a'  # disk thread
94          args += ' -f sat.diskthread.b'
95
96          os.chdir(os.path.join(self.srcdir, 'src'))
97          sat = utils.run('./stressapptest' + args)
98          logging.debug(sat.stdout)
99          if not re.search('Status: PASS', sat.stdout):
100             raise error.TestFail(sat.stdout)
```

6.26.2.2 def hardware_SAT.hardware_SAT.setup (*self*, *tarball* = 'stressapptest-1.0.3_-autoconf.tar.gz')

Definition at line 42 of file hardware_SAT.py.

```
43                                                                    :
44         # clean
45         if os.path.exists(self.srcdir):
46             utils.system('rm -rf %s' % self.srcdir)
47
48         tarball = utils.unmap_url(self.bindir, tarball, self.tmpdir)
49         utils.extract_tarball_to_dir(tarball, self.srcdir)
50
51         self.job.setup_dep(['libaio'])
52         ldflags = '-L' + self.autodir + '/deps/libaio/lib'
53         cflags = '-I' + self.autodir + '/deps/libaio/include'
54         # Add paths to libaio files.
55         var_flags = 'LDFLAGS="' + ldflags + '"'
56         var_flags += ' CXXFLAGS="' + cflags + '"'
57         var_flags += ' CFLAGS="' + cflags + '"'
58         var_flags += ' LIBS="-static -laio"'
59
60         os.chdir(self.srcdir)
61         # ./configure stores relevant path and environment variables.
62         utils.configure(configure=var_flags + ' ./configure')
63         utils.system('make -j %d' % utils.count_cpus())
64
```

6.26.3 Member Data Documentation

6.26.3.1 int hardware_SAT.hardware_SAT.version = 1 [static]

Definition at line 39 of file hardware_SAT.py.

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

- testsource/hardware_SAT.py

6.27 hardware_SsdDetection.hardware_SsdDetection Class Reference

Public Member Functions

- def setup
- def run_once

Static Public Attributes

- int version = 1

6.27.1 Detailed Description

Definition at line 32 of file hardware_SsdDetection.py.

6.27.2 Member Function Documentation

6.27.2.1 def hardware_SsdDetection.hardware_SsdDetection.run_once (*self*)

Definition at line 42 of file hardware_SsdDetection.py.

```
43                     :
44         # TODO(ericli): need to find a general solution to install dep packages
45         # when tests are pre-compiled, so setup() is not called from client any
46         # more.
47         dep = 'hdparm'
48         dep_dir = os.path.join(self.autodir, 'deps', dep)
49         self.job.install_pkg(dep, 'dep', dep_dir)
50
51         cmdline = file('/proc/cmdline').read()
52         match = re.search(r'root=([^ ]+)', cmdline)
53         if not match:
54             raise error.TestError('Unable to find the root partition')
55         device = match.group(1)[:-1]
56
57         path = self.autodir + '/deps/hdparm/sbin/'
58         hdparm = utils.run(path + 'hdparm -I %s' % device)
59
60         match = re.search(r'Nominal Media Rotation Rate: (.+)$',
61                           hdparm.stdout, re.MULTILINE)
62         if match and match.group(1):
63             if match.group(1) != 'Solid State Device':
64                 raise error.TestFail('The main disk is not a SSD, '
65                     'Rotation Rate: %s' % match.group(1))
66         else:
67             raise error.TestNAError(
68                 'Rotation Rate not reported from the device, '
69                 'unable to ensure it is a SSD')
```

6.27.2.2 def hardware_SsdDetection.hardware_SsdDetection.setup (*self*)

Definition at line 35 of file hardware_SsdDetection.py.

```
36                     :
37         self.job.setup_dep(['hdparm'])
38         # create a empty srcdir to prevent the error that checks .version file
39         if not os.path.exists(self.srcdir):
40             utils.system('mkdir %s' % self.srcdir)
41
```

6.27.3 Member Data Documentation

6.27.3.1 int hardware_SsdDetection.hardware_SsdDetection.version = 1 [static]

Definition at line 33 of file hardware_SsdDetection.py.

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

- testsource/hardware_SsdDetection.py

6.28 hardware_StorageFio.hardware_StorageFio Class Reference

Public Member Functions

- def setup
- def initialize
- def run_once

Static Public Attributes

- int version = 3

Private Member Functions

- def __find_free_root_partition
- def __get_file_size
- def __get_device_description
- def __parse_fio
- def __RunFio

Private Attributes

- __filename
- __filesize
- __description

6.28.1 Detailed Description

Definition at line 38 of file hardware_StorageFio.py.

6.28.2 Member Function Documentation

6.28.2.1 def hardware_StorageFio.hardware_StorageFio.__find_free_root_partition (*self*) [private]

```
Locate the spare root partition that we didn't boot off
```

Definition at line 61 of file hardware_StorageFio.py.

```
62                                          :
63         """Locate the spare root partition that we didn't boot off"""
64
65         spare_root_map = {
66             '3': '5',
67             '5': '3',
68         }
69         rootdev = utils.system_output('rootdev')
70         spare_root = rootdev[:-1] + spare_root_map[rootdev[-1]]
71         self.__filename = spare_root
72
```

6.28.2.2 def hardware_StorageFio.hardware_StorageFio.__get_device_description (*self*) [private]

Get the device vendor and model name as its description

Definition at line 97 of file hardware_StorageFio.py.

```
98                                      :
99         """Get the device vendor and model name as its description"""
100
101         # Find the block device in sysfs. For example, a card read device may
102         # be in /sys/devices/pci0000:00/0000:00:1d.7/usb1/1-5/1-5:1.0/host4/
103         # target4:0:0/4:0:0:0/block/sdb.
104         # Then read the vendor and model name in its grand-parent directory.
105         device = os.path.basename(self.__filename[:-1])
106         findsys = utils.run('find /sys/devices -name %s' % device)
107         path = findsys.stdout.rstrip()
108
109         vendor = file(path.replace('block/%s' % device,
110                                    'vendor')).read().strip()
111         model = file(path.replace('block/%s' % device, 'model')).read().strip()
112         self.__description = vendor + ' ' + model
113
```

6.28.2.3 def hardware_StorageFio.hardware_StorageFio.__get_file_size (*self*) [private]

Return the size in bytes of the device pointed to by __filename

Definition at line 73 of file hardware_StorageFio.py.

```
74                                :
75         """Return the size in bytes of the device pointed to by __filename"""
76
77         device = os.path.basename(self.__filename)
78         for line in file('/proc/partitions'):
79             try:
80                 major, minor, blocks, name = re.split(r' +', line.strip())
81             except ValueError:
82                 continue
83             if name == device:
84                 blocks = int(blocks)
85                 self.__filesize = 1024 * blocks
86                 break
87         else:
88             if device[:-1] in ['sda', 'mmcblk0p', 'mmcblk1p']:
89                 raise error.TestError(
90                     'Unable to determine free partitions size')
91             else:
92                 raise error.TestNAError(
93                     'Unable to find the partition %s, please plug in a USB '
94                     'flash drive and a SD card for testing external storage' %
95                     self.__filename)
96
```

6.28.2.4 def hardware_StorageFio.hardware_StorageFio.__parse_fio (*self*, *lines*) [private]

Parse the human readable fio output

```
This only collects bandwidth and iops numbers from fio.
```

Definition at line 114 of file hardware_StorageFio.py.

```
115                                 :
116         """Parse the human readable fio output
117
118         This only collects bandwidth and iops numbers from fio.
119
120         """
121
122         # fio --minimal doesn't output information about the number of ios
123         # that occurred, making it unsuitable for this test.  Instead we parse
124         # the human readable output with some regular expressions
125         read_re = re.compile(r'read :.*bw=([0-9]*K?)B/s.*iops=([0-9]*)')
126         write_re = re.compile(r'write:.*bw=([0-9]*K?)B/s.*iops=([0-9]*)')
127
128         results = {}
129         for line in lines.split('\n'):
130             line = line.rstrip()
131             match = read_re.search(line)
132             if match:
133                 results['read_bw'] = match.group(1)
134                 results['read_iops'] = match.group(2)
135                 continue
136             match = write_re.search(line)
137             if match:
138                 results['write_bw'] = match.group(1)
139                 results['write_iops'] = match.group(2)
140                 continue
141
142         # Turn the values into numbers
143         for metric, result in results.iteritems():
144             if result[-1] == 'K':
145                 result = int(result[:-1]) * 1024
146             else:
147                 result = int(result)
148             results[metric] = result
149
150         results['bw'] = (results.get('read_bw', 0) +
151                          results.get('write_bw', 0))
152         results['iops'] = (results.get('read_iops', 0) +
153                            results.get('write_iops', 0))
154         return results
155
```

6.28.2.5 def hardware_StorageFio.hardware_StorageFio.__RunFio (*self*, *test*) [private]

Definition at line 156 of file hardware_StorageFio.py.

```
157                             :
158         os.chdir(self.srcdir)
159         vars = 'LD_LIBRARY_PATH="' + self.autodir + '/deps/libaio/lib"'
160         os.putenv('FILENAME', self.__filename)
161         os.putenv('FILESIZE', str(self.__filesize))
162         fio = utils.run(vars + ' ./fio "%s"' % os.path.join(self.bindir, test))
163         logging.debug(fio.stdout)
164         return self.__parse_fio(fio.stdout)
165
```

6.28.2.6 def hardware_StorageFio.hardware_StorageFio.initialize (*self*, *dev = ''*)

Definition at line 166 of file hardware_StorageFio.py.

```
167                                 :
168         if dev in ['', '/dev/sda', '/dev/mmcblk0', '/dev/mmcblk1']:
169             self.__find_free_root_partition()
170         else:
171             # Use the first partition of the external drive
172             if dev[5:7] == 'sd':
173                 self.__filename = dev + '1'
174             else:
175                 self.__filename = dev + 'p1'
176         self.__get_file_size()
177         self.__get_device_description()
178 
179         # Restrict test to using 1GiB
180         self.__filesize = min(self.__filesize, 1024 * 1024 * 1024)
181 
```

6.28.2.7 def hardware_StorageFio.hardware_StorageFio.run_once (*self*, *dev = ''*)

Definition at line 182 of file hardware_StorageFio.py.

```
183                                 :
184         # TODO(ericli): need to find a general solution to install dep packages
185         # when tests are pre-compiled, so setup() is not called from client any
186         # more.
187         dep = 'libaio'
188         dep_dir = os.path.join(self.autodir, 'deps', dep)
189         self.job.install_pkg(dep, 'dep', dep_dir)
190 
191         if dev in ['', '/dev/sda', '/dev/mmcblk0', '/dev/mmcblk1']:
192             requirements = {
193                 'surfing': 'iops',
194                 'boot': 'bw',
195                 'login': 'bw',
196                 'seq_read': 'bw',
197                 'seq_write': 'bw',
198                 '16k_read': 'iops',
199                 '16k_write': 'iops',
200                 '8k_read': 'iops',
201                 '8k_write': 'iops',
202                 '4k_read': 'iops',
203                 '4k_write': 'iops',
204             }
205         else:
206             # TODO(waihong@): Add more test cases for external storage
207             requirements = {
208                 'seq_read': 'bw',
209                 'seq_write': 'bw',
210                 '16k_read': 'iops',
211                 '16k_write': 'iops',
212                 '8k_read': 'iops',
213                 '8k_write': 'iops',
214                 '4k_read': 'iops',
215                 '4k_write': 'iops',
216             }
217 
218         results = {}
219         for test, metric in requirements.iteritems():
220             result = self.__RunFio(test)
```

```
221             units = metric
222             if metric == 'bw':
223                 units = 'bytes_per_sec'
224             results[units + '_' + test] = result[metric]
225
226         # Output keys relevent to the performance, larger filesize will run
227         # slower, and sda5 should be slightly slower than sda3 on a rotational
228         # disk
229         self.write_test_keyval({'filesize': self.__filesize,
230                                 'filename': self.__filename,
231                                 'device': self.__description})
232         logging.info('Device Description: %s' % self.__description)
233         self.write_perf_keyval(results)
```

6.28.2.8 def hardware_StorageFio.hardware_StorageFio.setup (*self*, *tarball* = 'fio-1.36.tar.bz2')

Definition at line 42 of file hardware_StorageFio.py.

```
43                                                  :
44         # clean
45         if os.path.exists(self.srcdir):
46             utils.system('rm -rf %s' % self.srcdir)
47
48         tarball = utils.unmap_url(self.bindir, tarball, self.tmpdir)
49         utils.extract_tarball_to_dir(tarball, self.srcdir)
50
51         self.job.setup_dep(['libaio'])
52         ldflags = '-L' + self.autodir + '/deps/libaio/lib'
53         cflags = '-I' + self.autodir + '/deps/libaio/include'
54         var_ldflags = 'LDFLAGS="' + ldflags + '"'
55         var_cflags  = 'CFLAGS="' + cflags + '"'
56
57         os.chdir(self.srcdir)
58         utils.system('patch -p1 < ../Makefile.patch')
59         utils.system('%s %s make' % (var_ldflags, var_cflags))
60
```

6.28.3 Member Data Documentation

6.28.3.1 hardware_StorageFio.hardware_StorageFio.__description [private]

Definition at line 111 of file hardware_StorageFio.py.

6.28.3.2 hardware_StorageFio.hardware_StorageFio.__filename [private]

Definition at line 70 of file hardware_StorageFio.py.

6.28.3.3 hardware_StorageFio.hardware_StorageFio.__filesize [private]

Definition at line 84 of file hardware_StorageFio.py.

6.28.3.4 int hardware_StorageFio.hardware_StorageFio.version = 3 [static]

Definition at line 39 of file hardware_StorageFio.py.

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

- testsource/hardware_StorageFio.py

6.29 hardware_Touchpad.hardware_Touchpad Class Reference

Public Member Functions

- def run_once

Static Public Attributes

- int version = 1
- preserve_srcdir = True

6.29.1 Detailed Description

Definition at line 33 of file hardware_Touchpad.py.

6.29.2 Member Function Documentation

6.29.2.1 def hardware_Touchpad.hardware_Touchpad.run_once (*self*, *restart_ui* = `False`)

Definition at line 38 of file hardware_Touchpad.py.

```
39                                          :
40
41         # kill chrome
42         utils.system('/sbin/initctl stop ui', ignore_status=True)
43
44         os.chdir(self.srcdir)
45         args = ''
46         if restart_ui:
47             args += '--exit-on-error'
48         status = utils.system('./start_test.sh ' + args, ignore_status=True)
49
50         if restart_ui:
51             utils.system('/sbin/initctl start ui', ignore_status=True)
52
53         if status:
54             raise error.TestFail('Test failed.')
```

6.29.3 Member Data Documentation

6.29.3.1 hardware_Touchpad.hardware_Touchpad.preserve_srcdir = True `[static]`

Definition at line 35 of file hardware_Touchpad.py.

6.29.3.2 int hardware_Touchpad.hardware_Touchpad.version = 1 `[static]`

Definition at line 34 of file hardware_Touchpad.py.

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

- testsource/hardware_Touchpad.py

6.30 hardware_VideoOutSemiAuto.hardware_VideoOutSemiAuto Class Reference

Public Member Functions

- def run_once

Static Public Attributes

- int version = 1
- string XRANDR_PATH = "/usr/bin/xrandr"
- string RECONFIG_PATH = "/usr/sbin/monitor_reconfigure"
- string HDMI_ID = "HDMI"
- string VGA_ID = "VGA"

Private Member Functions

- def __query_for_output
- def __output_connected
- def __output_is_set
- def __configure_and_check_output

6.30.1 Detailed Description

Definition at line 38 of file hardware_VideoOutSemiAuto.py.

6.30.2 Member Function Documentation

6.30.2.1 def hardware_VideoOutSemiAuto.hardware_VideoOutSemiAuto.__configure_and_check_output (*self*, *output*) [private]

Definition at line 98 of file hardware_VideoOutSemiAuto.py.

```
99                                                  :
100         connected = self.__output_connected(output)
101         if not connected:
102             logging.warning(
103                 "%s port detected but no connected device" % output
104                 )
105             return False
106         else:
107             #TODO(sosa@chromium.org) - Verify this is synchronous.
108             utils.system(site_ui.xcommand(self.RECONFIG_PATH))
109             self.__output_is_set(output)
110             return True
111
```

6.30.2.2 def hardware_VideoOutSemiAuto.hardware_VideoOutSemiAuto.__output_connected (*self*, *output*) [private]

Definition at line 55 of file hardware_VideoOutSemiAuto.py.

```
56                                      :
57         query_cmd = "%s -q | grep '%s[0-9] connected' -c" % \
58             (self.XRANDR_PATH, output)
59         xrandr_out = utils.system_output(site_ui.xcommand(query_cmd),
60                                          ignore_status=True)
61         return int(xrandr_out) > 0
62
63
64     # Returns if given |output| port has a device that has been configured
    # otherwise raises TestFail
```

6.30.2.3 def hardware_VideoOutSemiAuto.hardware_VideoOutSemiAuto.__output_is_set (*self*, *output*) [private]

Definition at line 65 of file hardware_VideoOutSemiAuto.py.

