

# BRIAR Client Documentation

Generated by Doxygen 1.9.1



<b>1 BRIAR Command Line Interface (CLI) and Client</b>	<b>1</b>
<b>2 Namespace Index</b>	<b>3</b>
2.1 Packages	3
<b>3 Hierarchical Index</b>	<b>5</b>
3.1 Class Hierarchy	5
<b>4 Class Index</b>	<b>7</b>
4.1 Class List	7
<b>5 File Index</b>	<b>9</b>
5.1 File List	9
<b>6 Namespace Documentation</b>	<b>11</b>
6.1 briar Namespace Reference	11
6.1.1 Function Documentation	12
6.1.1.1 <code>_wait_forever()</code>	12
6.1.1.2 <code>CLIServe()</code>	12
6.1.1.3 <code>dyn_import()</code>	13
6.1.1.4 <code>get_process_number()</code>	13
6.1.1.5 <code>get_thread_number()</code>	13
6.1.1.6 <code>multiproc_serve()</code>	13
6.1.1.7 <code>parse_ports()</code>	14
6.1.1.8 <code>serve()</code>	14
6.1.2 Variable Documentation	14
6.1.2.1 <code>__version__</code>	14
6.1.2.2 <code>_ONE_DAY</code>	14
6.1.2.3 <code>DEFAULT_MAX_MESSAGE_SIZE</code>	14
6.1.2.4 <code>DEFAULT_PORT</code>	15
6.1.2.5 <code>DEFAULT_SERVE_PORT</code>	15
6.1.2.6 <code>PLATFORM</code>	15
6.2 <code>briar.__main__</code> Namespace Reference	15
6.3 <code>briar.briar_cli</code> Namespace Reference	15
6.3.1 Detailed Description	16
6.3.2 Function Documentation	16
6.3.2.1 <code>briar_command_line()</code>	16
6.3.2.2 <code>briar_database_command_line()</code>	16
6.3.2.3 <code>briar_test_command_line()</code>	17
6.3.2.4 <code>incomplete()</code>	17
6.3.3 Variable Documentation	17
6.3.3.1 <code>COMMANDS</code>	17
6.3.3.2 <code>DATABASE_COMMANDS</code>	18
6.3.3.3 <code>DETECTION_FILE_EXT</code>	18

6.3.3.4	FACE_COUNT	18
6.3.3.5	MATCHES_FILE_EXT	18
6.3.3.6	TEMPLATE_FILE_EXT	18
6.4	briar.briar_client Namespace Reference	18
6.4.1	Detailed Description	19
6.4.2	Function Documentation	19
6.4.2.1	_initialize_worker()	19
6.4.2.2	_shutdown_worker()	20
6.4.3	Variable Documentation	20
6.4.3.1	_client_identifier_singleton	20
6.4.3.2	_worker_channel_singleton	20
6.4.3.3	_worker_port_singleton	20
6.4.3.4	_worker_proccess_position_singleton	20
6.4.3.5	_worker_stub_singleton	20
6.4.3.6	_worker_thread_position_singleton	20
6.5	briar.briar_media Namespace Reference	21
6.5.1	Detailed Description	21
6.5.2	Function Documentation	21
6.5.2.1	briar_media_from_pb2()	21
6.5.2.2	briar_media_to_pb2()	21
6.5.2.3	load_media_from_folder()	21
6.5.2.4	load_media_from_image()	22
6.5.2.5	load_media_from_numpy()	22
6.6	briar.cli Namespace Reference	22
6.6.1	Detailed Description	22
6.7	briar.cli.connection Namespace Reference	23
6.7.1	Function Documentation	23
6.7.1.1	addConnectionOptions()	23
6.7.2	Variable Documentation	23
6.7.2.1	DEFAULT_MAX_ASYNC	23
6.7.2.2	DEFAULT_MAX_MESSAGE_SIZE	23
6.8	briar.cli.database Namespace Reference	24
6.9	briar.cli.database.checkpoint Namespace Reference	24
6.9.1	Function Documentation	24
6.9.1.1	database_checkpoint()	24
6.10	briar.cli.database.checkpoint_subject Namespace Reference	24
6.10.1	Function Documentation	25
6.10.1.1	database_checkpoint_subject()	25
6.10.1.2	parseDatabaseCheckpointSubjectOptions()	25
6.11	briar.cli.database.common Namespace Reference	25
6.11.1	Function Documentation	25
6.11.1.1	db_no_exist()	25

6.12 briar.cli.database.compute_scores Namespace Reference	26
6.12.1 Function Documentation	26
6.12.1.1 addDatabaseComputeScore_options2proto()	26
6.12.1.2 addDatabaseComputeScoreOptions()	26
6.12.1.3 database_compute_verify()	26
6.12.1.4 parseDatabaseComputeScoreOptions()	27
6.13 briar.cli.database.compute_search Namespace Reference	27
6.13.1 Function Documentation	27
6.13.1.1 database_compute_search()	27
6.14 briar.cli.database.create Namespace Reference	28
6.14.1 Function Documentation	28
6.14.1.1 database_create()	28
6.15 briar.cli.database.delete Namespace Reference	28
6.15.1 Function Documentation	28
6.15.1.1 database_delete()	28
6.15.1.2 parseDatabaseDeleteOptions()	29
6.16 briar.cli.database.finalize Namespace Reference	29
6.16.1 Function Documentation	29
6.16.1.1 database_finalize()	29
6.16.1.2 finalizeParseOptions()	29
6.17 briar.cli.database.info Namespace Reference	30
6.17.1 Function Documentation	30
6.17.1.1 database_info()	30
6.17.1.2 parseDatabaseInfoOptions()	30
6.18 briar.cli.database.list Namespace Reference	30
6.18.1 Function Documentation	31
6.18.1.1 database_list()	31
6.18.1.2 parseDatabaseListOptions()	31
6.19 briar.cli.database.list_entries Namespace Reference	31
6.19.1 Function Documentation	31
6.19.1.1 database_list_entries()	31
6.19.1.2 parseDatabaseListEntriesOptions()	32
6.20 briar.cli.database.load Namespace Reference	32
6.20.1 Function Documentation	32
6.20.1.1 database_load()	32
6.21 briar.cli.database.merge Namespace Reference	32
6.21.1 Function Documentation	32
6.21.1.1 database_merge()	33
6.21.1.2 parseDatabaseMergeOptions()	33
6.22 briar.cli.database.move_entry Namespace Reference	33
6.22.1 Function Documentation	33
6.22.1.1 database_move_entry()	33

6.22.1.2 parseDatabaseMoveEntryOptions()	34
6.23 briar.cli.database.refresh Namespace Reference	34
6.23.1 Function Documentation	34
6.23.1.1 database_checkpoint()	34
6.23.1.2 database_refresh()	34
6.23.1.3 parseDatabaseRefreshOptions()	35
6.24 briar.cli.database.rename Namespace Reference	35
6.24.1 Function Documentation	35
6.24.1.1 database_load()	35
6.24.1.2 database_rename()	35
6.24.1.3 parseDatabaseRenameOptions()	36
6.25 briar.cli.database.retrieve Namespace Reference	36
6.25.1 Function Documentation	36
6.25.1.1 database_retrieve()	36
6.25.1.2 parseDatabaseRetrieveOptions()	36
6.26 briar.cli.detect Namespace Reference	37
6.26.1 Function Documentation	37
6.26.1.1 addDetectorOptions()	37
6.26.1.2 addTrackingOptions()	37
6.26.1.3 detect()	38
6.26.1.4 detect_options2proto()	38
6.26.1.5 detectParseOptions()	38
6.26.1.6 detectRequestConstructor()	39
6.26.1.7 get_detection_path()	39
6.26.1.8 isFinalReply()	39
6.26.1.9 save_detections()	39
6.26.1.10 tracking_options2proto()	39
6.26.2 Variable Documentation	40
6.26.2.1 DETECTION_FILE_EXT	40
6.27 briar.cli.enhance Namespace Reference	40
6.27.1 Function Documentation	40
6.27.1.1 addEnhanceOptions()	40
6.27.1.2 enhance()	41
6.27.1.3 enhance_options2proto()	41
6.27.1.4 enhanceParseOptions()	41
6.27.1.5 enhanceRequestConstructor()	41
6.27.1.6 save_Enhancement()	42
6.27.2 Variable Documentation	42
6.27.2.1 ENHANCE_FILE_EXT	42
6.28 briar.cli.enroll Namespace Reference	42
6.28.1 Function Documentation	42
6.28.1.1 addEnrollOptions()	42

6.28.1.2 enroll()	43
6.28.1.3 enroll_options2proto()	43
6.28.1.4 enrollParseOptions()	43
6.28.1.5 enrollRequestConstructor()	44
6.29 briar.cli.extract Namespace Reference	44
6.29.1 Function Documentation	44
6.29.1.1 addExtractOptions()	44
6.29.1.2 extract()	45
6.29.1.3 extract_options2proto()	45
6.29.1.4 extractParseOptions()	45
6.29.1.5 extractRequestConstructor()	46
6.29.1.6 save_extractions()	46
6.29.2 Variable Documentation	46
6.29.2.1 TEMPLATE_FILE_EXT	46
6.30 briar.cli.finalize Namespace Reference	46
6.30.1 Function Documentation	46
6.30.1.1 database_finalize()	47
6.30.1.2 finalizeParseOptions()	47
6.31 briar.cli.media Namespace Reference	47
6.31.1 Function Documentation	47
6.31.1.1 addMediaOptions()	47
6.31.1.2 collect_files()	48
6.31.1.3 hasExtension()	48
6.31.2 Variable Documentation	48
6.31.2.1 DEFAULT_MAX_SIZE	48
6.32 briar.cli.search Namespace Reference	49
6.32.1 Function Documentation	49
6.32.1.1 addSearchOptions()	49
6.32.1.2 search()	49
6.32.1.3 search_options2proto()	50
6.32.1.4 searchParseOptions()	50
6.32.1.5 searchRequestConstructor()	50
6.32.2 Variable Documentation	50
6.32.2.1 MATCHES_FILE_EXT	50
6.33 briar.cli.sigset Namespace Reference	51
6.33.1 Function Documentation	51
6.33.1.1 checkpoint_subject_threaded()	51
6.33.1.2 df_batch_consumer()	51
6.33.1.3 df_batch_generator()	52
6.33.1.4 df_batch_producer()	52
6.33.1.5 df_row_generator()	53
6.33.1.6 enroll_call_threaded()	53

6.33.1.7 inner_pool_mapper()	53
6.33.1.8 parseSigsetEnrollOptions()	54
6.33.1.9 parseSigsetStatsOptions()	54
6.33.1.10 progress_consumer()	54
6.33.1.11 sigset_enroll()	55
6.33.1.12 sigset_stats()	55
6.33.2 Variable Documentation	55
6.33.2.1 my_pool	55
6.33.2.2 proc_number	55
6.33.2.3 service_address_number	55
6.34 briar.cli.status Namespace Reference	56
6.34.1 Function Documentation	56
6.34.1.1 get_service_configuration()	56
6.34.1.2 print_service_configuration()	56
6.34.1.3 status()	56
6.34.1.4 statusParseOptions()	57
6.35 briar.cli.test Namespace Reference	57
6.35.1 Function Documentation	57
6.35.1.1 detection_output_tests()	57
6.35.1.2 extraction_output_tests()	58
6.35.2 Variable Documentation	58
6.35.2.1 use_colorama	58
6.36 briar.cli.track Namespace Reference	58
6.36.1 Function Documentation	58
6.36.1.1 get_tracklet_path()	58
6.36.1.2 save_tracklets()	59
6.36.1.3 track()	59
6.36.1.4 trackRequestConstructor()	59
6.36.2 Variable Documentation	59
6.36.2.1 TRACKLET_FILE_EXT	59
6.37 briar.cli.verify Namespace Reference	60
6.37.1 Function Documentation	60
6.37.1.1 addVerifyOptions()	60
6.37.1.2 save_verifications()	60
6.37.1.3 verify()	61
6.37.1.4 verify_options2proto()	61
6.37.1.5 verifyParseOptions()	61
6.37.2 Variable Documentation	61
6.37.2.1 VERIFICATION_FILE_EXT	61
6.38 briar.cli.viz Namespace Reference	62
6.38.1 Function Documentation	62
6.38.1.1 viz()	62