```
66                                      :
67         query_cmd = "%s -q | grep '%s[0-9] connected' -n" % \
68             (self.XRANDR_PATH, output)
69         start_line = int(
70             utils.system_output(site_ui.xcommand(query_cmd)).split(':')[0]
71         )
72
73         # Gets 100 lines (to be safe) after context to get output after
74         query_cmd = \
75             "%s -q | grep '%s[0-9] connected' -n -A 100 | grep connected" % \
76                 (self.XRANDR_PATH, output)
77
78         try:
79             end_line = int(utils.system_output(
80                 site_ui.xcommand(query_cmd)).split('\n')[1].split('-')[0])
81         except:
82             logging.info("End line not found, assuming last output")
83             end_line = -1
84
85         if end_line != -1:
86             lines_between = end_line - start_line - 1
87         else:
88             line_between = 100
89         query_cmd = "%s -q | grep '%s[0-9] connected' -A %d | grep \\*" % \
90                 (self.XRANDR_PATH, output, lines_between)
91         try:
92             utils.system(site_ui.xcommand(query_cmd))
93         except:
94             raise error.TestFail("%s not set with monitor_reconfigure" % output)
95
96
97     # Configures |output| and returns if |output| has been configured.
    # Also will return false immediately if no device detected on the port
```

6.30.2.4 def hardware_VideoOutSemiAuto.hardware_VideoOutSemiAuto.__query_for_output (*self*, *output*) [private]

Definition at line 47 of file hardware_VideoOutSemiAuto.py.

```
48                                          :
49         query_cmd = "%s -q | grep %s -c" % (self.XRANDR_PATH, output)
50         xrandr_out = utils.system_output(site_ui.xcommand(query_cmd),
51                                          ignore_status=True)
52         return int(xrandr_out) > 0
53
54
    # Returns True if given |output| port has a connected device.
```

6.30.2.5 def hardware_VideoOutSemiAuto.hardware_VideoOutSemiAuto.run_once (*self*)

Definition at line 112 of file hardware_VideoOutSemiAuto.py.

```
113                     :
114         # Sanity check for xrandr application.
115         if not os.path.isfile(self.XRANDR_PATH):
116             raise error.TestFail("""
117                 XRandr missing from device cannot complete test
118             """)
119
120         # Determine if devices of interest are on system.
121         hdmi_exists = self.__query_for_output(self.HDMI_ID)
122         vga_exists = self.__query_for_output(self.VGA_ID)
123
124         # Raises NAError since these are optional devices.
125         if (not hdmi_exists) and (not vga_exists):
126             raise error.TestFail("Neither VGA or HDMI ports detected")
127
128         # Sanity check to make sure we can configure the devices.
129         if not os.path.isfile(self.RECONFIG_PATH):
130             raise error.TestFail("""
131                 Device detected but missing monitor_reconfigure tool
132             """)
133
134         # If either device is connected and able to be configured
135         # the test is successful.
136         success = False
137
138         # If devices exist, we should be able to configure and enable them
139         if hdmi_exists:
140             success |= self.__configure_and_check_output(self.HDMI_ID)
141         if vga_exists:
142             success |= self.__configure_and_check_output(self.VGA_ID)
143
144         if not success:
145             raise error.TestFail("""
146                 HDMI port or VGA port detected but no actual device connected.
147           """)
```

6.30.3 Member Data Documentation

6.30.3.1 string hardware_VideoOutSemiAuto.hardware_VideoOutSemiAuto.HDMI_ID = "HDMI" [static]

Definition at line 42 of file hardware_VideoOutSemiAuto.py.

6.30.3.2 string hardware_VideoOutSemiAuto.hardware_VideoOutSemiAuto.RECONFIG_PATH = "/usr/sbin/monitor_reconfigure" [static]

Definition at line 41 of file hardware_VideoOutSemiAuto.py.

6.30.3.3 int hardware_VideoOutSemiAuto.hardware_VideoOutSemiAuto.version = 1 [static]

Definition at line 39 of file hardware_VideoOutSemiAuto.py.

6.30.3.4 string hardware_VideoOutSemiAuto.hardware_VideoOutSemiAuto.VGA_ID = "VGA" [static]

Definition at line 43 of file hardware_VideoOutSemiAuto.py.

6.30.3.5 string hardware_VideoOutSemiAuto.hardware_VideoOutSemiAuto.XRANDR_PATH = "/usr/bin/xrandr" [static]

Definition at line 40 of file hardware_VideoOutSemiAuto.py.

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

- testsource/hardware_VideoOutSemiAuto.py

6.31 ltp.ltp Class Reference

Public Member Functions

- def initialize
- def setup
- def run_once

Public Attributes

- site_ignore_tests

Static Public Attributes

- int version = 6

Private Member Functions

- def _import_site_config

6.31.1 Detailed Description

Definition at line 29 of file ltp.py.

6.31.2 Member Function Documentation

6.31.2.1 def ltp.ltp._import_site_config (*self*) `[private]`

Definition at line 32 of file ltp.py.

```
33                                  :
34         site_config_path = os.path.join(os.path.dirname(__file__),
35                                         'site_config.py')
36         if os.path.exists(site_config_path):
37             # for some reason __import__ with full path does not work within
38             # autotest, although it works just fine on the same client machine
39             # in the python interactive shell or separate testcases
40             execfile(site_config_path)
41             self.site_ignore_tests = locals().get('ignore_tests', [])
42         else:
43             self.site_ignore_tests = []
44
```

6.31.2.2 def ltp.ltp.initialize (*self*)

Definition at line 45 of file ltp.py.

```
46                          :
47          self._import_site_config()
48          self.job.require_gcc()
49
50
    # http://prdownloads.sourceforge.net/ltp/ltp-full-20091231.tgz
```

6.31.2.3 def ltp.ltp.run_once (*self*, *args* = '', *script* = 'runltp', *ignore_tests* = [])

Definition at line 76 of file ltp.py.

```
77                                                                          :
78
79          ignore_tests = ignore_tests + self.site_ignore_tests
80
81          # In case the user wants to run another test script
82          if script == 'runltp':
83              logfile = os.path.join(self.resultsdir, 'ltp.log')
84              outfile = os.path.join(self.resultsdir, 'ltp.out')
85              failcmdfile = os.path.join(self.debugdir, 'failcmdfile')
86              excludecmdfile = os.path.join(self.bindir, 'site_excluded')
87              args2 = '-q -l %s -C %s -d %s -o %s -S %s' % (logfile, failcmdfile,
88                                                            self.tmpdir, outfile,
89                                                            excludecmdfile)
90              args = args + ' ' + args2
91
92          ltpbin_dir = os.path.join(self.srcdir, 'bin')
93          cmd = os.path.join(ltpbin_dir, script) + ' ' + args
94          result = utils.run(cmd, ignore_status=True)
95
96          # look for if there is any failed test command.
97          failed_cmd = open(failcmdfile).read()
98          if failed_cmd:
99              raise error.TestFail(failed_cmd)
```

6.31.2.4 def ltp.ltp.setup (*self*, *tarball* = 'ltp-full-20091231.tar.bz2')

Definition at line 51 of file ltp.py.

```
52                                                          :
53          tarball = utils.unmap_url(self.bindir, tarball, self.tmpdir)
54          utils.extract_tarball_to_dir(tarball, self.srcdir)
55          os.chdir(self.srcdir)
56          ltpbin_dir = os.path.join(self.srcdir, 'bin')
57          os.mkdir(ltpbin_dir)
58
59          utils.system('patch -p1 < ../ltp.patch')
60
61          # comment the capability tests if we fail to load the capability module
62          try:
63              utils.system('modprobe capability')
64          except error.CmdError, detail:
65              utils.system('patch -p1 < ../ltp_capability.patch')
66
67          utils.system('cp ../scan.c pan/')   # saves having lex installed
68          utils.system('make autotools')
69          utils.configure('--prefix=%s' % ltpbin_dir)
70          utils.system('make -j %d all' % utils.count_cpus())
71          utils.system('yes n | make SKIP_IDCHECK=1 install')
72
73
```

```
74      # Note: to run a specific test, try '-f cmdfile -s test' in the
75      # in the args (-f for test file and -s for the test case)
    # eg, job.run_test('ltp', '-f math -s float_bessel')
```

6.31.3 Member Data Documentation

6.31.3.1 ltp.ltp.site_ignore_tests

Definition at line 40 of file ltp.py.

6.31.3.2 int ltp.ltp.version = 6 `[static]`

Definition at line 30 of file ltp.py.

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

- testsource/ltp.py

6.32 network_DisableInterface.network_DisableInterface Class Reference

Public Member Functions

- def run_once
- def is_iface_up

Static Public Attributes

- int version = 1

Private Attributes

- _ifconfig

6.32.1 Detailed Description

Definition at line 34 of file network_DisableInterface.py.

6.32.2 Member Function Documentation

6.32.2.1 def network_DisableInterface.network_DisableInterface.is_iface_up (*self*, *name*)

Definition at line 67 of file network_DisableInterface.py.

```
68                                :
69         try:
70             out = utils.system_output('%s %s' % (self._ifconfig, name))
71         except error.CmdError, e:
72             logging.info(e)
73             raise error.TestNAError('test interface not found')
74 
75         match = re.search('UP', out, re.S)
76         return match
```

6.32.2.2 def network_DisableInterface.network_DisableInterface.run_once (*self*, *iface_name* = 'wlan0')

Definition at line 37 of file network_DisableInterface.py.

```
38                                        :
39         forced_up = False
40 
41         # use the right interface configuration utility
42         self._ifconfig = 'ifconfig'
43         if iface_name.startswith('hci'):
44             self._ifconfig = 'hciconfig'
45 
46         # bring up the interface if its not already up
47         if not self.is_iface_up(iface_name):
48             utils.system('%s %s up' % (self._ifconfig, iface_name))
```

```
49              if not self.is_iface_up(iface_name):
50                  raise error.TestFail('interface failed to come up')
51              forced_up = True
52
53          # bring interface down
54          utils.system('%s %s down' % (self._ifconfig, iface_name))
55          if self.is_iface_up(iface_name):
56              raise error.TestFail('interface failed to go down')
57
58          # if initial interface state was down, don't bring it back up
59          if forced_up:
60              return
61
62          # bring interface back up
63          utils.system('%s %s up' % (self._ifconfig, iface_name))
64          if not self.is_iface_up(iface_name):
65              raise error.TestFail('interface failed to come up')
66
```

6.32.3 Member Data Documentation

6.32.3.1 network_DisableInterface.network_DisableInterface._ifconfig [private]

Definition at line 41 of file network_DisableInterface.py.

6.32.3.2 int network_DisableInterface.network_DisableInterface.version = 1 [static]

Definition at line 35 of file network_DisableInterface.py.

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

- testsource/network_DisableInterface.py

6.33 network_WiFiCaps.network_WiFiCaps Class Reference

Public Member Functions

- def setup
- def run_once

Static Public Attributes

- int version = 1

Private Member Functions

- def __parse_iwcap
- def __run_iwcap

6.33.1 Detailed Description

Definition at line 39 of file network_WiFiCaps.py.

6.33.2 Member Function Documentation

6.33.2.1 def network_WiFiCaps.network_WiFiCaps.__parse_iwcap (*self*, *lines*) [private]

Parse the iwcap output

Definition at line 49 of file network_WiFiCaps.py.

```
50                                  :
51         """Parse the iwcap output"""
52
53         results = {}
54         parse_re = re.compile(r'([a-z0-9]*):[ ]*(.*)')
55         for line in lines.split('\n'):
56             line = line.rstrip()
57             logging.info('==> %s' %line)
58             match = parse_re.search(line)
59             if match:
60                 results[match.group(1)] = match.group(2)
61                 continue
62         return results
63
```

6.33.2.2 def network_WiFiCaps.network_WiFiCaps.__run_iwcap (*self*, *phy*, *caps*) [private]

Definition at line 64 of file network_WiFiCaps.py.

```
65                                      :
66         dir = os.path.join(self.autodir, 'deps', 'iwcap', 'iwcap')
67         iwcap = utils.run(dir + ' ' + phy + ' ' + string.join(caps))
68         return self.__parse_iwcap(iwcap.stdout)
69
```

6.33.2.3 def network_WiFiCaps.network_WiFiCaps.run_once (*self*)

Definition at line 70 of file network_WiFiCaps.py.

```
71                      :
72         phy = 'phy0'
73         requiredCaps = {
74             'sta'    : 'true',        # station mode
75
76             '24ghz'  : 'true',        # 2.4GHz band
77             '11b'    : 'true',
78             '11g'    : 'true',
79
80             '5ghz'   : 'true',        # 5GHz band
81             '11a'    : 'true',
82
83             '11n'    : 'true',        # 802.11n (both bands)
84             'ht40'   : 'true',        # HT40
85             'sgi40'  : 'true',        # Short GI in HT40
86         }
87
88         dep = 'iwcap'
89         dep_dir = os.path.join(self.autodir, 'deps', dep)
90         self.job.install_pkg(dep, 'dep', dep_dir)
91
92         results = self.__run_iwcap(phy, requiredCaps.keys())
93         for cap in requiredCaps:
94             if not cap in results:
95                 raise error.TestFail('Internal error, ' +
96                     'capability "%s" not handled' % cap)
97             if results[cap] != requiredCaps[cap]:
98                 raise error.TestFail('Requirement not met: ' +
99                     'cap "%s" is "%s" but expected "%s"'
100                     % (cap, results[cap], requiredCaps[cap]))
```

6.33.2.4 def network_WiFiCaps.network_WiFiCaps.setup (*self*)

Definition at line 42 of file network_WiFiCaps.py.

```
43                 :
44         self.job.setup_dep(['iwcap'])
45         # create a empty srcdir to prevent the error that checks .version
46         if not os.path.exists(self.srcdir):
47             os.mkdir(self.srcdir)
48
```

6.33.3 Member Data Documentation

6.33.3.1 int network_WiFiCaps.network_WiFiCaps.version = 1 `[static]`

Definition at line 40 of file network_WiFiCaps.py.

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

- testsource/network_WiFiCaps.py

6.34 platform_AesThroughput.platform_AesThroughput Class Reference

Public Member Functions

- def setup
- def run_once
- def openssl_speed
- def parse_results
- def update_stats
- def export_stats

Public Attributes

- results

Static Public Attributes

- int version = 1

6.34.1 Detailed Description

Definition at line 36 of file platform_AesThroughput.py.

6.34.2 Member Function Documentation

6.34.2.1 def platform_AesThroughput.platform_AesThroughput.export_stats (*self*)

Definition at line 84 of file platform_AesThroughput.py.

```
85                              :
86         self.write_perf_keyval(self.results)
```

6.34.2.2 def platform_AesThroughput.platform_AesThroughput.openssl_speed (*self*, *cipher*, *options* = '')

Definition at line 53 of file platform_AesThroughput.py.

```
54                                                  :
55         cmd = 'openssl speed %s -mr %s' % (cipher, options)
56         return utils.system_output(cmd, retain_output=True)
57
```

6.34.2.3 def platform_AesThroughput.platform_AesThroughput.parse_results (*self*, *results*, *name = ''*)

Definition at line 58 of file platform_AesThroughput.py.

```
59                                              :
60         # Split the results into lines.
61         # We really only want the final line for our purposes.
62         type, times = results.split("\n")[-1].split(' ')
63         # +F:num:aes-256 cbc -> aes_256_cbc
64         type = re.sub('[- ]', '_', type.split(':')[-1])
65         # cbc:time:time:time:... -> time, time, ...
66         times = times.split(':')[1:]
67
68         # Build the key names
69         if len(name) > 0:
70           name = name + '_'
71         key_prefix = 'bytes_per_sec_' + name + type + '_blocksz_'
72         keys = ['16_bytes', '64_bytes', '256_bytes', '1024_bytes', '8192_bytes']
73         keys = [key_prefix+k for k in keys]
74
75         if len(times) > len(keys):
76             logging.debug(results)
77             raise error.TestFail('openssl output format parsing failed')
78         return dict(zip(keys, times))
79
```

6.34.2.4 def platform_AesThroughput.platform_AesThroughput.run_once (*self*)

Definition at line 44 of file platform_AesThroughput.py.

```
45                       :
46         num_cpus = bin_utils.count_cpus()
47         logging.debug('Running using all cpus: %d' % num_cpus)
48         results = self.openssl_speed('aes-256-cbc', '-multi %d' % num_cpus)
49         parsed = self.parse_results(results)
50         self.update_stats(parsed)
51         self.export_stats()
52
```

6.34.2.5 def platform_AesThroughput.platform_AesThroughput.setup (*self*)

Definition at line 40 of file platform_AesThroughput.py.

```
41                     :
42         self.results = {'bytes_per_sec_ideal_min' : 20 * 1024 * 1024}
43
```

6.34.2.6 def platform_AesThroughput.platform_AesThroughput.update_stats (*self*, *keyvals*)

Definition at line 80 of file platform_AesThroughput.py.

```
81                                    :
82         self.results.update(keyvals)
83
```

6.34.3 Member Data Documentation

6.34.3.1 platform_AesThroughput.platform_AesThroughput.results

Definition at line 41 of file platform_AesThroughput.py.

6.34.3.2 int platform_AesThroughput.platform_AesThroughput.version = 1 [static]

Definition at line 37 of file platform_AesThroughput.py.

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

- testsource/platform_AesThroughput.py

6.35 platform_BootPerf.platform_BootPerf Class Reference

Public Member Functions

- def run_once

Static Public Attributes

- int version = 1

Private Member Functions

- def __parse_uptime
- def __parse_disk_login_prompt_ready
- def __parse_firmware_boot_time
- def __parse_syslog

6.35.1 Detailed Description

Definition at line 40 of file platform_BootPerf.py.

6.35.2 Member Function Documentation

6.35.2.1 def platform_BootPerf.platform_BootPerf.__parse_disk_login_prompt_ready (*self*, *results*) [private]

Definition at line 55 of file platform_BootPerf.py.

```
56                                                      :
57         filename = '/tmp/disk-login-prompt-ready'
58         vals = []
59         try:
60             data = file(filename).read()
61             vals = re.split(r' +', data.strip())
62         except IOError:
63             raise error.TestFail('Test is unable to read "%s"' % filename)
64         results['sectors_read_kernel_to_login'] = float(vals[2])
65
```

6.35.2.2 def platform_BootPerf.platform_BootPerf.__parse_firmware_boot_time (*self*, *results*) [private]

Definition at line 66 of file platform_BootPerf.py.

```
67                                                  :
68         data = None
69         try:
70             # If the firmware boot time is not available, the file
71             # will not exists.
72             data = utils.read_one_line('/tmp/firmware-boot-time')
```

```
73          except IOError:
74              return
75          results['seconds_firmware_boot'] = float(data)
76
77
    # Find the reboot/shutdown time if the last boot was a reboot.
```

6.35.2.3 def platform_BootPerf.platform_BootPerf.__parse_syslog (*self*, *results*, *last_boot_was_reboot*) [private]

Definition at line 78 of file platform_BootPerf.py.

```
79                                                              :
80          file_handle = None
81          logfile = '/var/log/messages'
82          try:
83              file_handle = open(logfile, 'r')
84          except:
85              raise error.TestFail('Test is unable to read "%s"' % logfile)
86          mhz = 0
87          startups_found = 0
88          last_shutdown_time = None
89          kernel_start_time = None
90          datetime_re = r'(\d{4})-(\d{2})-(\d{2})[A-Z]' + \
91                        r'(\d{2}):(\d{2}):(\d{2})\.(\d{6})'
92          last_shutdown_re = re.compile(
93              datetime_re + r'.*(klog|tty2) main process.*killed by TERM')
94          startup_re = re.compile(datetime_re + r'.*000\] Linux version \d')
95          mhz_re = re.compile(r'Detected (\d+\.\d+) MHz processor.')
96          for line in file_handle.readlines():
97              match = startup_re.match(line)
98              if match is not None:
99                  mhz = 0
100                 datetime_args = tuple([int(x) for x in match.groups()[:7]])
101                 kernel_start_time = datetime.datetime(*datetime_args)
102                 startups_found += 1
103             match = last_shutdown_re.match(line)
104             if match is not None:
105                 datetime_args = tuple([int(x) for x in match.groups()[:7]])
106                 last_shutdown_time = datetime.datetime(*datetime_args)
107             match = mhz_re.search(line)
108             if match is not None:
109                 mhz = float(match.group(1))
110         file_handle.close()
111         if (last_shutdown_time != None and last_boot_was_reboot and
112             kernel_start_time != None):
113             logging.info('Kernel start time: %s, last shutdown time: %s' %
114                          (kernel_start_time, last_shutdown_time))
115             delta = kernel_start_time - last_shutdown_time
116             # There is no guarantee that we will have gotten a shutdown
117             # log message/time.  It's possible to not get any kill messages
118             # logged to syslog before rsyslogd itself is killed.  If
119             # that occurs, this reboot time will be completely wrong.
120             reboot_time = (float(delta.days) * 86400.0 +
121                            float(delta.seconds) +
122                            float(delta.microseconds) /
123                            1000000.0)
124             results['seconds_reboot_time'] = reboot_time
125             results['seconds_shutdown_time'] = \
126                 reboot_time - results.get('seconds_firmware_boot', 0.0)
127         results['reboots_in_syslog'] = startups_found
128         results['mhz_primary_cpu'] = mhz
129
```

6.35.2.4 def platform_BootPerf.platform_BootPerf.__parse_uptime (*self*, *filename*) [private]

Definition at line 44 of file platform_BootPerf.py.

```
45                                      :
46         vals = []
47         try:
48             data = file(filename).read()
49             vals = re.split(r' +', data.strip())
50         except IOError:
51             raise error.TestFail('Test is unable to read uptime file "%s"' %
52                                  filename)
53         return float(vals[0])
54
```

6.35.2.5 def platform_BootPerf.platform_BootPerf.run_once (*self*, *last_boot_was_reboot* = False)

Definition at line 130 of file platform_BootPerf.py.

```
131                                                 :
132         # Parse key metric files and generate key/value pairs
133         results = {}
134
135         uptime_files = [
136             ('seconds_kernel_to_startup', '/tmp/uptime-pre-startup'),
137             ('seconds_kernel_to_startup_done', '/tmp/uptime-post-startup'),
138             ('seconds_kernel_to_login', '/tmp/uptime-login-prompt-ready')]
139
140         for resultname, filename in uptime_files:
141             results[resultname] = self.__parse_uptime(filename)
142
143         self.__parse_firmware_boot_time(results)
144         self.__parse_disk_login_prompt_ready(results)
145         self.__parse_syslog(results, last_boot_was_reboot)
146
147         if ('seconds_firmware_boot' in results and
148             'seconds_kernel_to_login' in results):
149             results['seconds_power_on_to_login'] = \
150                 results['seconds_firmware_boot'] + \
151                 results['seconds_kernel_to_login']
152
153         self.write_perf_keyval(results)
```

6.35.3 Member Data Documentation

6.35.3.1 int platform_BootPerf.platform_BootPerf.version = 1 [static]

Definition at line 41 of file platform_BootPerf.py.

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

- testsource/platform_BootPerf.py

6.36 platform_KernelVersion.platform_KernelVersion Class Reference

Public Member Functions

- def run_once

Static Public Attributes

- int version = 1

6.36.1 Detailed Description

Definition at line 30 of file platform_KernelVersion.py.

6.36.2 Member Function Documentation

6.36.2.1 def platform_KernelVersion.platform_KernelVersion.run_once (*self*, *kernel_version* = '2.6.31')

Definition at line 33 of file platform_KernelVersion.py.

```
34                                              :
35         try:
36             utils.check_kernel_ver(kernel_version)
37         except error.TestError, e:
38             logging.debug(e)
39             raise error.TestFail(e)
```

6.36.3 Member Data Documentation

6.36.3.1 int platform_KernelVersion.platform_KernelVersion.version = 1 [static]

Definition at line 31 of file platform_KernelVersion.py.

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

- testsource/platform_KernelVersion.py

6.37 power_BatteryCharge.power_BatteryCharge Class Reference

Public Member Functions

- def initialize
- def run_once
- def postprocess_iteration
- def on_ac

Public Attributes

- status
- remaining_time
- max_run_time
- charge_full_design
- initial_charge

Static Public Attributes

- int version = 1

6.37.1 Detailed Description

Definition at line 36 of file power_BatteryCharge.py.

6.37.2 Member Function Documentation

6.37.2.1 def power_BatteryCharge.power_BatteryCharge.initialize (*self*)

Definition at line 39 of file power_BatteryCharge.py.

```
40                         :
41         self.status = site_power_status.get_status()
42
43         if not self.on_ac():
44             raise error.TestNAError(
45                   'This test needs to be run with the AC power online')
46
```

6.37.2.2 def power_BatteryCharge.power_BatteryCharge.on_ac (*self*)

Definition at line 121 of file power_BatteryCharge.py.

```
122                     :
123         return self.status.linepower[0].online
```

6.37.2.3 def power_BatteryCharge.power_BatteryCharge.postprocess_iteration (*self*)

Definition at line 106 of file power_BatteryCharge.py.

```
107                                     :
108         keyvals = {}
109         keyvals['ah_charge_full'] = self.status.battery[0].charge_full
110         keyvals['ah_charge_full_design'] = self.charge_full_design
111         keyvals['ah_initial_charge'] = self.initial_charge
112         keyvals['ah_final_charge'] = self.status.battery[0].charge_now
113         keyvals['s_time_taken'] = self.max_run_time - self.remaining_time
114         keyvals['percent_initial_charge'] = self.initial_charge * 100 / \
115                                             keyvals['ah_charge_full_design']
116         keyvals['percent_final_charge'] = keyvals['ah_final_charge'] * 100 / \
117                                           keyvals['ah_charge_full_design']
118
119         self.write_perf_keyval(keyvals)
120
```

6.37.2.4 def power_BatteryCharge.power_BatteryCharge.run_once (*self*, *max_run_time* = `180`, *percent_charge_to_add* = `1`, *percent_initial_charge_max* = `None`, *percent_target_charge* = `None`)

```
max_run_time: maximum time the test will run for
percent_charge_to_add: percentage of the design capacity charge to
  add. The target charge will be capped at the design capacity.
percent_initial_charge_max: maxium allowed initial charge.
```

Definition at line 47 of file power_BatteryCharge.py.

```
50                                          :
51         """
52         max_run_time: maximum time the test will run for
53         percent_charge_to_add: percentage of the design capacity charge to
54                   add. The target charge will be capped at the design capacity.
55         percent_initial_charge_max: maxium allowed initial charge.
56         """
57
58         time_to_sleep = 60
59         self.remaining_time = self.max_run_time = max_run_time
60
61         self.charge_full_design = self.status.battery[0].charge_full_design
62         self.initial_charge = self.status.battery[0].charge_now
63         percent_initial_charge = self.initial_charge * 100 / \
64                                  self.charge_full_design
65         if percent_initial_charge_max and percent_initial_charge > \
66                                           percent_initial_charge_max:
67             raise error.TestError('Initial charge (%f) higher than max (%f)'
68                       % (percent_initial_charge, percent_initial_charge_max))
69
70         current_charge = self.initial_charge
71         if percent_target_charge is None:
72             charge_to_add = self.charge_full_design * \
73                             float(percent_charge_to_add) / 100
74             target_charge = current_charge + charge_to_add
75         else:
76             target_charge = self.charge_full_design * \
77                             float(percent_target_charge) / 100
78
79         # trim target_charge if it exceeds designed capacity
80         if target_charge > self.charge_full_design:
```

```
81              target_charge = self.charge_full_design
82
83          logging.info('max_run_time: %d' % self.max_run_time)
84          logging.info('initial_charge: %f' % self.initial_charge)
85          logging.info('target_charge: %f' % target_charge)
86
87          while self.remaining_time and current_charge < target_charge:
88              if time_to_sleep > self.remaining_time:
89                  time_to_sleep = self.remaining_time
90              self.remaining_time -= time_to_sleep
91
92              time.sleep(time_to_sleep)
93
94              self.status.refresh()
95              if not self.on_ac():
96                  raise error.TestError(
97                        'This test needs to be run with the AC power online')
98
99              new_charge = self.status.battery[0].charge_now
100              logging.info('time_to_sleep: %d' % time_to_sleep)
101              logging.info('charge_added: %f' % (new_charge - current_charge))
102
103              current_charge = new_charge
104              logging.info('current_charge: %f' % current_charge)
105
```

6.37.3 Member Data Documentation

6.37.3.1 power_BatteryCharge.power_BatteryCharge.charge_full_design

Definition at line 58 of file power_BatteryCharge.py.

6.37.3.2 power_BatteryCharge.power_BatteryCharge.initial_charge

Definition at line 59 of file power_BatteryCharge.py.

6.37.3.3 power_BatteryCharge.power_BatteryCharge.max_run_time

Definition at line 56 of file power_BatteryCharge.py.

6.37.3.4 power_BatteryCharge.power_BatteryCharge.remaining_time

Definition at line 56 of file power_BatteryCharge.py.

6.37.3.5 power_BatteryCharge.power_BatteryCharge.status

Definition at line 40 of file power_BatteryCharge.py.

6.37.3.6 int power_BatteryCharge.power_BatteryCharge.version = 1 [static]

Definition at line 37 of file power_BatteryCharge.py.

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

- testsource/power_BatteryCharge.py

6.38 power_CPUFreq.power_CPUFreq Class Reference

Public Member Functions

- def run_once

Static Public Attributes

- int version = 1

6.38.1 Detailed Description

Definition at line 35 of file power_CPUFreq.py.

6.38.2 Member Function Documentation

6.38.2.1 def power_CPUFreq.power_CPUFreq.run_once (*self*)

Definition at line 38 of file power_CPUFreq.py.

```
39                      :
40         cpufreq_path = '/sys/devices/system/cpu/cpu*/cpufreq'
41
42         dirs  = glob.glob(cpufreq_path)
43         if not dirs:
44             raise error.TestFail('cpufreq not supported')
45
46         for dir in dirs:
47             cpu = cpufreq(dir)
48
49             if 'userspace' not in cpu.get_available_governors():
50                 raise error.TestError('userspace governor not supported')
51
52             available_frequencies = cpu.get_available_frequencies()
53             if len(available_frequencies) == 1:
54                 raise error.TestFail('Not enough frequencies supported!')
55
56             # save cpufreq state so that it can be restored at the end
57             # of the test
58             cpu.save_state()
59
60             # set cpufreq governor to userspace
61             cpu.set_governor('userspace')
62
63             # cycle through all available frequencies
64             for freq in available_frequencies:
65                 cpu.set_frequency(freq)
66                 if freq != cpu.get_current_frequency():
67                     cpu.restore_state()
68                     raise error.TestFail('Unable to set frequency')
69
70             # restore cpufreq state
71             cpu.restore_state()
72
```

6.38.3 Member Data Documentation

6.38.3.1 int power_CPUFreq.power_CPUFreq.version = 1 [static]

Definition at line 36 of file power_CPUFreq.py.

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

- testsource/power_CPUFreq.py

6.39 power_CPUIdle.power_CPUIdle Class Reference

Public Member Functions

- def run_once

Static Public Attributes

- int version = 1

6.39.1 Detailed Description

Definition at line 31 of file power_CPUIdle.py.

6.39.2 Member Function Documentation

6.39.2.1 def power_CPUIdle.power_CPUIdle.run_once (*self*, *sleep_time* = 5)

Definition at line 34 of file power_CPUIdle.py.

```
35                                      :
36         all_cpus = cpus()
37
38         idle_time_at_start = all_cpus.idle_time()
39         logging.info('idle_time_at_start: %d' % idle_time_at_start)
40
41         # sleep for some time to allow the CPUs to drop into idle states
42         time.sleep(sleep_time)
43
44         idle_time_at_end = all_cpus.idle_time()
45         logging.info('idle_time_at_end: %d' % idle_time_at_end)
46
47         idle_time_delta_ms = (idle_time_at_end - idle_time_at_start) / 1000
48         logging.info('idle time delta (ms): %d' % idle_time_delta_ms)
49
50         if idle_time_at_end == idle_time_at_start:
51             raise error.TestFail('No Idle cycles')
52
53
```

6.39.3 Member Data Documentation

6.39.3.1 int power_CPUIdle.power_CPUIdle.version = 1 [static]

Definition at line 32 of file power_CPUIdle.py.

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

- testsource/power_CPUIdle.py

6.40 power_Draw.power_Draw Class Reference

Public Member Functions

- def run_once

Static Public Attributes

- int version = 1

6.40.1 Detailed Description

Definition at line 35 of file power_Draw.py.

6.40.2 Member Function Documentation

6.40.2.1 def power_Draw.power_Draw.run_once (*self*, *seconds* = 200)

Definition at line 39 of file power_Draw.py.