6.38.1.2 vizParseOptions()	62
6.39 briar.evaluation Namespace Reference	62
6.39.1 Function Documentation	63
6.39.1.1 runStages()	63
6.40 briar.evaluation.full_evaluation Namespace Reference	63
6.40.1 Function Documentation	65
6.40.1.1 compute_search()	65
6.40.1.2 compute_verify()	65
6.40.1.3 get_info()	65
6.40.1.4 get_multi_info()	66
6.40.1.5 merge_dbs()	66
6.40.1.6 run_on_multi()	66
6.40.1.7 runall()	66
6.40.1.8 setUpClass_main()	66
6.40.1.9 setUpModule()	66
6.40.2 Variable Documentation	66
6.40.2.1 args_string	67
6.40.2.2 blended_gallery_1_filename	67
6.40.2.3 blended_gallery_2_filename	67
6.40.2.4 category	67
6.40.2.5 database_blended_gallery_1_name	67
6.40.2.6 database_blended_gallery_2_name	67
6.40.2.7 database_gallery_1_name	67
6.40.2.8 database_gallery_2_name	67
6.40.2.9 database_multi_probe_name	68
6.40.2.10 database_probe_name	68
6.40.2.11 DATABASE_SUFFIX_FLAG	68
6.40.2.12 DATASET_DIR	68
6.40.2.13 enroll_args	68
6.40.2.14 EVAL_PHASE	68
6.40.2.15 EVALUATION_DIR	68
6.40.2.16 EVALUATION_MULTISUBJECT_DIR	68
6.40.2.17 gallery1_blended_sigset	69
6.40.2.18 gallery1_sigset	69
6.40.2.19 gallery2_blended_sigset	69
6.40.2.20 gallery2_sigset	69
6.40.2.21 gallery_1_blended_sigset_path	69
6.40.2.22 gallery_1_filename	69
6.40.2.23 gallery_1_sigset_path	69
6.40.2.24 gallery_2_blended_sigset_path	69
6.40.2.25 gallery_2_filename	70
6.40.2.26 gallery_2_sigset_path	70

6.40.2.27 generate_report . . . . .	70
6.40.2.28 media_args . . . . .	70
6.40.2.29 module . . . . .	70
6.40.2.30 multisubject_probe_filename . . . . .	70
6.40.2.31 number_of_partitions . . . . .	70
6.40.2.32 OUTPUT_DIR . . . . .	70
6.40.2.33 port_list . . . . .	71
6.40.2.34 probe_filename . . . . .	71
6.40.2.35 probe_multisubject_sigset . . . . .	71
6.40.2.36 probe_multisubject_sigset_path . . . . .	71
6.40.2.37 probe_sigset . . . . .	71
6.40.2.38 probe_sigset_path . . . . .	71
6.40.2.39 requires_database_merge . . . . .	71
6.40.2.40 run_multisubject_evaluation . . . . .	71
6.40.2.41 RUN_STAGES . . . . .	72
6.40.2.42 sortTestMethodsUsing . . . . .	72
6.40.2.43 stages_temp . . . . .	72
6.40.2.44 USE_SINGLE_SUBJECT . . . . .	72
6.40.2.45 USES_FRONTEND_MERGING . . . . .	72
6.41 briar.evaluation.stage1_probe_enroll Namespace Reference . . . . .	72
6.41.1 Variable Documentation . . . . .	72
6.41.1.1 generate_report . . . . .	72
6.41.1.2 main . . . . .	73
6.42 briar.evaluation.stage2 Namespace Reference . . . . .	73
6.42.1 Variable Documentation . . . . .	73
6.42.1.1 generate_report . . . . .	73
6.42.1.2 main . . . . .	73
6.43 briar.evaluation.stage3_result_scoring Namespace Reference . . . . .	73
6.43.1 Variable Documentation . . . . .	73
6.43.1.1 generate_report . . . . .	73
6.43.1.2 main . . . . .	74
6.44 briar.grpc_json Namespace Reference . . . . .	74
6.44.1 Detailed Description . . . . .	74
6.44.2 Function Documentation . . . . .	74
6.44.2.1 dict_to_proto_obj() . . . . .	74
6.44.2.2 load() . . . . .	75
6.44.2.3 proto_obj_to_dict() . . . . .	75
6.44.2.4 save() . . . . .	76
6.44.3 Variable Documentation . . . . .	76
6.44.3.1 ATTRIB_IGNORE . . . . .	76
6.45 briar.media Namespace Reference . . . . .	76
6.45.1 Function Documentation . . . . .	77

6.45.1.1 aenumerate()	77
6.45.1.2 decodeMedia()	77
6.45.1.3 enroll_frames_iter()	78
6.45.1.4 enroll_frames_iter_async()	79
6.45.1.5 file_iter()	80
6.45.1.6 frame_iter()	81
6.45.1.7 ImageGenerator()	81
6.45.1.8 isFinalFrame()	82
6.45.1.9 single_frame_generate()	82
6.46 briar.media.VideoStream Namespace Reference	82
6.47 briar.media.visualize Namespace Reference	83
6.47.1 Function Documentation	83
6.47.1.1 decode_track()	83
6.47.1.2 get_frame()	83
6.47.1.3 playVideo()	84
6.47.1.4 update_annot()	84
6.47.1.5 update_annot_filename_only()	84
6.47.1.6 visualize_detection()	84
6.47.1.7 visualize_matches()	84
6.47.1.8 visualize_track()	84
6.47.1.9 windowclick()	85
6.47.1.10 windowhover()	85
6.47.1.11 windowhover_filename_only()	85
6.47.2 Variable Documentation	85
6.47.2.1 fdir	85
6.47.2.2 files	85
6.48 briar.media_converters Namespace Reference	85
6.48.1 Detailed Description	86
6.48.2 Function Documentation	86
6.48.2.1 attribute_find()	87
6.48.2.2 attribute_proto2val()	87
6.48.2.3 attribute_retrieve()	87
6.48.2.4 attribute_val2proto()	88
6.48.2.5 check_if_delete_request()	88
6.48.2.6 check_if_delete_request_due_to_error()	88
6.48.2.7 get_entry_id_list()	89
6.48.2.8 image_cv2proto()	89
6.48.2.9 image_file2proto()	89
6.48.2.10 image_np2proto()	89
6.48.2.11 image_proto2cv()	91
6.48.2.12 image_proto2np()	91
6.48.2.13 matrix_np2proto()	92

6.48.2.14 matrix_proto2np()	92
6.48.2.15 modality_proto2string()	92
6.48.2.16 modality_string2proto()	93
6.48.2.17 pathmap_path2remotepath()	93
6.48.2.18 pathmap_str2dict()	93
6.48.2.19 subjectID_int2str()	94
6.48.2.20 subjectID_str2int()	94
6.48.2.21 subjectList_list2string()	94
6.48.2.22 subjectList_string2list()	95
6.48.2.23 tracklet_list2proto()	95
6.48.2.24 vector_np2proto()	95
6.48.2.25 vector_proto2np()	96
6.48.2.26 video_file2proto()	96
6.48.3 Variable Documentation	96
6.48.3.1 attribute_type_name_map	96
6.48.3.2 modalityDict	97
6.48.3.3 reverseModalityDict	97
6.49 briar.sigset Namespace Reference	97
6.50 briar.sigset.parse Namespace Reference	97
6.50.1 Function Documentation	97
6.50.1.1 create_test_sigset()	97
6.50.1.2 expandTree()	98
6.50.1.3 parseBriarSigset()	98
6.50.2 Variable Documentation	98
6.50.2.1 args	98
6.51 briar.timing Namespace Reference	98
6.51.1 Function Documentation	99
6.51.1.1 end_duration()	99
6.51.1.2 generate_progress()	99
6.51.1.3 loadDurationsFolder()	99
6.51.1.4 parseDurations()	99
6.51.1.5 print_duration()	99
6.51.1.6 print_durations()	100
6.51.1.7 save_durations()	100
6.51.1.8 start_duration()	100
6.51.1.9 timeElapsed()	100
6.51.1.10 timestamp()	100
6.51.2 Variable Documentation	100
6.51.2.1 DURATION_FILE_EXT	100
<b>7 Class Documentation</b>	<b>101</b>
7.1 BriarClient Class Reference	101

7.1.1 Detailed Description . . . . .	103
7.1.2 Constructor & Destructor Documentation . . . . .	103
7.1.2.1 <code>__init__()</code> . . . . .	103
7.1.3 Member Function Documentation . . . . .	104
7.1.3.1 <code>database_create()</code> . . . . .	104
7.1.3.2 <code>database_insert()</code> . . . . .	104
7.1.3.3 <code>database_list_templates()</code> . . . . .	105
7.1.3.4 <code>database_refresh()</code> . . . . .	105
7.1.3.5 <code>database_remove_templates()</code> . . . . .	105
7.1.3.6 <code>database_retrieve()</code> . . . . .	106
7.1.3.7 <code>detect()</code> . . . . .	106
7.1.3.8 <code>enhance()</code> . . . . .	106
7.1.3.9 <code>enroll()</code> . . . . .	107
7.1.3.10 <code>enroll_frames_iter()</code> . . . . .	107
7.1.3.11 <code>enroll_frames_iter_async()</code> . . . . .	108
7.1.3.12 <code>extract()</code> . . . . .	108
7.1.3.13 <code>finalize()</code> . . . . .	108
7.1.3.14 <code>get_database_names()</code> . . . . .	109
7.1.3.15 <code>get_service_configuration()</code> . . . . .	109
7.1.3.16 <code>get_status()</code> . . . . .	109
7.1.3.17 <code>iter_over_async()</code> . . . . .	109
7.1.3.18 <code>load_database()</code> . . . . .	110
7.1.3.19 <code>print_verbose()</code> . . . . .	110
7.1.3.20 <code>retrieve_req_iter()</code> . . . . .	110
7.1.3.21 <code>search()</code> . . . . .	111
7.1.3.22 <code>sync_enroll_frames_iter()</code> . . . . .	111
7.1.3.23 <code>track()</code> . . . . .	112
7.1.3.24 <code>verify()</code> . . . . .	112
7.1.3.25 <code>verify_file_iter()</code> . . . . .	113
7.1.3.26 <code>verify_files()</code> . . . . .	114
7.1.4 Member Data Documentation . . . . .	115
7.1.4.1 <code>channel</code> . . . . .	115
7.1.4.2 <code>DEFAULT_PORT</code> . . . . .	115
7.1.4.3 <code>options</code> . . . . .	116
7.1.4.4 <code>port</code> . . . . .	116
7.1.4.5 <code>stub</code> . . . . .	116
7.2 BriarMedia Class Reference . . . . .	116
7.2.1 Constructor & Destructor Documentation . . . . .	116
7.2.1.1 <code>__init__()</code> . . . . .	117
7.2.2 Member Data Documentation . . . . .	117
7.2.2.1 <code>channels</code> . . . . .	117
7.2.2.2 <code>DATA_TYPES</code> . . . . .	117