```
40                                      :
41         status = site_power_status.get_status()
42         if status.linepower[0].online:
43             logging.warn('AC power is online -- '
44                          'unable to monitor energy consumption')
45             return
46
47         start_energy = status.battery[0].energy
48
49         # Let the test run
50         time.sleep(seconds)
51
52         status.refresh()
53         end_energy = status.battery[0].energy
54
55         consumed_energy = start_energy - end_energy
56         energy_rate = consumed_energy * 60 * 60 / seconds
57
58         keyvals = {}
59         keyvals['wh_energy_full'] = status.battery[0].energy_full
60         keyvals['wh_start_energy'] = start_energy
61         keyvals['wh_end_energy'] = end_energy
62         keyvals['wh_consumed_energy'] = consumed_energy
63         keyvals['w_average_energy_rate'] = energy_rate
64         keyvals['w_end_energy_rate'] = status.battery[0].energy_rate
65         self.write_perf_keyval(keyvals)
```

6.40.3 Member Data Documentation

6.40.3.1 int power_Draw.power_Draw.version = 1 [static]

Definition at line 36 of file power_Draw.py.

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

- testsource/power_Draw.py

6.41 power_LoadTest.power_LoadTest Class Reference

Public Member Functions

- def setup
- def run_once
- def postprocess_iteration
- def cleanup

Static Public Attributes

- int version = 1

Private Member Functions

- def _is_network_iface_running
- def _percent_current_charge
- def _write_ext_params
- def _do_wait

Private Attributes

- _loop_time
- _loop_count
- _mseconds
- _verbose
- _low_battery_threshold
- _should_scroll
- _should_scroll_up
- _scroll_loop
- _scroll_interval_ms
- _scroll_by_pixels
- _tmp_keyvals
- _power_status
- _ext_path
- _testServer
- _usb_stats
- _cpufreq_stats
- _cpuidle_stats
- _ah_charge_start
- _wh_energy_start

6.41.1 Detailed Description

Definition at line 46 of file power_LoadTest.py.

6.41.2 Member Function Documentation

6.41.2.1 def power_LoadTest.power_LoadTest._do_wait (*self*, *verbose*, *seconds*, *latch*, *session*) [private]

Definition at line 263 of file power_LoadTest.py.

```
264                                                         :
265         latched = False
266         low_battery = False
267         total_time = seconds + 60
268         elapsed_time = 0
269         wait_time = 60
270
271         while elapsed_time < total_time:
272             time.sleep(wait_time)
273             elapsed_time += wait_time
274
275             self._power_status.refresh()
276             if verbose:
277                 logging.debug('ah_charge_now %f' \
278                     % self._power_status.battery[0].charge_now)
279                 logging.debug('w_energy_rate %f' \
280                     % self._power_status.battery[0].energy_rate)
281
282             low_battery = (self._percent_current_charge() <
283                            self._low_battery_threshold)
284
285             latched = latch.is_set()
286
287             if latched or low_battery:
288                 break
289
290         if latched:
291             # record chrome power extension stats
292             form_data = self._testServer.get_form_entries()
293             logging.debug(form_data)
294             for e in form_data:
295                 key = 'ext_' + e
296                 if key in self._tmp_keyvals:
297                     self._tmp_keyvals[key] += form_data[e]
298                 else:
299                     self._tmp_keyvals[key] = form_data[e]
300         else:
301             logging.debug("Didn't get status back from power extension")
302
303         return low_battery
```

6.41.2.2 def power_LoadTest.power_LoadTest._is_network_iface_running (*self*, *name*) [private]

Definition at line 234 of file power_LoadTest.py.

```
235                                             :
236         try:
237             out = utils.system_output('ifconfig %s' % name)
238         except error.CmdError, e:
239             logging.info(e)
240             return False
241
242         match = re.search('RUNNING', out, re.S)
243         return match
```

```
244
```

6.41.2.3 def power_LoadTest.power_LoadTest._percent_current_charge (*self*) [private]

Definition at line 245 of file power_LoadTest.py.

```
246                                     :
247         return self._power_status.battery[0].charge_now * 100 / \
248                self._power_status.battery[0].charge_full_design
249
```

6.41.2.4 def power_LoadTest.power_LoadTest._write_ext_params (*self*) [private]

Definition at line 250 of file power_LoadTest.py.

```
251                                 :
252         data = ''
253         template= 'var %s = %s;\n'
254         for k in params_dict:
255             data += template % (k, getattr(self, params_dict[k]))
256
257         filename = os.path.join(self._ext_path, 'params.js')
258         utils.open_write_close(filename, data)
259
260         logging.debug('filename ' + filename)
261         logging.debug(data)
262
```

6.41.2.5 def power_LoadTest.power_LoadTest.cleanup (*self*)

Definition at line 228 of file power_LoadTest.py.

```
229                     :
230         # re-enable screen locker and powerd. This also re-enables dpms.
231         os.system('start powerd')
232         os.system('start screen-locker')
233
```

6.41.2.6 def power_LoadTest.power_LoadTest.postprocess_iteration (*self*)

Definition at line 182 of file power_LoadTest.py.

```
183                                     :
184         keyvals = {}
185
186         # refresh power related statistics
187         usb_stats = self._usb_stats.refresh()
188         cpufreq_stats = self._cpufreq_stats.refresh()
189         cpuidle_stats = self._cpuidle_stats.refresh()
190
```

```
191         # record percent time USB devices were not in suspended state
192         keyvals['percent_usb_active'] = usb_stats
193
194         # record percent time spent in each CPU C-state
195         for state in cpuidle_stats:
196             keyvals['percent_cpuidle_%s_time' % state] = cpuidle_stats[state]
197
198         # record percent time spent at each CPU frequency
199         for freq in cpufreq_stats:
200             keyvals['percent_cpufreq_%s_time' % freq] = cpufreq_stats[freq]
201
202         # record battery stats
203         keyvals['a_current_now'] = self._power_status.battery[0].current_now
204         keyvals['ah_charge_full'] = self._power_status.battery[0].charge_full
205         keyvals['ah_charge_full_design'] = \
206                                 self._power_status.battery[0].charge_full_design
207         keyvals['ah_charge_start'] = self._ah_charge_start
208         keyvals['ah_charge_now'] = self._power_status.battery[0].charge_now
209         keyvals['ah_charge_used'] = keyvals['ah_charge_start'] - \
210                                     keyvals['ah_charge_now']
211         keyvals['wh_energy_start'] = self._wh_energy_start
212         keyvals['wh_energy_now'] = self._power_status.battery[0].energy
213         keyvals['wh_energy_used'] = keyvals['wh_energy_start'] - \
214                                     keyvals['wh_energy_now']
215         keyvals['v_voltage_min_design'] = \
216                                 self._power_status.battery[0].voltage_min_design
217         keyvals['v_voltage_now'] = self._power_status.battery[0].voltage_now
218
219         keyvals.update(self._tmp_keyvals)
220
221         keyvals['a_current_rate'] = keyvals['ah_charge_used'] * 60 / \
222                                     keyvals['minutes_battery_life']
223         keyvals['w_energy_rate'] = keyvals['wh_energy_used'] * 60 / \
224                                    keyvals['minutes_battery_life']
225
226         self.write_perf_keyval(keyvals)
227
```

6.41.2.7 def power_LoadTest.power_LoadTest.run_once (*self*, *percent_initial_charge_min* = `None`, *check_network* = `True`, *loop_time* = `3600`, *loop_count* = `1`, *should_scroll* = `'true'`, *should_scroll_up* = `'true'`, *scroll_loop* = `'false'`, *scroll_interval_ms* = `'10000'`, *scroll_by_pixels* = `'600'`, *low_battery_threshold* = `3`, *verbose* = `True`)

```
percent_initial_charge_min: min battery charge at start of test
check_network: check that Ethernet interface is not running
loop_count: number of times to loop the test for
loop_time: length of time to run the test for in each loop
should_scroll: should the extension scroll pages
should_scroll_up: should scroll in up direction
scroll_loop: continue scrolling indefinitely
scroll_interval_ms: how often to scoll
scroll_by_pixels: number of pixels to scroll each time
```

Definition at line 57 of file power_LoadTest.py.

```
63                              :
64         """
65         percent_initial_charge_min: min battery charge at start of test
66         check_network: check that Ethernet interface is not running
67         loop_count: number of times to loop the test for
68         loop_time: length of time to run the test for in each loop
69         should_scroll: should the extension scroll pages
```

```
70          should_scroll_up: should scroll in up direction
71          scroll_loop: continue scrolling indefinitely
72          scroll_interval_ms: how often to scoll
73          scroll_by_pixels: number of pixels to scroll each time
74          """
75
76          self._loop_time = loop_time
77          self._loop_count = loop_count
78          self._mseconds = self._loop_time * 1000
79          self._verbose = verbose
80          self._low_battery_threshold = low_battery_threshold
81          self._should_scroll = should_scroll
82          self._should_scroll_up = should_scroll_up
83          self._scroll_loop = scroll_loop
84          self._scroll_interval_ms = scroll_interval_ms
85          self._scroll_by_pixels = scroll_by_pixels
86          self._tmp_keyvals = {}
87
88          self._power_status = site_power_status.get_status()
89
90          # verify that initial conditions are met:
91          if self._power_status.linepower[0].online:
92              raise error.TestNAError(
93                  'Running on AC power. Please remove AC power cable')
94
95          percent_initial_charge = self._percent_current_charge()
96          if percent_initial_charge_min and percent_initial_charge < \
97                                           percent_initial_charge_min:
98              raise error.TestError('Initial charge (%f) less than min (%f)'
99                        % (percent_initial_charge, percent_initial_charge_min))
100
101          if check_network and self._is_network_iface_running('eth0'):
102              raise error.TestNAError(
103                  'Ethernet interface is active. Please remove Ethernet cable')
104
105          # TODO (snanda):
106          # - set brightness level
107          # - turn off suspend on idle (not implemented yet in Chrome OS)
108
109          # record the current and max backlight levels
110          cmd = 'backlight-tool --get_max_brightness'
111          self._tmp_keyvals['level_backlight_max'] = int(
112                                              utils.system_output(cmd).rstrip())
113
114          cmd = 'backlight-tool --get_brightness'
115          self._tmp_keyvals['level_backlight_current'] = int(
116                                              utils.system_output(cmd).rstrip())
117
118          # disable screen locker and powerd
119          os.system('stop screen-locker')
120          os.system('stop powerd')
121
122          # disable screen blanking. Stopping screen-locker isn't
123          # synchronous :(. Add a sleep for now, till powerd comes around
124          # and fixes all this for us.
125          time.sleep(5)
126          site_ui.xsystem(os.path.join(self.bindir, 'xset') + ' s off')
127          site_ui.xsystem(os.path.join(self.bindir, 'xset') + ' dpms 0 0 0')
128          site_ui.xsystem(os.path.join(self.bindir, 'xset') + ' -dpms')
129
130          # fix up file perms for the power test extension so that chrome
131          # can access it
132          os.system('chmod -R 755 %s' % self.bindir)
133
134          # write test parameters to the power extension's params.js file
135          self._ext_path = os.path.join(self.bindir, 'extension')
136          self._write_ext_params()
```

```
137
138         # setup a HTTP Server to listen for status updates from the power
139         # test extension
140         self._testServer = site_httpd.HTTPListener(8001, docroot=self.bindir)
141         self._testServer.run()
142
143         # initialize various interesting power related stats
144         self._usb_stats = site_power_status.USBSuspendStats()
145         self._cpufreq_stats = site_power_status.CPUFreqStats()
146         self._cpuidle_stats = site_power_status.CPUIdleStats()
147
148
149         self._usb_stats.refresh()
150         self._cpufreq_stats.refresh()
151         self._cpuidle_stats.refresh()
152         self._power_status.refresh()
153
154         self._ah_charge_start = self._power_status.battery[0].charge_now
155         self._wh_energy_start = self._power_status.battery[0].energy
156
157         t0 = time.time()
158
159         for i in range(self._loop_count):
160             # the power test extension will report its status here
161             latch = self._testServer.add_wait_url('/status')
162
163             # launch chrome with power test extension
164             args = '--load-extension=%s' % self._ext_path
165             session = site_ui.ChromeSession(args, clean_state=False)
166
167             low_battery = self._do_wait(self._verbose, self._loop_time,
168                                         latch, session)
169             session.close()
170
171             if self._verbose:
172                 logging.debug('loop %d completed' % i)
173                 logging.debug(utils.system_output('xset q'))
174
175             if low_battery:
176                 logging.info('Exiting due to low battery')
177                 break
178
179         t1 = time.time()
180         self._tmp_keyvals['minutes_battery_life'] = (t1 - t0) / 60
181
```

6.41.2.8 def power_LoadTest.power_LoadTest.setup (*self*)

Definition at line 49 of file power_LoadTest.py.

```
50                 :
51         # TODO(snanda): Remove once power manager is in
52         shutil.copy(os.path.join(os.environ['SYSROOT'], 'usr/bin/xset'),
53                                  self.bindir)
54         if not os.path.exists(self.srcdir):
55             os.mkdir(self.srcdir)
56
```

6.41.3 Member Data Documentation

6.41.3.1 power_LoadTest.power_LoadTest._ah_charge_start [private]

Definition at line 148 of file power_LoadTest.py.

6.41.3.2 power_LoadTest.power_LoadTest._cpufreq_stats [private]

Definition at line 139 of file power_LoadTest.py.

6.41.3.3 power_LoadTest.power_LoadTest._cpuidle_stats [private]

Definition at line 140 of file power_LoadTest.py.

6.41.3.4 power_LoadTest.power_LoadTest._ext_path [private]

Definition at line 129 of file power_LoadTest.py.

6.41.3.5 power_LoadTest.power_LoadTest._loop_count [private]

Definition at line 71 of file power_LoadTest.py.

6.41.3.6 power_LoadTest.power_LoadTest._loop_time [private]

Definition at line 70 of file power_LoadTest.py.

6.41.3.7 power_LoadTest.power_LoadTest._low_battery_threshold [private]

Definition at line 74 of file power_LoadTest.py.

6.41.3.8 power_LoadTest.power_LoadTest._mseconds [private]

Definition at line 72 of file power_LoadTest.py.

6.41.3.9 power_LoadTest.power_LoadTest._power_status [private]

Definition at line 82 of file power_LoadTest.py.

6.41.3.10 power_LoadTest.power_LoadTest._scroll_by_pixels [private]

Definition at line 79 of file power_LoadTest.py.

6.41.3.11 power_LoadTest.power_LoadTest._scroll_interval_ms [private]

Definition at line 78 of file power_LoadTest.py.

6.41.3.12 power_LoadTest.power_LoadTest._scroll_loop [private]

Definition at line 77 of file power_LoadTest.py.

6.41.3.13 power_LoadTest.power_LoadTest._should_scroll [private]

Definition at line 75 of file power_LoadTest.py.

6.41.3.14 power_LoadTest.power_LoadTest._should_scroll_up [private]

Definition at line 76 of file power_LoadTest.py.

6.41.3.15 power_LoadTest.power_LoadTest._testServer [private]

Definition at line 134 of file power_LoadTest.py.

6.41.3.16 power_LoadTest.power_LoadTest._tmp_keyvals [private]

Definition at line 80 of file power_LoadTest.py.

6.41.3.17 power_LoadTest.power_LoadTest._usb_stats [private]

Definition at line 138 of file power_LoadTest.py.

6.41.3.18 power_LoadTest.power_LoadTest._verbose [private]

Definition at line 73 of file power_LoadTest.py.

6.41.3.19 power_LoadTest.power_LoadTest._wh_energy_start [private]

Definition at line 149 of file power_LoadTest.py.

6.41.3.20 int power_LoadTest.power_LoadTest.version = 1 [static]

Definition at line 47 of file power_LoadTest.py.

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

- testsource/power_LoadTest.py

6.42 power_Resume.power_Resume Class Reference

Public Member Functions

- def run_once

Static Public Attributes

- int version = 1
- preserve_srcdir = True

Private Member Functions

- def __get_last_msg_time
- def __get_start_suspend_time
- def __get_end_suspend_time
- def __is_iface_up
- def __sanity_check_system

6.42.1 Detailed Description

Definition at line 35 of file power_Resume.py.

6.42.2 Member Function Documentation

6.42.2.1 def power_Resume.power_Resume.__get_end_suspend_time (*self*) [private]

Definition at line 56 of file power_Resume.py.

```
57                                      :
58         return self.__get_last_msg_time('CPU [0-9] is now offline')
59
```

6.42.2.2 def power_Resume.power_Resume.__get_last_msg_time (*self*, *msg*) [private]

Definition at line 40 of file power_Resume.py.

```
41                                        :
42         data = commands.getoutput(
43             "grep '%s' /var/log/messages | tail -n 1" % msg)
44         match = re.search(r' \[\s*([0-9.]+)\] ', data)
45         if match is None:
46             raise error.TestError('Failed to find log message: ' + msg)
47
48         msg_time = float(match.group(1))
49         logging.debug("Last message '%s' time = %f" % (msg, msg_time))
50         return msg_time
51
```

6.42.2.3 def power_Resume.power_Resume.__get_start_suspend_time (*self*) [private]

Definition at line 52 of file power_Resume.py.

```
53                                      :
54         return self.__get_last_msg_time('Freezing user space')
55
```

6.42.2.4 def power_Resume.power_Resume.__is_iface_up (*self*, *name*) [private]

Definition at line 60 of file power_Resume.py.

```
61                                  :
62         try:
63             out = utils.system_output('/sbin/ifconfig %s' % name)
64         except error.CmdError, e:
65             logging.info(e)
66             raise error.TestError('interface %s not found' % name)
67
68         match = re.search('UP', out, re.S)
69         return match
70
```

6.42.2.5 def power_Resume.power_Resume.__sanity_check_system (*self*) [private]

Definition at line 71 of file power_Resume.py.

```
72                                    :
73         time.sleep(3)
74
75         iface = 'wlan0'
76         if not self.__is_iface_up(iface):
77             raise error.TestFail('%s failed to come up' % iface)
78
```

6.42.2.6 def power_Resume.power_Resume.run_once (*self*)

Definition at line 79 of file power_Resume.py.

```
80                       :
81         # Some idle time before initiating suspend-to-ram
82         idle_time = random.randint(1, 10)
83         time.sleep(idle_time)
84
85         # Safe enough number, can tweek if necessary
86         time_to_sleep = 10
87
88         # Set the alarm
89         alarm_time = int(utils.get_hwclock_seconds()) + time_to_sleep
90         logging.debug('alarm_time = %d' % alarm_time)
91         utils.set_wake_alarm(alarm_time)
92
93         # Suspend the system to RAM
```

```
94          utils.suspend_to_ram()
95
96          # Calculate the suspend/resume times
97          resume_time = utils.get_hwclock_seconds() - alarm_time
98          suspend_time = \
99              self.__get_end_suspend_time() - self.__get_start_suspend_time()
100
101          # Prepare the results
102          results = {}
103          results['seconds_system_suspend'] = suspend_time
104          results['seconds_system_resume'] = resume_time
105          self.write_perf_keyval(results)
106
107          # Finally, sanity check critical system components
108          self.__sanity_check_system()
```

6.42.3 Member Data Documentation

6.42.3.1 power_Resume.power_Resume.preserve_srcdir = True [static]

Definition at line 37 of file power_Resume.py.

6.42.3.2 int power_Resume.power_Resume.version = 1 [static]

Definition at line 36 of file power_Resume.py.

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

- testsource/power_Resume.py

6.43 realtimecomm_GTalkAudioPlayground.realtimecomm_GTalkAudioPlayground Class Reference

Public Member Functions

- def setup
- def run_verification
- def run_once

Public Attributes

- dep_dir
- performance_results

Static Public Attributes

- int version = 1
- string playground = '/home/autotest/playground'
- string dep = 'realtimecomm_playground'

6.43.1 Detailed Description

Definition at line 38 of file realtimecomm_GTalkAudioPlayground.py.

6.43.2 Member Function Documentation

6.43.2.1 def realtimecomm_GTalkAudioPlayground.realtimecomm_GTalkAudioPlayground.run_once (*self*)

Definition at line 55 of file realtimecomm_GTalkAudioPlayground.py.

```
56                     :
57         self.dep_dir = os.path.join(self.autodir, 'deps', self.dep)
58         sys.path.append(self.dep_dir)
59         import pgutil
60 
61         self.performance_results = {}
62         pgutil.cleanup_playground(self.playground)
63         pgutil.setup_playground(os.path.join(self.dep_dir, 'src'),
64             self.playground, os.path.join(self.bindir, 'options'))
65 
66         # Launch Playground
67         path = os.path.join(self.playground,
68             'buzz/javascript/media/examples')
69         page = 'videoplayground.html'
70         para = 'callType=a'
71         playground_url = "%s/%s?%s" % (path, page, para)
72         # Here we somehow have to use utils.run
73         # Other approaches like utils.system and site_ui.ChromeSession
74         # cause problem in video.
75         # http://code.google.com/p/chromium-os/issues/detail?id=1764
76         utils.run('su chronos -c \'DISPLAY=:0 \
77             XAUTHORITY=/home/chronos/.Xauthority \
78             /opt/google/chrome/chrome \
```

```
79              --no-first-run %s\' &' % playground_url)
80
81          # Collect ctime,stime for GoogleTalkPlugin
82          time.sleep(WARMUP_TIME)
83          gtalk_s = pgutil.get_utime_stime(pgutil.get_pids('GoogleTalkPlugin'))
84          pulse_s = pgutil.get_utime_stime(pgutil.get_pids('pulseaudio'))
85          time.sleep(SLEEP_DURATION)
86          gtalk_e = pgutil.get_utime_stime(pgutil.get_pids('GoogleTalkPlugin'))
87          pulse_e = pgutil.get_utime_stime(pgutil.get_pids('pulseaudio'))
88
89          self.performance_results['ctime_gtalk'] = \
90              pgutil.get_cpu_usage(SLEEP_DURATION, gtalk_e[0] - gtalk_s[0])
91          self.performance_results['stime_gtalk'] = \
92              pgutil.get_cpu_usage(SLEEP_DURATION, gtalk_e[1] - gtalk_s[1])
93          self.performance_results['ctime_pulse'] = \
94              pgutil.get_cpu_usage(SLEEP_DURATION, pulse_e[0] - pulse_s[0])
95          self.performance_results['stime_pulse'] = \
96              pgutil.get_cpu_usage(SLEEP_DURATION, pulse_e[1] - pulse_s[1])
97
98          # Verify log
99          try:
100              self.run_verification()
101         finally:
102             pgutil.cleanup_playground(self.playground, True)
103
104         # Report perf
105         self.write_perf_keyval(self.performance_results)
```

6.43.2.2 def realtimecomm_GTalkAudioPlayground.realtimecomm_GTalkAudioPlayground.run_verification (*self*)

Definition at line 47 of file realtimecomm_GTalkAudioPlayground.py.

```
48                              :
49          if not os.path.exists('/tmp/tmp.log'):
50              raise error.TestFail('GTalk log file not exist!')
51          content = utils.read_file('/tmp/tmp.log')
52          if not "voice state, recv=1 send=1" in content:
53              raise error.TestFail('Error in Audio send/recv!')
54
```

6.43.2.3 def realtimecomm_GTalkAudioPlayground.realtimecomm_GTalkAudioPlayground.setup (*self*)

Definition at line 43 of file realtimecomm_GTalkAudioPlayground.py.

```
44                      :
45          self.job.setup_dep([self.dep])
46
```

6.43.3 Member Data Documentation

6.43.3.1 string realtimecomm_GTalkAudioPlayground.realtimecomm_GTalkAudioPlayground.dep = 'realtimecomm_playground' [static]

Definition at line 41 of file realtimecomm_GTalkAudioPlayground.py.

6.43.3.2 realtimecomm_GTalkAudioPlayground.realtimecomm_GTalkAudioPlayground.dep_dir

Definition at line 56 of file realtimecomm_GTalkAudioPlayground.py.

6.43.3.3 realtimecomm_GTalkAudioPlayground.realtimecomm_GTalkAudioPlayground.performance_results

Definition at line 60 of file realtimecomm_GTalkAudioPlayground.py.

6.43.3.4 string realtimecomm_GTalkAudioPlayground.realtimecomm_GTalkAudioPlayground.playground = '/home/autotest/playground' [static]

Definition at line 40 of file realtimecomm_GTalkAudioPlayground.py.

6.43.3.5 int realtimecomm_GTalkAudioPlayground.realtimecomm_GTalkAudioPlayground.version = 1 [static]

Definition at line 39 of file realtimecomm_GTalkAudioPlayground.py.

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

- testsource/realtimecomm_GTalkAudioPlayground.py

6.44 realtimecomm_GTalkPlayground.realtimecomm_-GTalkPlayground Class Reference

Public Member Functions

- def setup
- def run_verification
- def get_framerate
- def run_once

Public Attributes

- dep_dir
- performance_results

Static Public Attributes

- int version = 1
- string playground = '/home/autotest/playground'
- string dep = 'realtimecomm_playground'

6.44.1 Detailed Description

Definition at line 46 of file realtimecomm_GTalkPlayground.py.

6.44.2 Member Function Documentation

6.44.2.1 def realtimecomm_GTalkPlayground.realtimecomm_GTalkPlayground.get_framerate (*self*, *log*)

Definition at line 74 of file realtimecomm_GTalkPlayground.py.

```
75                                  :
76         d = {}
77         # We get a framerate report every 10 seconds for both streams.
78         # We run for 5 mins, and should get around (5 * 60/10) * 2 = 60
79         # framerate reports for 2 streams.
80         # Ignore the first and last framerate since they are not accurate.
81         l = re.findall(r"Rendered framerate \((.*)\): (\d+\.?\d*) fps", log)
82         if len(l) < 57:
83             raise error.TestFail('Error in Video duration!')
84         for i in range(1, len(l) - 1):
85             if d.has_key(l[i][0]):
86                 d[l[i][0]] = d[l[i][0]] + float(l[i][1])
87             else:
88                 d[l[i][0]] = float(l[i][1])
89         if len(d) != 2:
90             raise error.TestFail('Number of video stream is NOT 2!')
91         # Get framerate for two streams.
92         fps = []
93         for k in d:
94            fps.insert(0, d[k] * 2 / (len(l) - 2))
95         self.performance_results['fps_gtalk_up'] = max(fps[0], fps[1])
```

```
96          self.performance_results['fps_gtalk_down'] = min(fps[0], fps[1])
97          # Very low framerate means something wrong. Video hang or crash.
98          if (min(fps[0], fps[1]) < 5.0):
99              raise error.TestFail('Error in Video framerate.')
100
```

6.44.2.2 def realtimecomm_GTalkPlayground.realtimecomm_GTalkPlayground.run_once (*self*)

Definition at line 101 of file realtimecomm_GTalkPlayground.py.

```
102                     :
103         self.dep_dir = os.path.join(self.autodir, 'deps', self.dep)
104         sys.path.append(self.dep_dir)
105         import pgutil
106
107         self.performance_results = {}
108         pgutil.cleanup_playground(self.playground)
109         pgutil.setup_playground(os.path.join(self.dep_dir, 'src'),
110             self.playground, os.path.join(self.bindir, 'options'))
111
112         # Launch Playground
113         path = os.path.join(self.playground,
114             'buzz/javascript/media/examples')
115         page = 'videoplayground.html'
116         para = 'callType=v'
117         playground_url = "%s/%s?%s" % (path, page, para)
118         # Here we somehow have to use utils.run
119         # Other approaches like utils.system and site_ui.ChromeSession
120         # cause problem in video.
121         # http://code.google.com/p/chromium-os/issues/detail?id=1764
122         utils.run('su chronos -c \'DISPLAY=:0 \
123             XAUTHORITY=/home/chronos/.Xauthority \
124             /opt/google/chrome/chrome \
125             --no-first-run %s\' &' % playground_url)
126
127         # Collect ctime,stime for GoogleTalkPlugin
128         time.sleep(WARMUP_TIME)
129         gtalk_s = pgutil.get_utime_stime(pgutil.get_pids('GoogleTalkPlugin'))
130         chrome_s = pgutil.get_utime_stime(pgutil.get_pids('chrome/chrome'))
131         pulse_s = pgutil.get_utime_stime(pgutil.get_pids('pulseaudio'))
132         time.sleep(SLEEP_DURATION)
133         gtalk_e = pgutil.get_utime_stime(pgutil.get_pids('GoogleTalkPlugin'))
134         chrome_e = pgutil.get_utime_stime(pgutil.get_pids('chrome/chrome'))
135         pulse_e = pgutil.get_utime_stime(pgutil.get_pids('pulseaudio'))
136
137         self.performance_results['ctime_gtalk'] = \
138             pgutil.get_cpu_usage(SLEEP_DURATION, gtalk_e[0] - gtalk_s[0])
139         self.performance_results['stime_gtalk'] = \
140             pgutil.get_cpu_usage(SLEEP_DURATION, gtalk_e[1] - gtalk_s[1])
141         self.performance_results['ctime_chrome'] = \
142             pgutil.get_cpu_usage(SLEEP_DURATION, chrome_e[0] - chrome_s[0])
143         self.performance_results['stime_chrome'] = \
144             pgutil.get_cpu_usage(SLEEP_DURATION, chrome_e[1] - chrome_s[1])
145         self.performance_results['ctime_pulse'] = \
146             pgutil.get_cpu_usage(SLEEP_DURATION, pulse_e[0] - pulse_s[0])
147         self.performance_results['stime_pulse'] = \
148             pgutil.get_cpu_usage(SLEEP_DURATION, pulse_e[1] - pulse_s[1])
149
150         # Verify log
151         try:
152             self.run_verification()
153         finally:
154             pgutil.cleanup_playground(self.playground, True)
```

```
155
156         # Report perf
157         self.write_perf_keyval(self.performance_results)
```

6.44.2.3 def realtimecomm_GTalkPlayground.realtimecomm_GTalkPlayground.run_verification (*self*)

Definition at line 55 of file realtimecomm_GTalkPlayground.py.

```
56                              :
57         # TODO(zhurun): Add more checking and perf data collection.
58         if not os.path.exists('/tmp/tmp.log'):
59             raise error.TestFail('GTalk log file not exist!')
60         content = utils.read_file('/tmp/tmp.log')
61         if not "Found V4L2 capture" in content:
62             raise error.TestFail('V4L2 not found!')
63         if not "video state, recv=1 send=1" in content:
64             raise error.TestFail('Error in Video send/recv!')
65         if not "voice state, recv=1 send=1" in content:
66             raise error.TestFail('Error in Audio send/recv!')
67         if not "Decoded framerate" in content:
68             raise error.TestFail('Error in Video upstream!')
69         if not "Rendered framerate" in content:
70             raise error.TestFail('Error in Video downstream!')
71         # Get framerate
72         self.get_framerate(content)
73
```

6.44.2.4 def realtimecomm_GTalkPlayground.realtimecomm_GTalkPlayground.setup (*self*)

Definition at line 51 of file realtimecomm_GTalkPlayground.py.

```
52                      :
53         self.job.setup_dep([self.dep])
54
```

6.44.3 Member Data Documentation

6.44.3.1 string realtimecomm_GTalkPlayground.realtimecomm_GTalkPlayground.dep = 'realtimecomm_playground' `[static]`

Definition at line 49 of file realtimecomm_GTalkPlayground.py.

6.44.3.2 realtimecomm_GTalkPlayground.realtimecomm_GTalkPlayground.dep_dir

Definition at line 102 of file realtimecomm_GTalkPlayground.py.

6.44.3.3 realtimecomm_GTalkPlayground.realtimecomm_GTalkPlayground.performance_results

Definition at line 106 of file realtimecomm_GTalkPlayground.py.

6.44.3.4 string realtimecomm_GTalkPlayground.realtimecomm_GTalkPlayground.playground = '/home/autotest/playground' [static]

Definition at line 48 of file realtimecomm_GTalkPlayground.py.

6.44.3.5 int realtimecomm_GTalkPlayground.realtimecomm_GTalkPlayground.version = 1 [static]

Definition at line 47 of file realtimecomm_GTalkPlayground.py.

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

- testsource/realtimecomm_GTalkPlayground.py

6.45 unixbench.unixbench Class Reference

Public Member Functions

- def initialize
- def setup
- def run_once
- def cleanup
- def check_for_error
- def postprocess_iteration

Public Attributes

- err
- report_data

Static Public Attributes

- int version = 2

6.45.1 Detailed Description

Definition at line 27 of file unixbench.py.

6.45.2 Member Function Documentation

6.45.2.1 def unixbench.unixbench.check_for_error (*self*, *words*)

Definition at line 70 of file unixbench.py.

```
71                                      :
72         l = len(words)
73         if l >= 3 and words[-3:l] == ['no', 'measured', 'results']:
74             # found a problem so record it in err string
75             key = '_'.join(words[:-3])
76             if self.err is None:
77                 self.err = key
78             else:
79                 self.err = self.err + " " + key
80             return True
81         else:
82             return False
83
```

6.45.2.2 def unixbench.unixbench.cleanup (*self*)

Definition at line 64 of file unixbench.py.

```
65                      :
66         # check err string and possible throw
67         if self.err is not None:
68             raise error.TestError(self.err)
69
```

6.45.2.3 def unixbench.unixbench.initialize (*self*)

Definition at line 30 of file unixbench.py.

```
31                          :
32         self.job.require_gcc()
33         self.err = None
34
35
    # http://www.tux.org/pub/tux/niemi/unixbench/unixbench-4.1.0.tgz
```

6.45.2.4 def unixbench.unixbench.postprocess_iteration (*self*)

Definition at line 84 of file unixbench.py.