7.2.2.3 datetime	117
7.2.2.4 description	117
7.2.2.5 fps	117
7.2.2.6 height	118
7.2.2.7 IMAGE_FORMATS	118
7.2.2.8 len	118
7.2.2.9 metadata	118
7.2.2.10 source	118
7.2.2.11 VIDEO_FORMATS	118
7.2.2.12 width	118
7.3 BriarProgress Class Reference	119
7.3.1 Constructor & Destructor Documentation	119
7.3.1.1 __init__()	119
7.3.2 Member Function Documentation	119
7.3.2.1 close()	120
7.3.2.2 refresh()	120
7.3.2.3 update()	120
7.3.3 Member Data Documentation	120
7.3.3.1 desc	120
7.3.3.2 enabled	120
7.3.3.3 leave	120
7.3.3.4 name	121
7.3.3.5 pbar	121
7.3.3.6 position	121
7.3.3.7 prevstep	121
7.3.3.8 tqdm	121
7.4 BriarTest Class Reference	121
7.4.1 Constructor & Destructor Documentation	122
7.4.1.1 __init__()	122
7.4.2 Member Function Documentation	122
7.4.2.1 description()	122
7.4.2.2 run()	122
7.4.2.3 test()	122
7.5 BriarResult Class Reference	123
7.5.1 Constructor & Destructor Documentation	123
7.5.1.1 __init__()	123
7.5.2 Member Data Documentation	123
7.5.2.1 level	123
7.5.2.2 name	123
7.5.2.3 passed	123
7.5.2.4 reason	124
7.6 BriarVideoIterator Class Reference	124

7.6.1 Constructor & Destructor Documentation	124
7.6.1.1 <code>__init__()</code>	125
7.6.2 Member Function Documentation	125
7.6.2.1 <code>__aiter__()</code>	125
7.6.2.2 <code>__anext__()</code>	125
7.6.2.3 <code>__iter__()</code>	126
7.6.2.4 <code>__len__()</code>	126
7.6.2.5 <code>__next__()</code>	126
7.7 DatabaseTest Class Reference	127
7.7.1 Member Function Documentation	127
7.7.1.1 <code>test()</code>	127
7.8 DetectTest Class Reference	128
7.8.1 Member Function Documentation	129
7.8.1.1 <code>description()</code>	129
7.8.1.2 <code>test_1_detection_image()</code>	129
7.8.1.3 <code>test_2_detection_image_output()</code>	129
7.8.1.4 <code>test_3_detection_image_withreturn()</code>	129
7.8.1.5 <code>test_4_detection_image_output_withreturn()</code>	129
7.8.2 Member Data Documentation	129
7.8.2.1 <code>detection_file_path</code>	130
7.8.2.2 <code>output_path</code>	130
7.8.2.3 <code>testim_path</code>	130
7.9 EnrollTest Class Reference	130
7.9.1 Member Function Documentation	131
7.9.1.1 <code>test()</code>	131
7.10 ExtractTest Class Reference	131
7.10.1 Member Function Documentation	132
7.10.1.1 <code>description()</code>	132
7.10.1.2 <code>test_1_extraction_image()</code>	132
7.10.1.3 <code>test_2_extraction_image_output()</code>	132
7.10.2 Member Data Documentation	133
7.10.2.1 <code>detection_file_path</code>	133
7.10.2.2 <code>output_path</code>	133
7.10.2.3 <code>template_file_path</code>	133
7.10.2.4 <code>testim_path</code>	133
7.11 FileVideoStream_cv2 Class Reference	133
7.11.1 Constructor & Destructor Documentation	134
7.11.1.1 <code>__init__()</code>	134
7.11.2 Member Function Documentation	134
7.11.2.1 <code>get_fps()</code>	134
7.11.2.2 <code>get_height()</code>	134
7.11.2.3 <code>get_length()</code>	134

7.11.2.4 <code>get_position()</code>	135
7.11.2.5 <code>get_width()</code>	135
7.11.2.6 <code>is_open()</code>	135
7.11.2.7 <code>more()</code>	135
7.11.2.8 <code>read()</code>	135
7.11.2.9 <code>running()</code>	135
7.11.2.10 <code>scrub_to()</code>	135
7.11.2.11 <code>start()</code>	136
7.11.2.12 <code>stop()</code>	136
7.11.2.13 <code>update()</code>	136
7.11.3 Member Data Documentation	136
7.11.3.1 <code>options</code>	136
7.11.3.2 <code>Q</code>	136
7.11.3.3 <code>stopped</code>	136
7.11.3.4 <code>stream</code>	136
7.11.3.5 <code>thread</code>	137
7.11.3.6 <code>transform</code>	137
7.12 FileVideoStream_imageio Class Reference	137
7.12.1 Constructor & Destructor Documentation	137
7.12.1.1 <code>__init__()</code>	138
7.12.2 Member Function Documentation	138
7.12.2.1 <code>get_fps()</code>	138
7.12.2.2 <code>get_height()</code>	138
7.12.2.3 <code>get_length()</code>	138
7.12.2.4 <code>get_position()</code>	138
7.12.2.5 <code>get_width()</code>	138
7.12.2.6 <code>is_open()</code>	139
7.12.2.7 <code>more()</code>	139
7.12.2.8 <code>read()</code>	139
7.12.2.9 <code>running()</code>	139
7.12.2.10 <code>scrub_to()</code>	139
7.12.2.11 <code>start()</code>	139
7.12.2.12 <code>stop()</code>	139
7.12.2.13 <code>update()</code>	140
7.12.3 Member Data Documentation	140
7.12.3.1 <code>backend</code>	140
7.12.3.2 <code>fps</code>	140
7.12.3.3 <code>options</code>	140
7.12.3.4 <code>Q</code>	140
7.12.3.5 <code>stopped</code>	140
7.12.3.6 <code>stream</code>	140
7.12.3.7 <code>thread</code>	141



7.12.3.8 transform	141
7.13 Fore Class Reference	141
7.13.1 Member Data Documentation	141
7.13.1.1 BLUE	141
7.13.1.2 GREEN	141
7.13.1.3 RED	141
7.13.1.4 RESET	142
7.13.1.5 YELLOW	142
7.14 GrpcDecoder Class Reference	142
7.14.1 Detailed Description	143
7.14.2 Constructor & Destructor Documentation	143
7.14.2.1 __init__()	143
7.14.3 Member Function Documentation	143
7.14.3.1 default()	143
7.14.4 Member Data Documentation	144
7.14.4.1 options	144
7.15 GrpcEncoder Class Reference	144
7.15.1 Detailed Description	145
7.15.2 Constructor & Destructor Documentation	145
7.15.2.1 __init__()	145
7.15.3 Member Function Documentation	145
7.15.3.1 default()	145
7.15.4 Member Data Documentation	146
7.15.4.1 options	146
7.16 ImageIterator Class Reference	146
7.16.1 Constructor & Destructor Documentation	147
7.16.1.1 __init__()	147
7.16.2 Member Function Documentation	148
7.16.2.1 __iter__()	148
7.16.2.2 __len__()	148
7.16.2.3 __next__()	148
7.16.3 Member Data Documentation	148
7.16.3.1 debug_empty	149
7.16.3.2 filepath	149
7.16.3.3 fps	149
7.16.3.4 frame	149
7.16.3.5 frame_count	149
7.16.3.6 frame_height	149
7.16.3.7 frame_width	149
7.16.3.8 i	149
7.16.3.9 isOpened	150
7.16.3.10 length	150

7.16.3.11 msec	150
7.16.3.12 pos	150
7.16.3.13 processed	150
7.16.3.14 start_frame	150
7.16.3.15 stop_frame	150
7.17 match_matrix_visualizer Class Reference	151
7.17.1 Constructor & Destructor Documentation	151
7.17.1.1 __init__()	151
7.17.2 Member Function Documentation	151
7.17.2.1 showmat_interactive()	151
7.17.3 Member Data Documentation	152
7.17.3.1 annotations	152
7.17.3.2 ax	152
7.17.3.3 fig	152
7.17.3.4 figures	152
7.17.3.5 gallerydb_name	152
7.17.3.6 gt	152
7.17.3.7 mat	152
7.17.3.8 prevx	153
7.17.3.9 prevy	153
7.17.3.10 probedb_name	153
7.17.3.11 searchReply	153
7.17.3.12 xlabs	153
7.17.3.13 xsources	153
7.17.3.14 ylabs	153
7.17.3.15 ysources	154
7.18 MediaSetIterator Class Reference	154
7.18.1 Constructor & Destructor Documentation	155
7.18.1.1 __init__()	155
7.18.2 Member Function Documentation	155
7.18.2.1 __iter__()	155
7.18.2.2 __len__()	156
7.18.2.3 __next__()	156
7.18.3 Member Data Documentation	156
7.18.3.1 filepaths	156
7.18.3.2 i	156
7.18.3.3 isOpened	156
7.18.3.4 media_set	157
7.18.3.5 processed	157
7.18.3.6 start_frame	157
7.18.3.7 start_frames	157
7.18.3.8 stop_frame	157

7.18.3.9 stop_frames . . . . .	157
7.19 MediaStream Class Reference . . . . .	157
7.19.1 Constructor & Destructor Documentation . . . . .	158
7.19.1.1 __init__() . . . . .	158
7.19.2 Member Function Documentation . . . . .	158
7.19.2.1 __iter__() . . . . .	158
7.19.3 Member Data Documentation . . . . .	158
7.19.3.1 _media_list . . . . .	158
7.20 Rect Class Reference . . . . .	158
7.20.1 Detailed Description . . . . .	159
7.20.2 Constructor & Destructor Documentation . . . . .	159
7.20.2.1 __init__() . . . . .	159
7.20.3 Member Data Documentation . . . . .	159
7.20.3.1 height . . . . .	159
7.20.3.2 width . . . . .	159
7.20.3.3 x . . . . .	160
7.20.3.4 y . . . . .	160
7.21 Test000InitialConfig Class Reference . . . . .	160
7.21.1 Member Function Documentation . . . . .	161
7.21.1.1 setUpClass() . . . . .	161
7.21.1.2 test_01_config_portlist() . . . . .	161
7.21.1.3 test_02_port_connections() . . . . .	161
7.21.1.4 test_03_num_service_ports() . . . . .	162
7.21.1.5 test_04_num_procs_per_port() . . . . .	162
7.21.1.6 test_05_num_threads_per_port() . . . . .	162
7.21.1.7 test_06_correct_database_creation() . . . . .	162
7.21.1.8 testDatasetDir() . . . . .	162
7.21.1.9 testOutDir() . . . . .	162
7.21.1.10 testValidationDir() . . . . .	163
7.21.2 Member Data Documentation . . . . .	163
7.21.2.1 config_reply . . . . .	163
7.22 Test001SigsetEnrollProbe Class Reference . . . . .	163
7.22.1 Member Function Documentation . . . . .	164
7.22.1.1 setUpClass() . . . . .	164
7.22.1.2 test_01_sigset_enroll_probe() . . . . .	164
7.22.1.3 test_02_probe_checkpoint() . . . . .	164
7.23 Test002ProbeDatabaseMerge Class Reference . . . . .	164
7.23.1 Member Function Documentation . . . . .	165
7.23.1.1 setUpClass() . . . . .	165
7.23.1.2 test_02_merge_probe_dbs() . . . . .	165
7.23.1.3 test_03_checkpoint_merged_probe_db() . . . . .	166
7.23.2 Member Data Documentation . . . . .	166

7.23.2.1 merged_dbinfo . . . . .	166
7.23.2.2 total_entries . . . . .	166
7.23.2.3 total_failed . . . . .	166
7.23.2.4 total_templates . . . . .	166
7.24 Test003SigsetEnrollMultiProbe Class Reference . . . . .	167
7.24.1 Member Function Documentation . . . . .	167
7.24.1.1 setUpClass() . . . . .	167
7.24.1.2 test_01_sigset_enroll_probe() . . . . .	168
7.24.1.3 test_02_probe_checkpoint() . . . . .	168
7.25 Test004MultiProbeDatabaseMerge Class Reference . . . . .	168
7.25.1 Member Function Documentation . . . . .	169
7.25.1.1 setUpClass() . . . . .	169
7.25.1.2 test_02_merge_probe_dbs() . . . . .	169
7.25.1.3 test_03_checkpoint_merged_probe_db() . . . . .	169
7.25.2 Member Data Documentation . . . . .	169
7.25.2.1 merged_dbinfo . . . . .	169
7.25.2.2 total_entries . . . . .	170
7.25.2.3 total_failed . . . . .	170
7.25.2.4 total_templates . . . . .	170
7.26 Test005SigsetEnrollGalleries Class Reference . . . . .	170
7.26.1 Member Function Documentation . . . . .	171
7.26.1.1 runGallery() . . . . .	171
7.26.1.2 setUpClass() . . . . .	171
7.26.1.3 test_01_sigset_enroll_gallery1() . . . . .	171
7.26.1.4 test_02_gallery1_partitioned_checkpoint() . . . . .	171
7.26.1.5 test_03_sigset_enroll_gallery2() . . . . .	172
7.26.1.6 test_04_gallery2_partitioned_checkpoint() . . . . .	172
7.26.1.7 test_05_sigset_enroll_gallery1() . . . . .	172
7.26.1.8 test_06_gallery1_partitioned_checkpoint() . . . . .	172
7.26.1.9 test_07_sigset_enroll_gallery2() . . . . .	172
7.26.1.10 test_08_gallery2_partitioned_checkpoint() . . . . .	172
7.27 Test006GalleryDatabaseMerge Class Reference . . . . .	173
7.27.1 Member Function Documentation . . . . .	173
7.27.1.1 merge_db_func() . . . . .	174
7.27.1.2 setUpClass() . . . . .	174
7.27.1.3 test_01_merge_gallery1_dbs() . . . . .	174
7.27.1.4 test_02_merge_gallery2_dbs() . . . . .	174
7.27.1.5 test_03_merge_blended_gallery2_dbs() . . . . .	174
7.27.1.6 test_04_merge_blended_gallery2_dbs() . . . . .	174
7.27.1.7 test_05_finalized_merged_gallery1_db() . . . . .	174
7.27.1.8 test_06_finalized_merged_gallery1_db() . . . . .	175
7.27.1.9 test_07_finalized_merged_gallery1_db() . . . . .	175

7.27.1.10 test_08_finalized_merged_gallery1_db()	175
7.28 Test007SigsetScoreG1 Class Reference	175
7.28.1 Member Function Documentation	176
7.28.1.1 setUpClass()	176
7.28.1.2 test_01_sigset_verify_probe_gallery1()	176
7.28.1.3 test_02_sigset_search_probe_gallery1()	176
7.29 Test008SigsetScoreGaitG1 Class Reference	176
7.29.1 Member Function Documentation	177
7.29.1.1 setUpClass()	177
7.29.1.2 test_01_sigset_verify_gait_probe_gallery1()	177
7.29.1.3 test_02_sigset_search_gait_probe_gallery1()	177
7.30 Test009SigsetScoreFaceG1 Class Reference	178
7.30.1 Member Function Documentation	178
7.30.1.1 setUpClass()	178
7.30.1.2 test_01_sigset_verify_face_probe_gallery1()	179
7.30.1.3 test_02_sigset_search_face_probe_gallery1()	179
7.31 Test010SigsetScoreWholeBodyG1 Class Reference	179
7.31.1 Member Function Documentation	180
7.31.1.1 setUpClass()	180
7.31.1.2 test_01_sigset_verify_WB_probe_gallery1()	180
7.31.1.3 test_02_sigset_search_WB_probe_gallery1()	180
7.32 Test011SigsetScoreG2 Class Reference	180
7.32.1 Member Function Documentation	181
7.32.1.1 setUpClass()	181
7.32.1.2 test_01_sigset_verify_probe_gallery2()	181
7.32.1.3 test_02_sigset_search_probe_gallery2()	181
7.33 Test012SigsetScoreGaitG2 Class Reference	182
7.33.1 Member Function Documentation	182
7.33.1.1 setUpClass()	182
7.33.1.2 test_01_sigset_verify_gait_probe_gallery2()	183
7.33.1.3 test_02_sigset_search_gait_probe_gallery2()	183
7.34 Test013SigsetScoreFaceG2 Class Reference	183
7.34.1 Member Function Documentation	184
7.34.1.1 setUpClass()	184
7.34.1.2 test_01_sigset_verify_face_probe_gallery2()	184
7.34.1.3 test_02_sigset_search_face_probe_gallery2()	184
7.35 Test014SigsetScoreWholeBodyG2 Class Reference	184
7.35.1 Member Function Documentation	185
7.35.1.1 setUpClass()	185
7.35.1.2 test_01_sigset_verify_WB_probe_gallery2()	185
7.35.1.3 test_02_sigset_search_WB_probe_gallery2()	185
7.36 Test015SigsetBlendedScoreG1 Class Reference	186

7.36.1 Member Function Documentation	186
7.36.1.1 setUpClass()	186
7.36.1.2 test_01_sigset_verify_probe_blended_gallery1()	187
7.36.1.3 test_02_sigset_search_probe_blended_gallery1()	187
7.37 Test016SigsetScoreBlendedGaitG1 Class Reference	187
7.37.1 Member Function Documentation	188
7.37.1.1 setUpClass()	188
7.37.1.2 test_01_sigset_verify_gait_probe_blended_gallery1()	188
7.37.1.3 test_02_sigset_search_gait_probe_blended_gallery1()	188
7.38 Test017SigsetScoreBlendedFaceG1 Class Reference	188
7.38.1 Member Function Documentation	189
7.38.1.1 setUpClass()	189
7.38.1.2 test_01_sigset_verify_face_probe_blended_gallery1()	189
7.38.1.3 test_02_sigset_search_face_probe_blended_gallery1()	189
7.39 Test018SigsetScoreBlendedWholeBodyG1 Class Reference	190
7.39.1 Member Function Documentation	190
7.39.1.1 setUpClass()	190
7.39.1.2 test_01_sigset_verify_WB_probe_blended_gallery1()	191
7.39.1.3 test_02_sigset_search_WB_probe_blended_gallery1()	191
7.40 Test019SigsetBlendedScoreG2 Class Reference	191
7.40.1 Member Function Documentation	192
7.40.1.1 setUpClass()	192
7.40.1.2 test_01_sigset_verify_probe_blended_gallery2()	192
7.40.1.3 test_02_sigset_search_probe_blended_gallery2()	192
7.41 Test020SigsetScoreBlendedGaitG2 Class Reference	192
7.41.1 Member Function Documentation	193
7.41.1.1 setUpClass()	193
7.41.1.2 test_01_sigset_verify_gait_probe_blended_gallery2()	193
7.41.1.3 test_02_sigset_search_gait_probe_blended_gallery2()	193
7.42 Test021SigsetScoreBlendedFaceG2 Class Reference	194
7.42.1 Member Function Documentation	194
7.42.1.1 setUpClass()	194
7.42.1.2 test_01_sigset_verify_face_probe_blended_gallery2()	195
7.42.1.3 test_02_sigset_search_face_probe_blended_gallery2()	195
7.43 Test022SigsetScoreBlendedWholeBodyG2 Class Reference	195
7.43.1 Member Function Documentation	196
7.43.1.1 setUpClass()	196
7.43.1.2 test_01_sigset_verify_WB_probe_blended_gallery2()	196
7.43.1.3 test_02_sigset_search_WB_probe_blended_gallery2()	196
7.44 Test023MultiSigsetScoreG1 Class Reference	196
7.44.1 Member Function Documentation	197
7.44.1.1 setUpClass()	197

7.44.1.2 test_01_sigset_verify_multiprobe_gallery1()	197
7.44.1.3 test_02_sigset_search_multiprobe_gallery1()	197
7.45 Test023SigsetSearchOutputFormatting Class Reference	198
7.45.1 Member Function Documentation	199
7.45.1.1 search_file_check()	199
7.45.1.2 test_01_sigset_search_pickle_fileG1()	199
7.45.1.3 test_02_sigset_search_gait_pickle_fileG1()	199
7.45.1.4 test_03_sigset_search_face_pickle_fileG1()	199
7.45.1.5 test_04_sigset_search_wb_pickle_fileG1()	199
7.45.1.6 test_05_sigset_search_pickle_fileG2()	199
7.45.1.7 test_06_sigset_search_gait_pickle_fileG2()	200
7.45.1.8 test_07_sigset_search_face_pickle_fileG2()	200
7.45.1.9 test_08_sigset_search_wb_pickle_fileG2()	200
7.45.1.10 test_09_sigset_blended_search_pickle_fileG1()	200
7.45.1.11 test_10_sigset_blended_search_gait_pickle_fileG1()	200
7.45.1.12 test_11_sigset_blended_search_face_pickle_fileG1()	200
7.45.1.13 test_12_sigset_blended_search_wb_pickle_fileG1()	200
7.45.1.14 test_13_sigset_blended_search_pickle_fileG2()	201
7.45.1.15 test_14_sigset_blended_search_gait_pickle_fileG2()	201
7.45.1.16 test_15_sigset_blended_search_face_pickle_fileG2()	201
7.45.1.17 test_16_sigset_blended_search_wb_pickle_fileG2()	201
7.46 Test024MultiSigsetScoreGaitG1 Class Reference	201
7.46.1 Member Function Documentation	202
7.46.1.1 setUpClass()	202
7.46.1.2 test_01_sigset_verify_gait_multiprobe_gallery1()	202
7.46.1.3 test_02_sigset_search_gait_multiprobe_gallery1()	202
7.47 Test024SigsetVerifyOutputFormatting Class Reference	203
7.47.1 Member Function Documentation	204
7.47.1.1 score_file_check()	204
7.47.1.2 test_01_sigset_verify_pickle_fileG1()	204
7.47.1.3 test_02_sigset_verify_gait_pickle_fileG1()	204
7.47.1.4 test_03_sigset_verify_face_pickle_fileG1()	204
7.47.1.5 test_04_sigset_verify_wb_pickle_fileG1()	204
7.47.1.6 test_05_sigset_verify_pickle_fileG2()	204
7.47.1.7 test_06_sigset_verify_gait_pickle_fileG2()	205
7.47.1.8 test_07_sigset_verify_face_pickle_fileG2()	205
7.47.1.9 test_08_sigset_verify_wb_pickle_fileG2()	205
7.47.1.10 test_09_sigset_blended_verify_pickle_fileG1()	205
7.47.1.11 test_10_sigset_blended_verify_gait_pickle_fileG1()	205
7.47.1.12 test_11_sigset_blended_verify_face_pickle_fileG1()	205
7.47.1.13 test_12_sigset_blended_verify_wb_pickle_fileG1()	205
7.47.1.14 test_13_sigset_blended_verify_pickle_fileG2()	206