```
85                                  :
86         keyval = {}
87         for line in self.report_data:
88             if not line.strip():
89                 break
90
91             words = line.split()
92             # look for problems first
93             if self.check_for_error(words):
94                 continue
95
96             # we should make sure that there are at least
97             # 6 guys before we start accessing the array
98             if len(words) >= 6:
99                 key = '_'.join(words[:-6])
100                key = re.sub('\W', '', key)
101                value = words[-6]
102                keyval[key] = value
103         for line in self.report_data:
104             if 'FINAL SCORE' in line:
105                 keyval['score'] = line.split()[-1]
106                 break
107         self.write_perf_keyval(keyval)
108
109
110 """ Here is a sample report file:
111
112   BYTE UNIX Benchmarks (Version 4.1.0)
113   System -- Linux adrianbg 2.6.18.5 #1 SMP Thu J  Start Benchmark Run: Tue Sep 1
114    9 interactive users.
115    21:03:50 up 5 days,  7:38,  9 users,  load average: 0.71, 0.40, 0.25
116   lrwxrwxrwx 1 root root 4 Aug 15 09:53 /bin/sh -> bash
117   /bin/sh: symbolic link to `bash'
118   /dev/sda6            192149596  91964372  90424536  51% /home
119 Dhrystone 2 using register variables     7918001.7 lps   (10.0 secs, 10 samples)
120 System Call Overhead                     1427272.7 lps   (10.0 secs, 10 samples)
121 Process Creation                          11508.6 lps   (30.0 secs, 3 samples)
122 Execl Throughput                           4159.7 lps   (29.7 secs, 3 samples)
123 File Read 1024 bufsize 2000 maxblocks    1708109.0 KBps  (30.0 secs, 3 samples)
```

```
124 File Write 1024 bufsize 2000 maxblocks   788024.0 KBps  (30.0 secs, 3 samples)
125 File Copy 1024 bufsize 2000 maxblocks    452986.0 KBps  (30.0 secs, 3 samples)
126 File Read 256 bufsize 500 maxblocks      508752.0 KBps  (30.0 secs, 3 samples)
127 File Write 256 bufsize 500 maxblocks     214772.0 KBps  (30.0 secs, 3 samples)
128 File Copy 256 bufsize 500 maxblocks      143989.0 KBps  (30.0 secs, 3 samples)
129 File Read 4096 bufsize 8000 maxblocks    2626923.0 KBps  (30.0 secs, 3 samples)
130 File Write 4096 bufsize 8000 maxblocks   1175070.0 KBps  (30.0 secs, 3 samples)
131 File Copy 4096 bufsize 8000 maxblocks    793041.0 KBps  (30.0 secs, 3 samples)
132 Shell Scripts (1 concurrent)               4417.4 lpm   (60.0 secs, 3 samples)
133 Shell Scripts (8 concurrent)               1109.0 lpm   (60.0 secs, 3 samples)
134 Shell Scripts (16 concurrent)               578.3 lpm   (60.0 secs, 3 samples)
135 Arithmetic Test (type = short)           1843690.0 lps   (10.0 secs, 3 samples)
136 Arithmetic Test (type = int)             1873615.8 lps   (10.0 secs, 3 samples)
137 Arithmetic Test (type = long)            1888345.9 lps   (10.0 secs, 3 samples)
138 Arithmetic Test (type = float)           616260.3 lps   (10.0 secs, 3 samples)
139 Arithmetic Test (type = double)          615942.1 lps   (10.0 secs, 3 samples)
140 Arithoh                                  18864899.5 lps   (10.0 secs, 3 samples)
141 Dc: sqrt(2) to 99 decimal places         161726.0 lpm   (30.0 secs, 3 samples)
142 Recursion Test--Tower of Hanoi            89229.3 lps   (20.0 secs, 3 samples)
143
144
145                       INDEX VALUES
146 TEST                                        BASELINE     RESULT      INDEX
147
148 Dhrystone 2 using register variables        116700.0  7918001.7      678.5
149 Double-Precision Whetstone                     55.0     1948.2      354.2
150 Execl Throughput                               43.0     4159.7      967.4
151 File Copy 1024 bufsize 2000 maxblocks         3960.0   452986.0     1143.9
152 File Copy 256 bufsize 500 maxblocks           1655.0   143989.0      870.0
153 File Copy 4096 bufsize 8000 maxblocks         5800.0   793041.0     1367.3
154 Pipe Throughput                              12440.0  1048491.9      842.8
155 Pipe-based Context Switching                  4000.0   300778.3      751.9
156 Process Creation                               126.0    11508.6      913.4
157 Shell Scripts (8 concurrent)                     6.0     1109.0     1848.3
158 System Call Overhead                         15000.0  1427272.7      951.5
159                                                                   =========
160      FINAL SCORE                                                     902.1
161 """
```

6.45.2.5 def unixbench.unixbench.run_once (*self*, *args* = *''*, *stepsecs* = 0)

Definition at line 47 of file unixbench.py.

```
48                                          :
49         vars = ('TMPDIR=\"%s\" RESULTDIR=\"%s\" FLAVOR=Linux' %
50                (self.tmpdir, self.resultsdir))
51         if stepsecs:
52             # change time per subtest from unixbench's defaults of
53             #   10 secs for small tests, 30 secs for bigger tests
54             vars += ' systime=%i looper=%i seconds=%i'\
55                     ' dhrytime=%i arithtime=%i' \
56                     % ((stepsecs,)*5)
57
58         os.chdir(self.srcdir)
59         utils.system(vars + ' ./Run ' + args)
60
61         report_path = os.path.join(self.resultsdir, 'report')
62         self.report_data = open(report_path).readlines()[9:]
63
```

6.45.2.6 def unixbench.unixbench.setup (*self*, *tarball* = 'unixbench-4.1.0.tar.bz2')

Definition at line 36 of file unixbench.py.

```
37                                                      :
38         tarball = utils.unmap_url(self.bindir, tarball, self.tmpdir)
39         utils.extract_tarball_to_dir(tarball, self.srcdir)
40         os.chdir(self.srcdir)
41
42         utils.system('patch -p1 < ../unixbench.patch')
43         utils.system('patch -p1 < ../Makefile.patch')
44         utils.system('make')
45         utils.system('rm pgms/select')
46
```

6.45.3 Member Data Documentation

6.45.3.1 unixbench.unixbench.err

Definition at line 32 of file unixbench.py.

6.45.3.2 unixbench.unixbench.report_data

Definition at line 61 of file unixbench.py.

6.45.3.3 int unixbench.unixbench.version = 2 [static]

Definition at line 28 of file unixbench.py.

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

- testsource/unixbench.py

Chapter 7

File Documentation

7.1 testsource/audiovideo_FFMPEG.py File Reference

Classes

- class audiovideo_FFMPEG.audiovideo_FFMPEG

Packages

- package audiovideo_FFMPEG

  This test exercises the ffmpeg-based software video decoder.

7.2 testsource/audiovideo_V4L2.py File Reference

Classes

- class audiovideo_V4L2.audiovideo_V4L2

Packages

- package audiovideo_V4L2

  Exercises v4l2 camera devices to verify required operations.

7.3 testsource/compilebench.py File Reference

Classes

- class compilebench.compilebench

Packages

- package compilebench

  Benchmark the filesystem performance.

Variables

- string compilebench.test_name = 'compilebench'

7.4 testsource/desktopui_SunSpiderBench.py File Reference

Classes

- class desktopui_SunSpiderBench.desktopui_SunSpiderBench

Packages

- package desktopui_SunSpiderBench

  Measure the performance of Chrome's JavaScript.

7.5 testsource/desktopui_V8Bench.py File Reference

Classes

- class desktopui_V8Bench.desktopui_V8Bench

Packages

- package desktopui_V8Bench

  Benchmark javascript operations of a web browser.

7.6 testsource/disktest.py File Reference

Classes

- class disktest.disktest

Packages

- package disktest

  Verify the integrity of the disk and disk controller.

7.7 testsource/firmware_RomSize.py File Reference

Classes

- class firmware_RomSize.firmware_RomSize

Packages

- package firmware_RomSize

  Ensure the firmware size is large enough.

7.8 testsource/firmware_VbootCrypto.py File Reference

Classes

- class firmware_VbootCrypto.firmware_VbootCrypto

Packages

- package firmware_VbootCrypto

  Verifies Firmware Verified Boot Reference Implementation, it's components, and crypto performance.

7.9 testsource/gl_Bench.py File Reference

Classes

- class gl_Bench.gl_Bench

Packages

- package gl_Bench

  Benchmark the graphics library performance.

7.10 testsource/graphics_GLAPICheck.py File Reference

Classes

- class graphics_GLAPICheck.graphics_GLAPICheck

Packages

- package graphics_GLAPICheck

  Verify correctness of OpenGL/GLES and X11 versions/extensions.

7.11 testsource/graphics_SanAngeles.py File Reference

Classes

- class graphics_SanAngeles.graphics_SanAngeles

Packages

- package graphics_SanAngeles

  Benchmark OpenGL object rendering.

7.12 testsource/graphics_TearTest.py File Reference

Classes

- class graphics_TearTest.graphics_TearTest

Packages

- package graphics_TearTest

  Verify Chrome does not tear with veritcal synchronization.

Functions

- def graphics_TearTest.html_button

Variables

- string graphics_TearTest.TEMPLATE

7.13 testsource/hardware_Backlight.py File Reference

Classes

- class hardware_Backlight.hardware_Backlight

Packages

- package hardware_Backlight

  Verify that the backlight can be adjusted in software.

Functions

- def hardware_Backlight.backlight_tool

7.14 testsource/hardware_BluetoothSemiAuto.py File Reference

Classes

- class hardware_BluetoothSemiAuto.Agent
- class hardware_BluetoothSemiAuto.hardware_BluetoothSemiAuto

Packages

- package hardware_BluetoothSemiAuto

  Verify the basic functionality of the Bluetooth adapter.

Variables

- string hardware_BluetoothSemiAuto._QUESTION_START
- string hardware_BluetoothSemiAuto._HREF_START = ''''''
- string hardware_BluetoothSemiAuto._HREF_END = ''''''

7.15 testsource/hardware_Components.py File Reference

Classes

- class hardware_Components.hardware_Components

Packages

- package hardware_Components

  Ensure system components are in the approved components database.

7.16 testsource/hardware_DiskSize.py File Reference

Classes

- class hardware_DiskSize.hardware_DiskSize

Packages

- package hardware_DiskSize

  Ensure the hard disk is large enough.

7.17 testsource/hardware_KeyboardAssembly.py File Reference

Classes

- class hardware_KeyboardAssembly.hardware_KeyboardAssembly

Packages

- package hardware_KeyboardAssembly

  Verify that keyboard keys function properly.

7.18 testsource/hardware_MemoryThroughput.py File Reference

Classes

- class hardware_MemoryThroughput.hardware_MemoryThroughput

Packages

- package hardware_MemoryThroughput

  Benchmark sequential and random access mode memory throughput.

7.19 testsource/hardware_MemoryTotalSize.py File Reference

Classes

- class hardware_MemoryTotalSize.hardware_MemoryTotalSize

Packages

- package hardware_MemoryTotalSize

  Verify there is enough memory to run Chrome OS.

7.20 testsource/hardware_Resolution.py File Reference

Classes

- class hardware_Resolution.hardware_Resolution

Packages

- package hardware_Resolution

  Determine if the current screen resolution is supported.

Variables

- string hardware_Resolution.__author__ = 'kdlucas@chromium.org (Kelly Lucas)'

7.21 testsource/hardware_SAT.py File Reference

Classes

- class hardware_SAT.hardware_SAT

Packages

- package hardware_SAT

  Stress test hardware devices.

7.22 testsource/hardware_SsdDetection.py File Reference

Classes

- class hardware_SsdDetection.hardware_SsdDetection

Packages

- package hardware_SsdDetection

  Determine if main disk is a solid state device.

7.23 testsource/hardware_StorageFio.py File Reference

Classes

- class hardware_StorageFio.hardware_StorageFio

Packages

- package hardware_StorageFio

  Benchmark storage performance using an unmounted root partition.

7.24 testsource/hardware_Touchpad.py File Reference

Classes

- class hardware_Touchpad.hardware_Touchpad

Packages

- package hardware_Touchpad

  Verify all touchpad functions.

7.25 testsource/hardware_VideoOutSemiAuto.py File Reference

Classes

- class hardware_VideoOutSemiAuto.hardware_VideoOutSemiAuto

Packages

- package hardware_VideoOutSemiAuto

  Verify external video ports are configurable with Chrome OS.

7.26 testsource/ltp.py File Reference

Classes

- class ltp.ltp

Packages

- package ltp

  Verify kernel system calls are operating correctly.

7.27 testsource/mainpage.txt File Reference

7.28 testsource/network_DisableInterface.py File Reference

Classes

- class network_DisableInterface.network_DisableInterface

Packages

- package network_DisableInterface

  Verify a network interface can be disabled.

7.29 testsource/network_WiFiCaps.py File Reference

Classes

- class network_WiFiCaps.network_WiFiCaps

Packages

- package network_WiFiCaps

  Verify that WiFi devices have the required capabilities.

7.30 testsource/platform_AesThroughput.py File Reference

Classes

- class platform_AesThroughput.platform_AesThroughput

Packages

- package platform_AesThroughput

  Benchmark processor performance using OpenSSL using AES options.

7.31 testsource/platform_BootPerf.py File Reference

Classes

- class platform_BootPerf.platform_BootPerf

Packages

- package platform_BootPerf

  Collect boot performance metrics from the last system reboot.

7.32 testsource/platform_KernelVersion.py File Reference

Classes

- class platform_KernelVersion.platform_KernelVersion

Packages

- package platform_KernelVersion

  Ensure the running kernel is supported.

7.33 testsource/power_BatteryCharge.py File Reference

Classes

- class power_BatteryCharge.power_BatteryCharge

Packages

- package power_BatteryCharge

  Measure the time required to charge the battery.

7.34 testsource/power_CPUFreq.py File Reference

Classes

- class power_CPUFreq.power_CPUFreq
- class power_CPUFreq.cpufreq

Packages

- package power_CPUFreq

  Verify that supported CPU frequencies can be set.

7.35 testsource/power_CPUIdle.py File Reference

Classes

- class power_CPUIdle.power_CPUIdle
- class power_CPUIdle.cpus
- class power_CPUIdle.cpuidle
- class power_CPUIdle.cpuidle_state

Packages

- package power_CPUIdle

  Ensure the processor drops into idle state when it is idle.

7.36 testsource/power_Draw.py File Reference

Classes

- class power_Draw.power_Draw

Packages

- package power_Draw

  Measure how much power is drawn over a given amount of time.

7.37 testsource/power_LoadTest.py File Reference

Classes

- class power_LoadTest.power_LoadTest

Packages

- package power_LoadTest

  Measure power draw when system is under load.

Variables

- dictionary power_LoadTest.params_dict

7.38 testsource/power_Resume.py File Reference

Classes

- class power_Resume.power_Resume

Packages

- package power_Resume

  Measure the amount of time it takes to resume from suspend.

7.39 testsource/realtimecomm_GTalkAudioPlayground.py File Reference

Classes

- class realtimecomm_GTalkAudioPlayground.realtimecomm_GTalkAudioPlayground

Packages

- package realtimecomm_GTalkAudioPlayground

  Verify that Google Talk Plugin Audio works with Chrome.

Variables

- int realtimecomm_GTalkAudioPlayground.WARMUP_TIME = 30
- int realtimecomm_GTalkAudioPlayground.SLEEP_DURATION = 90

7.40 testsource/realtimecomm_GTalkPlayground.py File Reference

Classes

- class realtimecomm_GTalkPlayground.realtimecomm_GTalkPlayground

Packages

- package realtimecomm_GTalkPlayground

  Verify that Google Talk Plugin executes.

Variables

- int realtimecomm_GTalkPlayground.WARMUP_TIME = 60
- int realtimecomm_GTalkPlayground.SLEEP_DURATION = 260

7.41 testsource/unixbench.py File Reference

Classes

- class unixbench.unixbench

Packages

- package unixbench

  Measure system level performance.

Index

_HREF_END
    hardware_BluetoothSemiAuto, 28
_HREF_START
    hardware_BluetoothSemiAuto, 28
_QUESTION_START
    hardware_BluetoothSemiAuto, 29
__RunFio
    hardware_StorageFio::hardware_StorageFio, 135
__author__
    hardware_Resolution, 35
__base_path
    power_CPUFreq::cpufreq, 76
    power_CPUIdle::cpuidle, 78
    power_CPUIdle::cpuidle_state, 80
    power_CPUIdle::cpus, 81
__check_extensions
    graphics_GLAPICheck::graphics_-GLAPICheck, 100
__check_gl
    graphics_GLAPICheck::graphics_-GLAPICheck, 100
__check_gl_extensions_1x
    graphics_GLAPICheck::graphics_-GLAPICheck, 101
__check_gl_extensions_2x
    graphics_GLAPICheck::graphics_-GLAPICheck, 101
__check_gles
    graphics_GLAPICheck::graphics_-GLAPICheck, 101
__check_gles_extensions
    graphics_GLAPICheck::graphics_-GLAPICheck, 102
__check_x_extensions
    graphics_GLAPICheck::graphics_-GLAPICheck, 102
__configure_and_check_output
    hardware_VideoOutSemiAuto::hardware_-VideoOutSemiAuto, 140
__cpus
    power_CPUIdle::cpus, 81
__description
    hardware_StorageFio::hardware_StorageFio, 137
__filename
    hardware_StorageFio::hardware_StorageFio, 137
__filesize
    hardware_StorageFio::hardware_StorageFio, 137
__find_free_root_partition
    hardware_StorageFio::hardware_StorageFio, 133
__format_results
    compilebench::compilebench, 70
__generate_test_cases
    firmware_VbootCrypto::firmware_-VbootCrypto, 92
__get_device_description
    hardware_StorageFio::hardware_StorageFio, 133
__get_end_suspend_time
    power_Resume::power_Resume, 174
__get_file_size
    hardware_StorageFio::hardware_StorageFio, 134
__get_last_msg_time
    power_Resume::power_Resume, 174
__get_start_suspend_time
    power_Resume::power_Resume, 174
__image_verification_test
    firmware_VbootCrypto::firmware_-VbootCrypto, 92
__init__
    power_CPUFreq::cpufreq, 73
    power_CPUIdle::cpuidle, 77
    power_CPUIdle::cpuidle_state, 79
    power_CPUIdle::cpus, 81
__is_iface_up
    power_Resume::power_Resume, 175
__name
    power_CPUIdle::cpuidle_state, 80
__output_connected
    hardware_VideoOutSemiAuto::hardware_-VideoOutSemiAuto, 140
__output_is_set
    hardware_VideoOutSemiAuto::hardware_-VideoOutSemiAuto, 141
__output_result_keyvals

firmware_VbootCrypto::firmware_-
VbootCrypto, 92
__parse_disk_login_prompt_ready
platform_BootPerf::platform_BootPerf, 154
__parse_fio
hardware_StorageFio::hardware_StorageFio,
134
__parse_firmware_boot_time
platform_BootPerf::platform_BootPerf, 154
__parse_iwcap
network_WiFiCaps::network_WiFiCaps, 149
__parse_syslog
platform_BootPerf::platform_BootPerf, 155
__parse_uptime
platform_BootPerf::platform_BootPerf, 155
__query_for_output
hardware_VideoOutSemiAuto::hardware_-
VideoOutSemiAuto, 141
__read_file
power_CPUFreq::cpufreq, 73
power_CPUIdle::cpuidle_state, 79
__rollback_tests
firmware_VbootCrypto::firmware_-
VbootCrypto, 92
__rsa_benchmark
firmware_VbootCrypto::firmware_-
VbootCrypto, 93
__rsa_test
firmware_VbootCrypto::firmware_-
VbootCrypto, 93
__run_iwcap
network_WiFiCaps::network_WiFiCaps, 149
__run_x_cmd
graphics_GLAPICheck::graphics_-
GLAPICheck, 103
__sanity_check_system
power_Resume::power_Resume, 175
__save_files_list
power_CPUFreq::cpufreq, 76
__sha_benchmark
firmware_VbootCrypto::firmware_-
VbootCrypto, 93
__sha_test
firmware_VbootCrypto::firmware_-
VbootCrypto, 94
__splicing_tests
firmware_VbootCrypto::firmware_-
VbootCrypto, 94
__states
power_CPUIdle::cpuidle, 78
__verify_image_benchmark
firmware_VbootCrypto::firmware_-
VbootCrypto, 94
__write_file
power_CPUFreq::cpufreq, 74
_ah_charge_start
power_LoadTest::power_LoadTest, 172
_approved
hardware_Components::hardware_-
Components, 119
_cids
hardware_Components::hardware_-
Components, 119
_cpufreq_stats
power_LoadTest::power_LoadTest, 172
_cpuidle_stats
power_LoadTest::power_LoadTest, 172
_do_wait
power_LoadTest::power_LoadTest, 167
_ext_path
power_LoadTest::power_LoadTest, 172
_failures
hardware_Components::hardware_-
Components, 119
_ifconfig
network_DisableInterface::network_-
DisableInterface, 148
_import_site_config
ltp::ltp, 144
_is_network_iface_running
power_LoadTest::power_LoadTest, 167
_loop_count
power_LoadTest::power_LoadTest, 172
_loop_time
power_LoadTest::power_LoadTest, 172
_low_battery_threshold
power_LoadTest::power_LoadTest, 172
_mseconds
power_LoadTest::power_LoadTest, 172
_not_present
hardware_Components::hardware_-
Components, 120
_pci_cids
hardware_Components::hardware_-
Components, 120
_percent_current_charge
power_LoadTest::power_LoadTest, 168
_power_status
power_LoadTest::power_LoadTest, 172
_pp
hardware_Components::hardware_-
Components, 120
_scroll_by_pixels
power_LoadTest::power_LoadTest, 172
_scroll_interval_ms
power_LoadTest::power_LoadTest, 172
_scroll_loop
power_LoadTest::power_LoadTest, 172

_should_scroll
    power_LoadTest::power_LoadTest, 173
_should_scroll_up
    power_LoadTest::power_LoadTest, 173
_system
    hardware_Components::hardware_-
        Components, 120
_testServer
    desktopui_SunSpiderBench::desktopui_-
        SunSpiderBench, 84
    desktopui_V8Bench::desktopui_V8Bench, 86
    power_LoadTest::power_LoadTest, 173
_test_url
    desktopui_SunSpiderBench::desktopui_-
        SunSpiderBench, 84
    desktopui_V8Bench::desktopui_V8Bench, 86
_tmp_keyvals
    power_LoadTest::power_LoadTest, 173
_usb_cids
    hardware_Components::hardware_-
        Components, 120
_usb_stats
    power_LoadTest::power_LoadTest, 173
_verbose
    power_LoadTest::power_LoadTest, 173
_wh_energy_start
    power_LoadTest::power_LoadTest, 173
_write_ext_params
    power_LoadTest::power_LoadTest, 168

assert_mandatory_controls
    audiovideo_V4L2::audiovideo_V4L2, 68
audiovideo_FFMPEG, 13
audiovideo_FFMPEG::audiovideo_FFMPEG, 61
    max_tpf_audio, 63
    min_fps_video, 63
    performance_results, 63
    run_once, 61
    run_testcase, 61
    setup, 63
    version, 63
audiovideo_V4L2, 15
audiovideo_V4L2::audiovideo_V4L2, 64
    assert_mandatory_controls, 68
    executable, 68
    find_video_capture_devices, 64
    preserve_srcdir, 68
    run_once, 65
    run_v4l2_capture_test, 65
    run_v4l2_capture_tests, 65
    run_v4l2_default_capture_test, 66
    run_v4l2_unittests, 66
    setup, 68
    support_readwrite, 68
    support_streaming, 68
    supported_controls, 69
    supported_formats, 69
    unittest_passed, 68
    v4l2_devices, 69
    v4l2_major_dev_num, 69
    v4l2_minor_dev_num_max, 69
    v4l2_minor_dev_num_min, 69
    version, 69

backlight_tool
    hardware_Backlight, 27

Cancel
    hardware_BluetoothSemiAuto::Agent, 59
charge_full_design
    power_BatteryCharge::power_BatteryCharge,
        160
check_approved_part_id_existence
    hardware_Components::hardware_-
        Components, 115
check_component
    hardware_Components::hardware_-
        Components, 115
check_for_error
    unixbench::unixbench, 184
chunk_mb
    disktest::disktest, 88
cleanup
    desktopui_SunSpiderBench::desktopui_-
        SunSpiderBench, 83
    desktopui_V8Bench::desktopui_V8Bench, 85
    hardware_BluetoothSemiAuto::hardware_-
        BluetoothSemiAuto, 110
    power_LoadTest::power_LoadTest, 168
    unixbench::unixbench, 184
compilebench, 16
    test_name, 16
compilebench::compilebench, 70
    __format_results, 70
    run_once, 71
    setup, 71
    tarball, 71
    version, 71

dep
    realtimecomm_-
        GTalkAudioPlayground::realtimecomm_-
        GTalkAudioPlayground, 178
    realtimecomm_-
        GTalkPlayground::realtimecomm_-
        GTalkPlayground, 182
dep_dir

realtimecomm_-
GTalkAudioPlayground::realtimecomm_-
GTalkAudioPlayground, 178
realtimecomm_-
GTalkPlayground::realtimecomm_-
GTalkPlayground, 182
desktopui_SunSpiderBench, 17
desktopui_SunSpiderBench::desktopui_-
SunSpiderBench, 83
_testServer, 84
_test_url, 84
cleanup, 83
initialize, 83
run_once, 83
setup, 84
version, 84
desktopui_V8Bench, 18
desktopui_V8Bench::desktopui_V8Bench, 85
_testServer, 86
_test_url, 86
cleanup, 85
initialize, 85
run_once, 85
setup, 86
version, 86
disktest, 19
disktest::disktest, 87
chunk_mb, 88
execute, 87
initialize, 88
memory_mb, 88
preserve_srcdir, 88
setup, 88
test_one_disk_chunk, 88
version, 89
do_connect
hardware_BluetoothSemiAuto::hardware_-
BluetoothSemiAuto, 110

err
unixbench::unixbench, 187
error_message
graphics_GLAPICheck::graphics_-
GLAPICheck, 104
executable
audiovideo_V4L2::audiovideo_V4L2, 68
execute
disktest::disktest, 87
export_stats
platform_AesThroughput::platform_-
AesThroughput, 151

find_video_capture_devices
audiovideo_V4L2::audiovideo_V4L2, 64
firmware_RomSize, 20
firmware_RomSize::firmware_RomSize, 90
run_once, 90
version, 90
firmware_VbootCrypto, 21
firmware_VbootCrypto::firmware_VbootCrypto,
91
__generate_test_cases, 92
__image_verification_test, 92
__output_result_keyvals, 92
__rollback_tests, 92
__rsa_benchmark, 93
__rsa_test, 93
__sha_benchmark, 93
__sha_test, 94
__splicing_tests, 94
__verify_image_benchmark, 94
keyvals, 96
preserve_srcdir, 96
results, 96
run_benchmarks, 95
run_crypto, 95
run_once, 95
run_rollback, 95
run_splicing, 95
run_verification, 96
setup, 96
version, 96

get_available_frequencies