7.47.1.15 test_14_sigset_blended_verify_gait_pickle_fileG2()	206
7.47.1.16 test_15_sigset_blended_verify_face_pickle_fileG2()	206
7.47.1.17 test_16_sigset_blended_verify_wb_pickle_fileG2()	206
7.48 Test025MultiSigsetScoreFaceG1 Class Reference	206
7.48.1 Member Function Documentation	207
7.48.1.1 setUpClass()	207
7.48.1.2 test_01_sigset_verify_face_multiprobe_gallery1()	207
7.48.1.3 test_02_sigset_search_face_multiprobe_gallery1()	207
7.49 Test026MultiSigsetScoreWholeBodyG1 Class Reference	208
7.49.1 Member Function Documentation	208
7.49.1.1 setUpClass()	208
7.49.1.2 test_01_sigset_verify_WB_multiprobe_gallery1()	209
7.49.1.3 test_02_sigset_search_WB_multiprobe_gallery1()	209
7.50 Test027MultiSigsetScoreG2 Class Reference	209
7.50.1 Member Function Documentation	210
7.50.1.1 setUpClass()	210
7.50.1.2 test_01_sigset_verify_multiprobe_gallery2()	210
7.50.1.3 test_02_sigset_search_multiprobe_gallery2()	210
7.51 Test028MultiSigsetScoreGaitG2 Class Reference	210
7.51.1 Member Function Documentation	211
7.51.1.1 setUpClass()	211
7.51.1.2 test_01_sigset_verify_gait_multiprobe_gallery2()	211
7.51.1.3 test_02_sigset_search_gait_multiprobe_gallery2()	211
7.52 Test029MultiSigsetScoreFaceG2 Class Reference	212
7.52.1 Member Function Documentation	212
7.52.1.1 setUpClass()	212
7.52.1.2 test_01_sigset_verify_face_multiprobe_gallery2()	213
7.52.1.3 test_02_sigset_search_face_multiprobe_gallery2()	213
7.53 Test030MultiSigsetScoreWholeBodyG2 Class Reference	213
7.53.1 Member Function Documentation	214
7.53.1.1 setUpClass()	214
7.53.1.2 test_01_sigset_verify_WB_multiprobe_gallery2()	214
7.53.1.3 test_02_sigset_search_WB_multiprobe_gallery2()	214
7.54 Test031MultiSigsetBlendedScoreG1 Class Reference	214
7.54.1 Member Function Documentation	215
7.54.1.1 setUpClass()	215
7.54.1.2 test_01_sigset_verify_multiprobe_blended_gallery1()	215
7.54.1.3 test_02_sigset_search_multiprobe_blended_gallery1()	215
7.55 Test032MultiSigsetScoreBlendedGaitG1 Class Reference	216
7.55.1 Member Function Documentation	216
7.55.1.1 setUpClass()	216
7.55.1.2 test_01_sigset_verify_gait_multiprobe_blended_gallery1()	217



7.55.1.3 test_02_sigset_search_gait_multiprobe_blended_gallery1()	217
7.56 Test033MultiSigsetScoreBlendedFaceG1 Class Reference	217
7.56.1 Member Function Documentation	218
7.56.1.1 setUpClass()	218
7.56.1.2 test_01_sigset_verify_face_multiprobe_blended_gallery1()	218
7.56.1.3 test_02_sigset_search_face_multiprobe_blended_gallery1()	218
7.57 Test034MultiSigsetScoreBlendedWholeBodyG1 Class Reference	218
7.57.1 Member Function Documentation	219
7.57.1.1 setUpClass()	219
7.57.1.2 test_01_sigset_verify_WB_multiprobe_blended_gallery1()	219
7.57.1.3 test_02_sigset_search_WB_multiprobe_blended_gallery1()	219
7.58 Test035MultiSigsetBlendedScoreG2 Class Reference	220
7.58.1 Member Function Documentation	220
7.58.1.1 setUpClass()	220
7.58.1.2 test_01_sigset_verify_multiprobe_blended_gallery2()	221
7.58.1.3 test_02_sigset_search_multiprobe_blended_gallery2()	221
7.59 Test036MultiSigsetScoreBlendedGaitG2 Class Reference	221
7.59.1 Member Function Documentation	222
7.59.1.1 setUpClass()	222
7.59.1.2 test_01_sigset_verify_gait_multiprobe_blended_gallery2()	222
7.59.1.3 test_02_sigset_search_gait_multiprobe_blended_gallery2()	222
7.60 Test037MultiSigsetScoreBlendedFaceG2 Class Reference	222
7.60.1 Member Function Documentation	223
7.60.1.1 setUpClass()	223
7.60.1.2 test_01_sigset_verify_face_multiprobe_blended_gallery2()	223
7.60.1.3 test_02_sigset_search_face_multiprobe_blended_gallery2()	223
7.61 Test038MultiSigsetScoreBlendedWholeBodyG2 Class Reference	224
7.61.1 Member Function Documentation	224
7.61.1.1 setUpClass()	224
7.61.1.2 test_01_sigset_verify_WB_multiprobe_blended_gallery2()	225
7.61.1.3 test_02_sigset_search_WB_multiprobe_blended_gallery2()	225
7.62 ThreadedVideoIterator Class Reference	225
7.62.1 Constructor & Destructor Documentation	226
7.62.1.1 __init__()	226
7.62.2 Member Function Documentation	227
7.62.2.1 __aiter__()	227
7.62.2.2 __anext__()	227
7.62.2.3 __iter__()	227
7.62.2.4 __len__()	228
7.62.2.5 __next__()	228
7.62.2.6 stop_iteration()	228
7.62.3 Member Data Documentation	228

7.62.3.1 cap	228
7.62.3.2 debug_empty	229
7.62.3.3 filepath	229
7.62.3.4 fps	229
7.62.3.5 frame_count	229
7.62.3.6 frame_height	229
7.62.3.7 frame_width	229
7.62.3.8 i	229
7.62.3.9 isOpened	229
7.62.3.10 isStarted	230
7.62.3.11 length	230
7.62.3.12 msec	230
7.62.3.13 options	230
7.62.3.14 pos	230
7.62.3.15 processed	230
7.62.3.16 start_frame	230
7.62.3.17 stop_frame	230
7.62.3.18 stream	231
7.63 Videolterator Class Reference	231
7.63.1 Constructor & Destructor Documentation	232
7.63.1.1 __init__()	232
7.63.2 Member Function Documentation	232
7.63.2.1 __aiter__()	233
7.63.2.2 __anext__()	233
7.63.2.3 __iter__()	233
7.63.2.4 __len__()	234
7.63.2.5 __next__()	234
7.63.3 Member Data Documentation	234
7.63.3.1 cap	234
7.63.3.2 debug_empty	234
7.63.3.3 filepath	235
7.63.3.4 fps	235
7.63.3.5 frame_count	235
7.63.3.6 frame_height	235
7.63.3.7 frame_width	235
7.63.3.8 i	235
7.63.3.9 isOpened	235
7.63.3.10 length	235
7.63.3.11 msec	236
7.63.3.12 pos	236
7.63.3.13 processed	236
7.63.3.14 start_frame	236

7.63.3.15 stop_frame . . . . .	236
<b>8 File Documentation</b>	<b>237</b>
8.1 __init__.py File Reference . . . . .	237
8.2 cli/__init__.py File Reference . . . . .	238
8.3 cli/database/__init__.py File Reference . . . . .	238
8.4 evaluation/__init__.py File Reference . . . . .	238
8.5 media/__init__.py File Reference . . . . .	238
8.6 sigset/__init__.py File Reference . . . . .	239
8.7 timing/__init__.py File Reference . . . . .	239
8.8 __main__.py File Reference . . . . .	240
8.9 briar_cli.py File Reference . . . . .	240
8.10 briar_client.py File Reference . . . . .	240
8.11 briar_media.py File Reference . . . . .	241
8.12 cli/connection.py File Reference . . . . .	241
8.13 cli/database/checkpoint.py File Reference . . . . .	242
8.14 cli/database/checkpoint_subject.py File Reference . . . . .	242
8.15 cli/database/common.py File Reference . . . . .	242
8.16 cli/database/compute_scores.py File Reference . . . . .	242
8.17 cli/database/compute_search.py File Reference . . . . .	243
8.18 cli/database/create.py File Reference . . . . .	243
8.19 cli/database/delete.py File Reference . . . . .	243
8.20 cli/database/finalize.py File Reference . . . . .	244
8.21 cli/finalize.py File Reference . . . . .	244
8.22 cli/database/info.py File Reference . . . . .	244
8.23 cli/database/list.py File Reference . . . . .	244
8.24 cli/database/list_entries.py File Reference . . . . .	245
8.25 cli/database/load.py File Reference . . . . .	245
8.26 cli/database/merge.py File Reference . . . . .	245
8.27 cli/database/move_entry.py File Reference . . . . .	246
8.28 cli/database/refresh.py File Reference . . . . .	246
8.29 cli/database/rename.py File Reference . . . . .	246
8.30 cli/database/retrieve.py File Reference . . . . .	246
8.31 cli/detect.py File Reference . . . . .	247
8.32 cli/enhance.py File Reference . . . . .	247
8.33 cli/enroll.py File Reference . . . . .	248
8.34 cli/extract.py File Reference . . . . .	248
8.35 cli/media.py File Reference . . . . .	249
8.36 cli/search.py File Reference . . . . .	249
8.37 cli/sigset.py File Reference . . . . .	249
8.38 cli/status.py File Reference . . . . .	250
8.39 cli/test.py File Reference . . . . .	250

8.40 cli/track.py File Reference . . . . .	251
8.41 cli/verify.py File Reference . . . . .	251
8.42 cli/viz.py File Reference . . . . .	252
8.43 evaluation/full_evaluation.py File Reference . . . . .	252
8.44 evaluation/stage1_probe_enroll.py File Reference . . . . .	254
8.45 evaluation/stage2.1_gallery1_simple_enroll.py File Reference . . . . .	254
8.46 evaluation/stage2.2_gallery2_simple_enroll.py File Reference . . . . .	255
8.47 evaluation/stage2.3_gallery1_blended_enroll.py File Reference . . . . .	255
8.48 evaluation/stage2.4_gallery2_blended_enroll.py File Reference . . . . .	255
8.49 evaluation/stage3_result_scoring.py File Reference . . . . .	255
8.50 grpc_json.py File Reference . . . . .	255
8.51 media/VideoStream.py File Reference . . . . .	256
8.52 media/visualize.py File Reference . . . . .	256
8.53 media_converters.py File Reference . . . . .	257
8.54 readme-cli.md File Reference . . . . .	258
8.55 sigset/parse.py File Reference . . . . .	258
<b>Index</b>	<b>259</b>

# Chapter 1

## BRIAR Command Line Interface (CLI) and Client

### Command Line Interface (CLI)

The command line interface provides a terminal based method of interacting with algorithms developed using the BRIAR API and unifies the commands given to the services into a set of universal commands shared across all projects developed using the API. It is the method of interfacing with any service built using the BRIAR framework as it shares the method calls assigned to the service and provides callable methods which invoke service functions. The client and the associated command line tools will be shared across all algorithms created with BRIAR and should not be modified, ensuring the Evaluation Harness and other sets of tests can be run across algorithms created by different sets of developers and generate comparable sets of results.

Like most command line tools, the command line interface can be scripted to act as part of a larger task. The client in [briar\\_client.py](#) (which the command line interface is an interface for) can also be imported into a python project

### Usage

After running `setup.py`, the briar command line can be run by entering `python -m briar` anywhere. This will print a help statement showing the different functions made available by the command line tool.

Before running any of these functions, however, you will need to first start the provided example service, so it can reply to the commands. This is done by either directly calling the service python file `python service.py` or calling it as a module `python -m briar.service`. This will start the service, and it will run until you forcefully exit it.

You can get the status and version of the example service with `python -m briar status` and you should see the results printed by the client. Attempting to run any of the other functions with the example service in `service.py` will raise a `NotImplementedError`.

### More Details

The stubs and protobuf files which the client uses are detailed more thoroughly in the briar protobuf and stubs documentation



## Chapter 2

# Namespace Index

### 2.1 Packages

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

<a href="#">briar</a>	11
<a href="#">briar.__main__</a>	15
<a href="#">briar.briar_cli</a>	
Created on 2021 at Oak Ridge National Laboratory	15
<a href="#">briar.briar_client</a>	
Copyright 2021 Oak Ridge National Laboratory	18
<a href="#">briar.briar_media</a>	
Defines a media class which acts as a wrapper for image and video files	21
<a href="#">briar.cli</a>	
The modules contained in the CLI package each contain a function (detect, extract, enroll, etc.) which is a command within the broader CLI toolkit along with the assorted helper functions	22
<a href="#">briar.cli.connection</a>	23
<a href="#">briar.cli.database</a>	24
<a href="#">briar.cli.database.checkpoint</a>	24
<a href="#">briar.cli.database.checkpoint_subject</a>	24
<a href="#">briar.cli.database.common</a>	25
<a href="#">briar.cli.database.compute_scores</a>	26
<a href="#">briar.cli.database.compute_search</a>	27
<a href="#">briar.cli.database.create</a>	28
<a href="#">briar.cli.database.delete</a>	28
<a href="#">briar.cli.database.finalize</a>	29
<a href="#">briar.cli.database.info</a>	30
<a href="#">briar.cli.database.list</a>	30
<a href="#">briar.cli.database.list_entries</a>	31
<a href="#">briar.cli.database.load</a>	32
<a href="#">briar.cli.database.merge</a>	32
<a href="#">briar.cli.database.move_entry</a>	33
<a href="#">briar.cli.database.refresh</a>	34
<a href="#">briar.cli.database.rename</a>	35
<a href="#">briar.cli.database.retrieve</a>	36
<a href="#">briar.cli.detect</a>	37
<a href="#">briar.cli.enhance</a>	40
<a href="#">briar.cli.enroll</a>	42
<a href="#">briar.cli.extract</a>	44
<a href="#">briar.cli.finalize</a>	46

<a href="#">briar.cli.media</a>	47
<a href="#">briar.cli.search</a>	49
<a href="#">briar.cli.sigset</a>	51
<a href="#">briar.cli.status</a>	56
<a href="#">briar.cli.test</a>	57
<a href="#">briar.cli.track</a>	58
<a href="#">briar.cli.verify</a>	60
<a href="#">briar.cli.viz</a>	62
<a href="#">briar.evaluation</a>	62
<a href="#">briar.evaluation.full_evaluation</a>	63
<a href="#">briar.evaluation.stage1_probe_enroll</a>	72
<a href="#">briar.evaluation.stage2</a>	73
<a href="#">briar.evaluation.stage3_result_scoring</a>	73
<a href="#">briar.grpc_json</a>	
I couldn't find a good library to write gRPC/protobuf objects to disk so this is my own implementation of a generalized json converter for said objects	74
<a href="#">briar.media</a>	76
<a href="#">briar.media.VideoStream</a>	82
<a href="#">briar.media.visualize</a>	83
<a href="#">briar.media_converters</a>	
Contained in this are functions for converting numpy arrays into various protobuf objects and back again since numpy arrays cannot be sent directly over gRPC	85
<a href="#">briar.sigset</a>	97
<a href="#">briar.sigset.parse</a>	97
<a href="#">briar.timing</a>	98



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

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

BriarMedia . . . . .	116
BriarProgress . . . . .	119
BriarTest . . . . .	121
DatabaseTest . . . . .	127
DetectTest . . . . .	128
EnrollTest . . . . .	130
ExtractTest . . . . .	131
BriarTestResult . . . . .	123
BriarVideoIterator . . . . .	124
ThreadedVideoIterator . . . . .	225
VideoIterator . . . . .	231
FileVideoStream_cv2 . . . . .	133
FileVideoStream_imageio . . . . .	137
Fore . . . . .	141
JSONDecoder	
GrpcDecoder . . . . .	142
JSONEncoder	
GrpcEncoder . . . . .	144
match_matrix_visualizer . . . . .	151
MediaStream . . . . .	157
object	
BriarClient . . . . .	101
ImageIterator . . . . .	146
MediaSetIterator . . . . .	154
Rect . . . . .	158
TestCase	
Test000InitialConfig . . . . .	160
Test001SigsetEnrollProbe . . . . .	163
Test002ProbeDatabaseMerge . . . . .	164
Test003SigsetEnrollMultiProbe . . . . .	167
Test004MultiProbeDatabaseMerge . . . . .	168
Test005SigsetEnrollGalleries . . . . .	170
Test006GalleryDatabaseMerge . . . . .	173
Test007SigsetScoreG1 . . . . .	175
Test008SigsetScoreGaitG1 . . . . .	176

Test009SigsetScoreFaceG1 . . . . .	178
Test010SigsetScoreWholeBodyG1 . . . . .	179
Test011SigsetScoreG2 . . . . .	180
Test012SigsetScoreGaitG2 . . . . .	182
Test013SigsetScoreFaceG2 . . . . .	183
Test014SigsetScoreWholeBodyG2 . . . . .	184
Test015SigsetBlendedScoreG1 . . . . .	186
Test016SigsetScoreBlendedGaitG1 . . . . .	187
Test017SigsetScoreBlendedFaceG1 . . . . .	188
Test018SigsetScoreBlendedWholeBodyG1 . . . . .	190
Test019SigsetBlendedScoreG2 . . . . .	191
Test020SigsetScoreBlendedGaitG2 . . . . .	192
Test021SigsetScoreBlendedFaceG2 . . . . .	194
Test022SigsetScoreBlendedWholeBodyG2 . . . . .	195
Test023MultiSigsetScoreG1 . . . . .	196
Test023SigsetSearchOutputFormatting . . . . .	198
Test024MultiSigsetScoreGaitG1 . . . . .	201
Test024SigsetVerifyOutputFormatting . . . . .	203
Test025MultiSigsetScoreFaceG1 . . . . .	206
Test026MultiSigsetScoreWholeBodyG1 . . . . .	208
Test027MultiSigsetScoreG2 . . . . .	209
Test028MultiSigsetScoreGaitG2 . . . . .	210
Test029MultiSigsetScoreFaceG2 . . . . .	212
Test030MultiSigsetScoreWholeBodyG2 . . . . .	213
Test031MultiSigsetBlendedScoreG1 . . . . .	214
Test032MultiSigsetScoreBlendedGaitG1 . . . . .	216
Test033MultiSigsetScoreBlendedFaceG1 . . . . .	217
Test034MultiSigsetScoreBlendedWholeBodyG1 . . . . .	218
Test035MultiSigsetBlendedScoreG2 . . . . .	220
Test036MultiSigsetScoreBlendedGaitG2 . . . . .	221
Test037MultiSigsetScoreBlendedFaceG2 . . . . .	222
Test038MultiSigsetScoreBlendedWholeBodyG2 . . . . .	224

## Chapter 4

# Class Index

### 4.1 Class List

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

<a href="#">BriarClient</a>	
Provide a client to a BRIAR service	101
<a href="#">BriarMedia</a>	116
<a href="#">BriarProgress</a>	119
<a href="#">BriarTest</a>	121
<a href="#">BriarTestResult</a>	123
<a href="#">BriarVideoIterator</a>	124
<a href="#">DatabaseTest</a>	127
<a href="#">DetectTest</a>	128
<a href="#">EnrollTest</a>	130
<a href="#">ExtractTest</a>	131
<a href="#">FileVideoStream_cv2</a>	133
<a href="#">FileVideoStream_imageio</a>	137
<a href="#">Fore</a>	141
<a href="#">GrpcDecoder</a>	
Object which extends the JSONDecoded to allow it to read saved gRPC files	142
<a href="#">GrpcEncoder</a>	
Encoder class which extends the normal JSON encoder to allow the encoding of gRPC objects	144
<a href="#">ImageIterator</a>	146
<a href="#">match_matrix_visualizer</a>	151
<a href="#">MediaSetIterator</a>	154
<a href="#">MediaStream</a>	157
<a href="#">Rect</a>	158
<a href="#">Test000InitialConfig</a>	160
<a href="#">Test001SigsetEnrollProbe</a>	163
<a href="#">Test002ProbeDatabaseMerge</a>	164
<a href="#">Test003SigsetEnrollMultiProbe</a>	167
<a href="#">Test004MultiProbeDatabaseMerge</a>	168
<a href="#">Test005SigsetEnrollGalleries</a>	170
<a href="#">Test006GalleryDatabaseMerge</a>	173
<a href="#">Test007SigsetScoreG1</a>	175
<a href="#">Test008SigsetScoreGaitG1</a>	176
<a href="#">Test009SigsetScoreFaceG1</a>	178
<a href="#">Test010SigsetScoreWholeBodyG1</a>	179
<a href="#">Test011SigsetScoreG2</a>	180

Test012SigsetScoreGaitG2 . . . . .	182
Test013SigsetScoreFaceG2 . . . . .	183
Test014SigsetScoreWholeBodyG2 . . . . .	184
Test015SigsetBlendedScoreG1 . . . . .	186
Test016SigsetScoreBlendedGaitG1 . . . . .	187
Test017SigsetScoreBlendedFaceG1 . . . . .	188
Test018SigsetScoreBlendedWholeBodyG1 . . . . .	190
Test019SigsetBlendedScoreG2 . . . . .	191
Test020SigsetScoreBlendedGaitG2 . . . . .	192
Test021SigsetScoreBlendedFaceG2 . . . . .	194
Test022SigsetScoreBlendedWholeBodyG2 . . . . .	195
Test023MultiSigsetScoreG1 . . . . .	196
Test023SigsetSearchOutputFormatting . . . . .	198
Test024MultiSigsetScoreGaitG1 . . . . .	201
Test024SigsetVerifyOutputFormatting . . . . .	203
Test025MultiSigsetScoreFaceG1 . . . . .	206
Test026MultiSigsetScoreWholeBodyG1 . . . . .	208
Test027MultiSigsetScoreG2 . . . . .	209
Test028MultiSigsetScoreGaitG2 . . . . .	210
Test029MultiSigsetScoreFaceG2 . . . . .	212
Test030MultiSigsetScoreWholeBodyG2 . . . . .	213
Test031MultiSigsetBlendedScoreG1 . . . . .	214
Test032MultiSigsetScoreBlendedGaitG1 . . . . .	216
Test033MultiSigsetScoreBlendedFaceG1 . . . . .	217
Test034MultiSigsetScoreBlendedWholeBodyG1 . . . . .	218
Test035MultiSigsetBlendedScoreG2 . . . . .	220
Test036MultiSigsetScoreBlendedGaitG2 . . . . .	221
Test037MultiSigsetScoreBlendedFaceG2 . . . . .	222
Test038MultiSigsetScoreBlendedWholeBodyG2 . . . . .	224
ThreadedVideoIterator . . . . .	225
VideoIterator . . . . .	231

## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

<a href="#">__init__.py</a>	237
<a href="#">__main__.py</a>	240
<a href="#">briar_cli.py</a>	240
<a href="#">briar_client.py</a>	240
<a href="#">briar_media.py</a>	241
<a href="#">grpc_json.py</a>	255
<a href="#">media_converters.py</a>	257
<a href="#">cli/__init__.py</a>	238
<a href="#">cli/connection.py</a>	241
<a href="#">cli/detect.py</a>	247
<a href="#">cli/enhance.py</a>	247
<a href="#">cli/enroll.py</a>	248
<a href="#">cli/extract.py</a>	248
<a href="#">cli/finalize.py</a>	244
<a href="#">cli/media.py</a>	249
<a href="#">cli/search.py</a>	249
<a href="#">cli/sigset.py</a>	249
<a href="#">cli/status.py</a>	250
<a href="#">cli/test.py</a>	250
<a href="#">cli/track.py</a>	251
<a href="#">cli/verify.py</a>	251
<a href="#">cli/viz.py</a>	252
<a href="#">cli/database/__init__.py</a>	238
<a href="#">cli/database/checkpoint.py</a>	242
<a href="#">cli/database/checkpoint_subject.py</a>	242
<a href="#">cli/database/common.py</a>	242
<a href="#">cli/database/compute_scores.py</a>	242
<a href="#">cli/database/compute_search.py</a>	243
<a href="#">cli/database/create.py</a>	243
<a href="#">cli/database/delete.py</a>	243
<a href="#">cli/database/finalize.py</a>	244
<a href="#">cli/database/info.py</a>	244
<a href="#">cli/database/list.py</a>	244
<a href="#">cli/database/list_entries.py</a>	245
<a href="#">cli/database/load.py</a>	245

<a href="#">cli/database/merge.py</a>	245
<a href="#">cli/database/move_entry.py</a>	246
<a href="#">cli/database/refresh.py</a>	246
<a href="#">cli/database/rename.py</a>	246
<a href="#">cli/database/retrieve.py</a>	246
<a href="#">evaluation/__init__.py</a>	238
<a href="#">evaluation/full_evaluation.py</a>	252
<a href="#">evaluation/stage1_probe_enroll.py</a>	254
<a href="#">evaluation/stage2.1_gallery1_simple_enroll.py</a>	254
<a href="#">evaluation/stage2.2_gallery2_simple_enroll.py</a>	255
<a href="#">evaluation/stage2.3_gallery1_blended_enroll.py</a>	255
<a href="#">evaluation/stage2.4_gallery2_blended_enroll.py</a>	255
<a href="#">evaluation/stage3_result_scoring.py</a>	255
<a href="#">media/__init__.py</a>	238
<a href="#">media/VideoStream.py</a>	256
<a href="#">media/visualize.py</a>	256
<a href="#">sigset/__init__.py</a>	239
<a href="#">sigset/parse.py</a>	258
<a href="#">timing/__init__.py</a>	239

## Chapter 6

# Namespace Documentation

### 6.1 briar Namespace Reference

#### Namespaces

- [\\_\\_main\\_\\_](#)
- [briar\\_cli](#)

*Created on 2021 at Oak Ridge National Laboratory.*

- [briar\\_client](#)

*Copyright 2021 Oak Ridge National Laboratory.*

- [briar\\_media](#)

*Defines a media class which acts as a wrapper for image and video files.*

- [cli](#)

*The modules contained in the CLI package each contain a function (detect, extract, enroll, etc.) which is a command within the broader CLI toolkit along with the assorted helper functions.*

- [evaluation](#)
- [grpc\\_json](#)

*I couldn't find a good library to write gRPC/protobuf objects to disk so this is my own implementation of a generalized json converter for said objects.*

- [media](#)
- [media\\_converters](#)

*Contained in this are functions for converting numpy arrays into various protobuf objects and back again since numpy arrays cannot be sent directly over gRPC.*

- [sigset](#)
- [timing](#)

#### Classes

- class [Rect](#)

#### Functions

- def [\\_wait\\_forever](#) (server)
- def [CLIServe](#) (serviceClass, add\_custom\_options=None)
- def [dyn\\_import](#) (name)
- def [get\\_process\\_number](#) ()
- def [get\\_thread\\_number](#) ()
- def [multiproc\\_serve](#) (serviceClass, options=None, serve\_port=None)
- def [parse\\_ports](#) (options)
- def [serve](#) (serviceClass, options=None, serve\_port=None)

## Variables

- string `__version__` = '2.3.6'
- `_ONE_DAY` = `datetime.timedelta(days=1)`
- int `DEFAULT_MAX_MESSAGE_SIZE` = `64 * 1024 * 1024 * 8`
- string `DEFAULT_PORT` = "0.0.0.0:50051"
- string `DEFAULT_SERVE_PORT` = "[::]:50051"
- string `PLATFORM` = "UNKNOWN"

## 6.1.1 Function Documentation

### 6.1.1.1 `_wait_forever()`

```
def briar._wait_forever (
    server ) [private]
```

The `_wait_forever` function is a simple function that waits forever. It's used to keep the server running until it receives a `KeyboardInterrupt` (Ctrl+C).

```
:param server: Stop the server
:return: Nothing
:doc-author: Joel Brogan
```

### 6.1.1.2 `CLIServe()`

```
def briar.CLIServe (
    serviceClass,
    add_custom_options = None )
```

```
:param serviceClass: The class that contains the implementation of the service to be served.
:param add_custom_options: A function that can be used to add custom options to the command line parser.
:return: None
```

This method sets up a command line interface (CLI) for serving a gRPC service. It takes in a `serviceClass` parameter to be served. It also takes an optional `add_custom_options` parameter, which is a function that can be used to

The method uses the `'optparse'` module to define and parse command line options. It creates an instance of `'OptionParser'` and adds command line options such as verbosity, maximum message size, port number(s), port range, number of services

If the `add_custom_options` parameter is not `None`, it calls the `add_custom_options` function to add custom options

The method then parses the command line arguments using the `'parse_args'` method of the `'OptionParser'` instance and returns a list of variables, respectively.

Depending on the number of ports specified or the number of services per port, the method either invokes the `'serve'` method of the `serviceClass`. In case of an `IndexError` exception, an `AssertionError` is raised.

Note: This method assumes the existence of other functions and variables such as `DEFAULT_MAX_MESSAGE_SIZE`, `DEFAULT_PORT`, and `PLATFORM`.



### 6.1.1.3 dyn\_import()

```
def briar.dyn_import (
    name )
```

The `dyn_import` function is a helper function that allows you to import modules dynamically. This means that you can pass in the name of a module as a string, and the `dyn_import` function will return the actual module object. For example:

```
:param name: Specify the name of the module to be imported
:return: A module object
:doc-author: Joel Brogan
```

### 6.1.1.4 get\_process\_number()

```
def briar.get_process_number ( )
```

The `get_process_number` function returns the process number of the current process. The main process is denoted by a 0 while all other processes are 1-indexed.

```
:return: The process number of the current process
:doc-author: Joel Brogan
```

### 6.1.1.5 get\_thread\_number()

```
def briar.get_thread_number ( )
```

The `get_thread_number` function returns the thread number of the current thread.

```
:return: The thread number of the current thread
:doc-author: Joel Brogan
```

### 6.1.1.6 multiproc\_serve()

```
def briar.multiproc_serve (
    serviceClass,
    options = None,
    serve_port = None )
```

The `multiproc_serve` function is a wrapper around the `serve` function that allows multiple instances of the same service to be run on different ports. This is useful for running multiple instances of a service in parallel, which can improve performance. The `multiproc_serve` function takes three arguments:

```
:param serviceClass: Specify the class of service that is being served
:param options: Pass in the options for the service
:param serve_port: Specify the port that the server will listen on
:return: The return value of the serve function
:doc-author: Joel Brogan
```

### 6.1.1.7 parse\_ports()

```
def briar.parse_ports (
    options )
```

The `parse_ports` function takes in a string of ports separated by commas, and returns a list of strings. If the port range is greater than 1, then it will return a list with all the ports in that range.

For example: `parse_ports("localhost:8080")` -> `["localhost:8080"]`

`parse_ports("localhost:8000-8002")` -> `["localhost:8000", "localhost:8001", "localhost:8002"]`

```
:param options: Parse the command line arguments
:return: A list of ports
:doc-author: Joel Brogan
```

### 6.1.1.8 serve()

```
def briar.serve (
    serviceClass,
    options = None,
    serve_port = None )
```

The `serve` function is the main entry point for a `BRIARService`. It initializes and runs the service until killed. Initialize and run the `BRIARService`. Runs until killed

```
:param serviceClass: Specify the service class to be used
:param options: Pass in the configuration options for the service
:param serve_port: Specify the port to serve on
:return: A server object
:doc-author: Joel Brogan
```

## 6.1.2 Variable Documentation

### 6.1.2.1 \_\_version\_\_

```
string __version__ = '2.3.6' [private]
```

### 6.1.2.2 \_ONE\_DAY

```
_ONE_DAY = datetime.timedelta(days=1) [private]
```

### 6.1.2.3 DEFAULT\_MAX\_MESSAGE\_SIZE

```
int DEFAULT_MAX_MESSAGE_SIZE = 64 * 1024 * 1024 * 8
```

#### 6.1.2.4 DEFAULT\_PORT

```
string DEFAULT_PORT = "0.0.0.0:50051"
```

#### 6.1.2.5 DEFAULT\_SERVE\_PORT

```
string DEFAULT_SERVE_PORT = '[:]:50051'
```

#### 6.1.2.6 PLATFORM

```
string PLATFORM = "UNKNOWN"
```

## 6.2 briar.\_\_main\_\_ Namespace Reference

## 6.3 briar.briar\_cli Namespace Reference

Created on 2021 at Oak Ridge National Laboratory.

### Functions

- def [briar\\_command\\_line](#) ()  
*Entry point for the CLI - switches on the first command line argument (such as 'status', 'detect', etc) and builds the parser and help messages based upon the callback defined in COMMANDS.*
- def [briar\\_database\\_command\\_line](#) ()  
*Entry point for the Database CLI - switches on the second command line argument (such as 'delete', 'merge', etc) and builds the parser and help messages based upon the callback defined in COMMANDS.*
- def [briar\\_test\\_command\\_line](#) ()  
*Entry point for the Test CLI - switches on the second command line argument (such as 'delete', 'merge', etc) and builds the parser and help messages based upon the callback defined in COMMANDS.*
- def [incomplete](#) ()

### Variables

- dictionary [COMMANDS](#)
- dictionary [DATABASE\\_COMMANDS](#)
- string [DETECTION\\_FILE\\_EXT](#) = ".detection"
- int [FACE\\_COUNT](#) = 0
- string [MATCHES\\_FILE\\_EXT](#) = '.matches'
- string [TEMPLATE\\_FILE\\_EXT](#) = '.template'

### 6.3.1 Detailed Description

Created on 2021 at Oak Ridge National Laboratory.