power_CPUFreq::cpufreq, 74
get_available_governors
power_CPUFreq::cpufreq, 74
get_current_frequency
power_CPUFreq::cpufreq, 74
get_current_governor
power_CPUFreq::cpufreq, 74
get_framerate
realtimecomm_-
GTalkPlayground::realtimecomm_-
GTalkPlayground, 180
get_part_id_audio_codec
hardware_Components::hardware_-
Components, 115
get_part_id_bios
hardware_Components::hardware_-
Components, 116
get_part_id_cpu
hardware_Components::hardware_-
Components, 116
get_part_id_embedded_controller
hardware_Components::hardware_-
Components, 116
get_part_id_ethernet

hardware_Components::hardware_-
Components, 116
get_part_id_flash_chip
hardware_Components::hardware_-
Components, 117
get_part_id_storage
hardware_Components::hardware_-
Components, 117
get_part_id_wireless
hardware_Components::hardware_-
Components, 117
get_resolution
hardware_Resolution::hardware_Resolution,
127
get_vendor_id_touchpad
hardware_Components::hardware_-
Components, 118
get_vendor_id_webcam
hardware_Components::hardware_-
Components, 118
gl_Bench, 22
gl_Bench::gl_Bench, 98
preserve_srcdir, 99
results, 99
run_once, 98
setup, 98
version, 99
graphics_GLAPICheck, 23
graphics_GLAPICheck::graphics_GLAPICheck,
100
__check_extensions, 100
__check_gl, 100
__check_gl_extensions_1x, 101
__check_gl_extensions_2x, 101
__check_gles, 101
__check_gles_extensions, 102
__check_x_extensions, 102
__run_x_cmd, 103
error_message, 104
preserve_srcdir, 104
run_once, 103
setup, 104
version, 104
graphics_SanAngeles, 24
graphics_SanAngeles::graphics_SanAngeles, 105
preserve_srcdir, 106
run_once, 105
setup, 105
version, 106
graphics_TearTest, 25
html_button, 25
TEMPLATE, 26
graphics_TearTest::graphics_TearTest, 107
run_once, 107
setup, 108
version, 108

handle_error
hardware_BluetoothSemiAuto::hardware_-
BluetoothSemiAuto, 111
handle_reply
hardware_BluetoothSemiAuto::hardware_-
BluetoothSemiAuto, 111
hardware_Backlight, 27
backlight_tool, 27
hardware_Backlight::hardware_Backlight, 109
run_once, 109
version, 109
hardware_BluetoothSemiAuto, 28
_HREF_END, 28
_HREF_START, 28
_QUESTION_START, 29
hardware_BluetoothSemiAuto::Agent, 59
Cancel, 59
in_signature, 60
Release, 59
RequestPinCode, 60
hardware_BluetoothSemiAuto::hardware_-
BluetoothSemiAuto, 110
cleanup, 110
do_connect, 110
handle_error, 111
handle_reply, 111
initialize, 111
mainloop, 112
run_once, 112
version, 112
hardware_Components, 30
hardware_Components::hardware_Components,
114
_approved, 119
_cids, 119
_failures, 119
_not_present, 120
_pci_cids, 120
_pp, 120
_system, 120
_usb_cids, 120
check_approved_part_id_existence, 115
check_component, 115
get_part_id_audio_codec, 115
get_part_id_bios, 116
get_part_id_cpu, 116
get_part_id_embedded_controller, 116
get_part_id_ethernet, 116
get_part_id_flash_chip, 117
get_part_id_storage, 117
get_part_id_wireless, 117

get_vendor_id_touchpad, 118
get_vendor_id_webcam, 118
initialize, 118
pformat, 118
run_once, 118
version, 120
hardware_DiskSize, 31
hardware_DiskSize::hardware_DiskSize, 121
run_once, 121
version, 121
hardware_KeyboardAssembly, 32
hardware_KeyboardAssembly::hardware_-KeyboardAssembly, 122
preserve_srcdir, 122
run_once, 122
version, 122
hardware_MemoryThroughput, 33
hardware_MemoryThroughput::hardware_-MemoryThroughput, 124
preserve_srcdir, 125
results, 125
run_once, 124
setup, 124
version, 125
hardware_MemoryTotalSize, 34
hardware_MemoryTotalSize::hardware_-MemoryTotalSize, 126
run_once, 126
version, 126
hardware_Resolution, 35
__author__, 35
hardware_Resolution::hardware_Resolution, 127
get_resolution, 127
run_once, 128
version, 128
hardware_SAT, 36
hardware_SAT::hardware_SAT, 129
run_once, 129
setup, 129
version, 130
hardware_SsdDetection, 37
hardware_SsdDetection::hardware_SsdDetection, 131
run_once, 131
setup, 131
version, 132
hardware_StorageFio, 38
hardware_StorageFio::hardware_StorageFio, 133
__RunFio, 135
__description, 137
__filename, 137
__filesize, 137
__find_free_root_partition, 133
__get_device_description, 133
__get_file_size, 134
__parse_fio, 134
initialize, 135
run_once, 136
setup, 137
version, 137
hardware_Touchpad, 39
hardware_Touchpad::hardware_Touchpad, 139
preserve_srcdir, 139
run_once, 139
version, 139
hardware_VideoOutSemiAuto, 40
hardware_VideoOutSemiAuto::hardware_-VideoOutSemiAuto, 140
__configure_and_check_output, 140
__output_connected, 140
__output_is_set, 141
__query_for_output, 141
HDMI_ID, 142
RECONFIG_PATH, 142
run_once, 142
version, 142
VGA_ID, 143
XRANDR_PATH, 143
HDMI_ID
hardware_VideoOutSemiAuto::hardware_-VideoOutSemiAuto, 142
html_button
graphics_TearTest, 25

idle_time
power_CPUIdle::cpuidle, 77
power_CPUIdle::cpuidle_state, 79
power_CPUIdle::cpus, 81
in_signature
hardware_BluetoothSemiAuto::Agent, 60
initial_charge
power_BatteryCharge::power_BatteryCharge, 160
initialize
desktopui_SunSpiderBench::desktopui_-SunSpiderBench, 83
desktopui_V8Bench::desktopui_V8Bench, 85
disktest::disktest, 88
hardware_BluetoothSemiAuto::hardware_-BluetoothSemiAuto, 111
hardware_Components::hardware_-Components, 118
hardware_StorageFio::hardware_StorageFio, 135
ltp::ltp, 144
power_BatteryCharge::power_BatteryCharge, 158
unixbench::unixbench, 185

is_iface_up
    network_DisableInterface::network_-
        DisableInterface, 147

keyvals
    firmware_VbootCrypto::firmware_-
        VbootCrypto, 96

ltp, 41
ltp::ltp, 144
    _import_site_config, 144
    initialize, 144
    run_once, 145
    setup, 145
    site_ignore_tests, 146
    version, 146

mainloop
    hardware_BluetoothSemiAuto::hardware_-
        BluetoothSemiAuto, 112
max_run_time
    power_BatteryCharge::power_BatteryCharge,
        160
max_tpf_audio
    audiovideo_FFMPEG::audiovideo_FFMPEG,
        63
memory_mb
    disktest::disktest, 88
min_fps_video
    audiovideo_FFMPEG::audiovideo_FFMPEG,
        63

network_DisableInterface, 42
network_DisableInterface::network_-
        DisableInterface, 147
    _ifconfig, 148
    is_iface_up, 147
    run_once, 147
    version, 148
network_WiFiCaps, 43
network_WiFiCaps::network_WiFiCaps, 149
    __parse_iwcap, 149
    __run_iwcap, 149
    run_once, 149
    setup, 150
    version, 150

on_ac
    power_BatteryCharge::power_BatteryCharge,
        158
openssl_speed
    platform_AesThroughput::platform_-
        AesThroughput, 151

params_dict
    power_LoadTest, 52
parse_results
    platform_AesThroughput::platform_-
        AesThroughput, 151
performance_results
    audiovideo_FFMPEG::audiovideo_FFMPEG,
        63
    realtimecomm_-
        GTalkAudioPlayground::realtimecomm_-
        GTalkAudioPlayground, 179
    realtimecomm_-
        GTalkPlayground::realtimecomm_-
        GTalkPlayground, 182
pformat
    hardware_Components::hardware_-
        Components, 118
platform_AesThroughput, 44
platform_AesThroughput::platform_-
        AesThroughput, 151
    export_stats, 151
    openssl_speed, 151
    parse_results, 151
    results, 153
    run_once, 152
    setup, 152
    update_stats, 152
    version, 153
platform_BootPerf, 45
platform_BootPerf::platform_BootPerf, 154
    __parse_disk_login_prompt_ready, 154
    __parse_firmware_boot_time, 154
    __parse_syslog, 155
    __parse_uptime, 155
    run_once, 156
    version, 156
platform_KernelVersion, 46
platform_KernelVersion::platform_KernelVersion,
        157
    run_once, 157
    version, 157
playground
    realtimecomm_-
        GTalkAudioPlayground::realtimecomm_-
        GTalkAudioPlayground, 179
    realtimecomm_-
        GTalkPlayground::realtimecomm_-
        GTalkPlayground, 182
postprocess_iteration
    power_BatteryCharge::power_BatteryCharge,
        158
    power_LoadTest::power_LoadTest, 168
    unixbench::unixbench, 185
power_BatteryCharge, 47
power_BatteryCharge::power_BatteryCharge, 158

charge_full_design, 160
initial_charge, 160
initialize, 158
max_run_time, 160
on_ac, 158
postprocess_iteration, 158
remaining_time, 160
run_once, 159
status, 160
version, 160
power_CPUFreq, 48
power_CPUFreq::cpufreq, 73
__base_path, 76
__init__, 73
__read_file, 73
__save_files_list, 76
__write_file, 74
get_available_frequencies, 74
get_available_governors, 74
get_current_frequency, 74
get_current_governor, 74
restore_state, 75
save_state, 75
set_frequency, 75
set_governor, 75
power_CPUFreq::power_CPUFreq, 162
run_once, 162
version, 163
power_CPUIdle, 49
power_CPUIdle::cpuidle, 77
__base_path, 78
__init__, 77
__states, 78
idle_time, 77
power_CPUIdle::cpuidle_state, 79
__base_path, 80
__init__, 79
__name, 80
__read_file, 79
idle_time, 79
power_CPUIdle::cpus, 81
__base_path, 81
__cpus, 81
__init__, 81
idle_time, 81
power_CPUIdle::power_CPUIdle, 164
run_once, 164
version, 164
power_Draw, 50
power_Draw::power_Draw, 165
run_once, 165
version, 165
power_LoadTest, 51
params_dict, 52
power_LoadTest::power_LoadTest, 166
_ah_charge_start, 172
_cpufreq_stats, 172
_cpuidle_stats, 172
_do_wait, 167
_ext_path, 172
_is_network_iface_running, 167
_loop_count, 172
_loop_time, 172
_low_battery_threshold, 172
_mseconds, 172
_percent_current_charge, 168
_power_status, 172
_scroll_by_pixels, 172
_scroll_interval_ms, 172
_scroll_loop, 172
_should_scroll, 173
_should_scroll_up, 173
_testServer, 173
_tmp_keyvals, 173
_usb_stats, 173
_verbose, 173
_wh_energy_start, 173
_write_ext_params, 168
cleanup, 168
postprocess_iteration, 168
run_once, 169
setup, 171
version, 173
power_Resume, 53
power_Resume::power_Resume, 174
__get_end_suspend_time, 174
__get_last_msg_time, 174
__get_start_suspend_time, 174
__is_iface_up, 175
__sanity_check_system, 175
preserve_srcdir, 176
run_once, 175
version, 176
preserve_srcdir
audiovideo_V4L2::audiovideo_V4L2, 68
disktest::disktest, 88
firmware_VbootCrypto::firmware_-VbootCrypto, 96
gl_Bench::gl_Bench, 99
graphics_GLAPICheck::graphics_-GLAPICheck, 104
graphics_SanAngeles::graphics_SanAngeles, 106
hardware_KeyboardAssembly::hardware_-KeyboardAssembly, 122
hardware_MemoryThroughput::hardware_-MemoryThroughput, 125

hardware_Touchpad::hardware_Touchpad, 139
power_Resume::power_Resume, 176

realtimecomm_GTalkAudioPlayground, 54
SLEEP_DURATION, 55
WARMUP_TIME, 55
realtimecomm_GTalkAudioPlayground::realtimecomm_GTalkAudioPlayground, 177
dep, 178
dep_dir, 178
performance_results, 179
playground, 179
run_once, 177
run_verification, 178
setup, 178
version, 179
realtimecomm_GTalkPlayground, 56
SLEEP_DURATION, 57
WARMUP_TIME, 57
realtimecomm_GTalkPlayground::realtimecomm_GTalkPlayground, 180
dep, 182
dep_dir, 182
get_framerate, 180
performance_results, 182
playground, 182
run_once, 181
run_verification, 182
setup, 182
version, 183
RECONFIG_PATH
hardware_VideoOutSemiAuto::hardware_VideoOutSemiAuto, 142
Release
hardware_BluetoothSemiAuto::Agent, 59
remaining_time
power_BatteryCharge::power_BatteryCharge, 160
report_data
unixbench::unixbench, 187
RequestPinCode
hardware_BluetoothSemiAuto::Agent, 60
restore_state
power_CPUFreq::cpufreq, 75
results
firmware_VbootCrypto::firmware_VbootCrypto, 96
gl_Bench::gl_Bench, 99
hardware_MemoryThroughput::hardware_MemoryThroughput, 125
platform_AesThroughput::platform_AesThroughput, 153
run_benchmarks
firmware_VbootCrypto::firmware_VbootCrypto, 95
run_crypto
firmware_VbootCrypto::firmware_VbootCrypto, 95
run_once
audiovideo_FFMPEG::audiovideo_FFMPEG, 61
audiovideo_V4L2::audiovideo_V4L2, 65
compilebench::compilebench, 71
desktopui_SunSpiderBench::desktopui_SunSpiderBench, 83
desktopui_V8Bench::desktopui_V8Bench, 85
firmware_RomSize::firmware_RomSize, 90
firmware_VbootCrypto::firmware_VbootCrypto, 95
gl_Bench::gl_Bench, 98
graphics_GLAPICheck::graphics_GLAPICheck, 103
graphics_SanAngeles::graphics_SanAngeles, 105
graphics_TearTest::graphics_TearTest, 107
hardware_Backlight::hardware_Backlight, 109
hardware_BluetoothSemiAuto::hardware_BluetoothSemiAuto, 112
hardware_Components::hardware_Components, 118
hardware_DiskSize::hardware_DiskSize, 121
hardware_KeyboardAssembly::hardware_KeyboardAssembly, 122
hardware_MemoryThroughput::hardware_MemoryThroughput, 124
hardware_MemoryTotalSize::hardware_MemoryTotalSize, 126
hardware_Resolution::hardware_Resolution, 128
hardware_SAT::hardware_SAT, 129
hardware_SsdDetection::hardware_SsdDetection, 131
hardware_StorageFio::hardware_StorageFio, 136
hardware_Touchpad::hardware_Touchpad, 139
hardware_VideoOutSemiAuto::hardware_VideoOutSemiAuto, 142
ltp::ltp, 145
network_DisableInterface::network_DisableInterface, 147
network_WiFiCaps::network_WiFiCaps, 149
platform_AesThroughput::platform_AesThroughput, 152
platform_BootPerf::platform_BootPerf, 156
platform_KernelVersion::platform_KernelVersion, 157

power_BatteryCharge::power_BatteryCharge, 159
power_CPUFreq::power_CPUFreq, 162
power_CPUIdle::power_CPUIdle, 164
power_Draw::power_Draw, 165
power_LoadTest::power_LoadTest, 169
power_Resume::power_Resume, 175
realtimecomm_GTalkAudioPlayground::realtimecomm_GTalkAudioPlayground, 177
realtimecomm_GTalkPlayground::realtimecomm_GTalkPlayground, 181
unixbench::unixbench, 186
run_rollback
firmware_VbootCrypto::firmware_VbootCrypto, 95
run_splicing
firmware_VbootCrypto::firmware_VbootCrypto, 95
run_testcase
audiovideo_FFMPEG::audiovideo_FFMPEG, 61
run_v4l2_capture_test
audiovideo_V4L2::audiovideo_V4L2, 65
run_v4l2_capture_tests
audiovideo_V4L2::audiovideo_V4L2, 65
run_v4l2_default_capture_test
audiovideo_V4L2::audiovideo_V4L2, 66
run_v4l2_unittests
audiovideo_V4L2::audiovideo_V4L2, 66
run_verification
firmware_VbootCrypto::firmware_VbootCrypto, 96
realtimecomm_GTalkAudioPlayground::realtimecomm_GTalkAudioPlayground, 178
realtimecomm_GTalkPlayground::realtimecomm_GTalkPlayground, 182

save_state
power_CPUFreq::cpufreq, 75
set_frequency
power_CPUFreq::cpufreq, 75
set_governor
power_CPUFreq::cpufreq, 75
setup
audiovideo_FFMPEG::audiovideo_FFMPEG, 63
audiovideo_V4L2::audiovideo_V4L2, 68
compilebench::compilebench, 71
desktopui_SunSpiderBench::desktopui_SunSpiderBench, 84
desktopui_V8Bench::desktopui_V8Bench, 86
disktest::disktest, 88
firmware_VbootCrypto::firmware_VbootCrypto, 96
gl_Bench::gl_Bench, 98
graphics_GLAPICheck::graphics_GLAPICheck, 104
graphics_SanAngeles::graphics_SanAngeles, 105
graphics_TearTest::graphics_TearTest, 108
hardware_MemoryThroughput::hardware_MemoryThroughput, 124
hardware_SAT::hardware_SAT, 129
hardware_SsdDetection::hardware_SsdDetection, 131
hardware_StorageFio::hardware_StorageFio, 137
ltp::ltp, 145
network_WiFiCaps::network_WiFiCaps, 150
platform_AesThroughput::platform_AesThroughput, 152
power_LoadTest::power_LoadTest, 171
realtimecomm_GTalkAudioPlayground::realtimecomm_GTalkAudioPlayground, 178
realtimecomm_GTalkPlayground::realtimecomm_GTalkPlayground, 182
unixbench::unixbench, 186
site_ignore_tests
ltp::ltp, 146
SLEEP_DURATION
realtimecomm_GTalkAudioPlayground, 55
realtimecomm_GTalkPlayground, 57
status
power_BatteryCharge::power_BatteryCharge, 160
support_readwrite
audiovideo_V4L2::audiovideo_V4L2, 68
support_streaming
audiovideo_V4L2::audiovideo_V4L2, 68
supported_controls
audiovideo_V4L2::audiovideo_V4L2, 69
supported_formats
audiovideo_V4L2::audiovideo_V4L2, 69

tarball
compilebench::compilebench, 71
TEMPLATE
graphics_TearTest, 26
test_name
compilebench, 16
test_one_disk_chunk
disktest::disktest, 88

testsource/audiovideo_FFMPEG.py, 189
testsource/audiovideo_V4L2.py, 190
testsource/compilebench.py, 191
testsource/desktopui_SunSpiderBench.py, 192
testsource/desktopui_V8Bench.py, 193
testsource/disktest.py, 194
testsource/firmware_RomSize.py, 195
testsource/firmware_VbootCrypto.py, 196
testsource/gl_Bench.py, 197
testsource/graphics_GLAPICheck.py, 198
testsource/graphics_SanAngeles.py, 199
testsource/graphics_TearTest.py, 200
testsource/hardware_Backlight.py, 201
testsource/hardware_BluetoothSemiAuto.py, 202
testsource/hardware_Components.py, 203
testsource/hardware_DiskSize.py, 204
testsource/hardware_KeyboardAssembly.py, 205
testsource/hardware_MemoryThroughput.py, 206
testsource/hardware_MemoryTotalSize.py, 207
testsource/hardware_Resolution.py, 208
testsource/hardware_SAT.py, 209
testsource/hardware_SsdDetection.py, 210
testsource/hardware_StorageFio.py, 211
testsource/hardware_Touchpad.py, 212
testsource/hardware_VideoOutSemiAuto.py, 213
testsource/ltp.py, 214
testsource/mainpage.txt, 215
testsource/network_DisableInterface.py, 216
testsource/network_WiFiCaps.py, 217
testsource/platform_AesThroughput.py, 218
testsource/platform_BootPerf.py, 219
testsource/platform_KernelVersion.py, 220
testsource/power_BatteryCharge.py, 221
testsource/power_CPUFreq.py, 222
testsource/power_CPUIdle.py, 223
testsource/power_Draw.py, 224
testsource/power_LoadTest.py, 225
testsource/power_Resume.py, 226
testsource/realtimecomm_GTalkAudioPlayground.py, 227
testsource/realtimecomm_GTalkPlayground.py, 228
testsource/unixbench.py, 229

unittest_passed
audiovideo_V4L2::audiovideo_V4L2, 68
unixbench, 58
unixbench::unixbench, 184
check_for_error, 184
cleanup, 184
err, 187
initialize, 185
postprocess_iteration, 185
report_data, 187
run_once, 186
setup, 186
version, 187
update_stats
platform_AesThroughput::platform_AesThroughput, 152

v4l2_devices
audiovideo_V4L2::audiovideo_V4L2, 69
v4l2_major_dev_num
audiovideo_V4L2::audiovideo_V4L2, 69
v4l2_minor_dev_num_max
audiovideo_V4L2::audiovideo_V4L2, 69
v4l2_minor_dev_num_min
audiovideo_V4L2::audiovideo_V4L2, 69
version
audiovideo_FFMPEG::audiovideo_FFMPEG, 63
audiovideo_V4L2::audiovideo_V4L2, 69
compilebench::compilebench, 71
desktopui_SunSpiderBench::desktopui_SunSpiderBench, 84
desktopui_V8Bench::desktopui_V8Bench, 86
disktest::disktest, 89
firmware_RomSize::firmware_RomSize, 90
firmware_VbootCrypto::firmware_VbootCrypto, 96
gl_Bench::gl_Bench, 99
graphics_GLAPICheck::graphics_GLAPICheck, 104
graphics_SanAngeles::graphics_SanAngeles, 106
graphics_TearTest::graphics_TearTest, 108
hardware_Backlight::hardware_Backlight, 109
hardware_BluetoothSemiAuto::hardware_BluetoothSemiAuto, 112
hardware_Components::hardware_Components, 120
hardware_DiskSize::hardware_DiskSize, 121
hardware_KeyboardAssembly::hardware_KeyboardAssembly, 122
hardware_MemoryThroughput::hardware_MemoryThroughput, 125
hardware_MemoryTotalSize::hardware_MemoryTotalSize, 126
hardware_Resolution::hardware_Resolution, 128
hardware_SAT::hardware_SAT, 130
hardware_SsdDetection::hardware_SsdDetection, 132
hardware_StorageFio::hardware_StorageFio, 137
hardware_Touchpad::hardware_Touchpad, 139

hardware_VideoOutSemiAuto::hardware_-
VideoOutSemiAuto, 142
ltp::ltp, 146
network_DisableInterface::network_-
DisableInterface, 148
network_WiFiCaps::network_WiFiCaps, 150
platform_AesThroughput::platform_-
AesThroughput, 153
platform_BootPerf::platform_BootPerf, 156
platform_KernelVersion::platform_-
KernelVersion, 157
power_BatteryCharge::power_BatteryCharge,
160
power_CPUFreq::power_CPUFreq, 163
power_CPUIdle::power_CPUIdle, 164
power_Draw::power_Draw, 165
power_LoadTest::power_LoadTest, 173
power_Resume::power_Resume, 176
realtimecomm_-
GTalkAudioPlayground::realtimecomm_-
GTalkAudioPlayground, 179
realtimecomm_-
GTalkPlayground::realtimecomm_-
GTalkPlayground, 183
unixbench::unixbench, 187
VGA_ID
hardware_VideoOutSemiAuto::hardware_-
VideoOutSemiAuto, 143

WARMUP_TIME
realtimecomm_GTalkAudioPlayground, 55
realtimecomm_GTalkPlayground, 57

XRANDR_PATH
hardware_VideoOutSemiAuto::hardware_-
VideoOutSemiAuto, 143