The Briar Command Line Interface (Briar CLI) provides a universal method to interface with different gRPC created with the compiled protobuf stubs. It provides a series of common functions to run detection and identification on faces, whole bodies, and walking gaits, as well as various database enrollment and search functions. Briar does not implement these detect, extract, enroll, etc functions itself, but rather acts as a means for connecting with servers (outlined with service.py)

#### Author

: Joel Brogan

### 6.3.2 Function Documentation

#### 6.3.2.1 `briar_command_line()`

```
def briar.briar_cli.briar_command_line ( )
```

Entry point for the CLI - switches on the first command line argument (such as 'status', 'detect', etc) and builds the parser and help messages based upon the callback defined in COMMANDS.

Each 'command' should be treated as a 'switch' which defines additional command line arguments.

#### Returns

:

#### 6.3.2.2 `briar_database_command_line()`

```
def briar.briar_cli.briar_database_command_line ( )
```

Entry point for the Database CLI - switches on the second command line argument (such as 'delete', 'merge', etc) and builds the parser and help messages based upon the callback defined in COMMANDS.

Each 'command' should be treated as a 'switch' which defines additional command line arguments.

#### Returns

:

### 6.3.2.3 briar\_test\_command\_line()

```
def briar.briar_cli.briar_test_command_line ( )
```

Entry point for the Test CLI - switches on the second command line argument (such as 'delete', 'merge', etc) and builds the parser and help messages based upon the callback defined in COMMANDS.

Each 'command' should be treated as a 'switch' which defines additional command line arguments.

#### Returns

:

### 6.3.2.4 incomplete()

```
def briar.briar_cli.incomplete ( )
```

The incomplete function is a placeholder for functions that have not yet been implemented. It raises a `NotImplementedError` exception to indicate that the function has not yet been implemented.

```
:return: A notimplementederror  
:doc-author: Joel Brogan
```

## 6.3.3 Variable Documentation

### 6.3.3.1 COMMANDS

dictionary COMMANDS

#### Initial value:

```
1 = {  
2     'detect': ['Run detection on media files.', detect],  
3     'track': ['Run tracking on media files (videos only).', track],  
4     'enhance': ['Run tracking on media files (videos only).', enhance],  
5     'extract': ['Run feature extraction to generate templates or embeddings.', extract],  
6     'enroll': ['Scan media files and enroll templates into a database.', enroll],  
7     'verify': ['Verify a given peice of media against a reference media.', verify],  
8     'finalize': ['Finalize a database.', database_finalize],  
9     'search': ['Search a database for an example identity.', search],  
10    'sigset-stats': ['Convert a sigset to a csv file and compute statistics.', sigset_stats],  
11    'sigset-enroll': ['Process a sigset and enroll in a database.', sigset_enroll],  
12    'status': ['Connects to the server and displays version and status information.', status],  
13    'service-configuration': ['Returns configuration settings from the service',  
14                               print_service_configuration],  
15    'database': ['Base call for database-related functions', briar_database_command_line],  
16    'vis': ['Visualize saved results output from API Calls', viz],  
17    'test': ['Base call for API test functions', briar_test_command_line]  
}
```

### 6.3.3.2 DATABASE\_COMMANDS

dictionary DATABASE\_COMMANDS

Initial value:

```

1 = {
2     'create': ['Create and initialize a new database.', database_create],
3     'delete': ['Delete a database from the service.', database_delete],
4     'rename': ['Rename a database from the service.', database_rename],
5     '!insert': ['Insert templates directly into a database.', incomplete],
6     'load': ['Load database onto the server.', database_load],
7     'list': ['List the names of the databases on this service.', database_list],
8     'ls': ['List the names of the databases on this service.', database_list],
9     'info': ['List information about a given database.', database_info],
10    'list-entries': ['List the entries contained within a database stored on this service.',
11                     database_list_entries],
12    'finalize': ['Finalize a database.', database_finalize],
13    'checkpoint': ['Checkpoint a database to save progress, without finalizing', database_checkpoint],
14    'checkpoint-subject': ['Checkpoint a database subject to save progress, without finalizing',
15                           database_checkpoint_subject],
16    'compute-search': ['Searches a probe database against a gallery database', database_compute_search],
17    'compute-verify': ['Performs batch verification ', database_compute_verify],
18    'refresh': ['Performs a refresh of the list of databases to keep them coherent between
19                services', database_refresh],
20    '!remove-entries': ['Remove entries from the database', incomplete],
21    'merge': ['Merge a list of existing databases together', database_merge],
22 }
```

### 6.3.3.3 DETECTION\_FILE\_EXT

string DETECTION\_FILE\_EXT = ".detection"

### 6.3.3.4 FACE\_COUNT

int FACE\_COUNT = 0

### 6.3.3.5 MATCHES\_FILE\_EXT

string MATCHES\_FILE\_EXT = '.matches'

### 6.3.3.6 TEMPLATE\_FILE\_EXT

string TEMPLATE\_FILE\_EXT = '.template'

## 6.4 briar.briar\_client Namespace Reference

## Classes

- class [BriarClient](#)

*Provide a client to a BRIAR service.*

## Functions

- def [\\_initialize\\_worker](#) (server\_address, proc\_number, thread\_number, count\_q)
- def [\\_shutdown\\_worker](#) ()

## Variables

- [\\_client\\_identifier\\_singleton](#) = None
- [\\_worker\\_channel\\_singleton](#) = None
- [\\_worker\\_port\\_singleton](#) = None
- [\\_worker\\_proccess\\_position\\_singleton](#) = None
- [\\_worker\\_stub\\_singleton](#) = None
- [\\_worker\\_thread\\_position\\_singleton](#) = None

### 6.4.1 Detailed Description

Copyright 2021 Oak Ridge National Laboratory.

The BRIAR API is divided into two primary parts, the client and the service. This, the client, is the part which interfaces with grpc servers based off `briar.service.BRIARService` using the `BRIARServiceStub`. The service stub contains the same methods contained in the service which, when invoked with the appropriate request, sends said request to the service which the client is connected to, and accepts the reply containing processed detections, extracts, templates, etc...

The BRIAR client is designed to serve as a unified interface with gRPC services which are designed after `BRIARService` and implement various performer algorithms for face and body detection/extraction. From a performer standpoint, The BRIAR client can be used as either part of the command line tools, or invoked alone as a module

### 6.4.2 Function Documentation

#### 6.4.2.1 `_initialize_worker()`

```
def briar.briar_client._initialize_worker (
    server_address,
    proc_number,
    thread_number,
    count_q ) [private]
```

#### 6.4.2.2 `_shutdown_worker()`

```
def briar.briar_client._shutdown_worker ( ) [private]
```

### 6.4.3 Variable Documentation

#### 6.4.3.1 `_client_identifier_singleton`

```
_client_identifier_singleton = None [private]
```

#### 6.4.3.2 `_worker_channel_singleton`

```
_worker_channel_singleton = None [private]
```

#### 6.4.3.3 `_worker_port_singleton`

```
_worker_port_singleton = None [private]
```

#### 6.4.3.4 `_worker_proccess_position_singleton`

```
_worker_proccess_position_singleton = None [private]
```

#### 6.4.3.5 `_worker_stub_singleton`

```
_worker_stub_singleton = None [private]
```

#### 6.4.3.6 `_worker_thread_position_singleton`

```
_worker_thread_position_singleton = None [private]
```



## 6.5 briar.briar\_media Namespace Reference

Defines a media class which acts as a wrapper for image and video files.

### Classes

- class [BriarMedia](#)
- class [MediaStream](#)

### Functions

- def [briar\\_media\\_from\\_pb2](#) (pb2\_object)
- def [briar\\_media\\_to\\_pb2](#) (media)
- def [load\\_media\\_from\\_folder](#) (folder\_path, recursive=False)
- def [load\\_media\\_from\\_image](#) (image\_path)
- def [load\\_media\\_from\\_numpy](#) (numpy\_array)

#### 6.5.1 Detailed Description

Defines a media class which acts as a wrapper for image and video files.

#### 6.5.2 Function Documentation

##### 6.5.2.1 [briar\\_media\\_from\\_pb2\(\)](#)

```
def briar.briar_media.briar_media_from_pb2 (  
    pb2_object )
```

##### 6.5.2.2 [briar\\_media\\_to\\_pb2\(\)](#)

```
def briar.briar_media.briar_media_to_pb2 (  
    media )
```

##### 6.5.2.3 [load\\_media\\_from\\_folder\(\)](#)

```
def briar.briar_media.load_media_from_folder (  
    folder_path,  
    recursive = False )
```

#### 6.5.2.4 load\_media\_from\_image()

```
def briar.briar_media.load_media_from_image (
    image_path )
```

#### 6.5.2.5 load\_media\_from\_numpy()

```
def briar.briar_media.load_media_from_numpy (
    numpy_array )
```

## 6.6 briar.cli Namespace Reference

The modules contained in the CLI package each contain a function (detect, extract, enroll, etc.) which is a command within the broader CLI toolkit along with the assorted helper functions.

### Namespaces

- [connection](#)
- [database](#)
- [detect](#)
- [enhance](#)
- [enroll](#)
- [extract](#)
- [finalize](#)
- [media](#)
- [search](#)
- [sigset](#)
- [status](#)
- [test](#)
- [track](#)
- [verify](#)
- [viz](#)

#### 6.6.1 Detailed Description

The modules contained in the CLI package each contain a function (detect, extract, enroll, etc.) which is a command within the broader CLI toolkit along with the assorted helper functions.

The [briar\\_cli](#) file has each of these important functions mapped in a dictionary which accesses them based off of user commands. The module functions then add additional command line options which can be parsed into options or displayed as part of a help message. From there, the module functions will connect to the specified service and send it messages and receive replies based off of the arguments passed in through the command line.

## 6.7 briar.cli.connection Namespace Reference

### Functions

- def `addConnectionOptions` (parser)  
*Accumulatively add options for connecting to the Briar API service.*

### Variables

- int `DEFAULT_MAX_ASYNC` = 8
- int `DEFAULT_MAX_MESSAGE_SIZE` = 64 \* 1024 \* 1024 \* 8

### 6.7.1 Function Documentation

#### 6.7.1.1 `addConnectionOptions()`

```
def briar.cli.connection.addConnectionOptions (
    parser )
```

Accumulatively add options for connecting to the Briar API service.

Modifiers the parser in plase

#### Parameters

<code>parser</code>	optparse.OptionParser: A parser to modify in place by adding connection options
---------------------	---

### 6.7.2 Variable Documentation

#### 6.7.2.1 `DEFAULT_MAX_ASYNC`

```
int DEFAULT_MAX_ASYNC = 8
```

#### 6.7.2.2 `DEFAULT_MAX_MESSAGE_SIZE`

```
int DEFAULT_MAX_MESSAGE_SIZE = 64 * 1024 * 1024 * 8
```

## 6.8 briar.cli.database Namespace Reference

### Namespaces

- [checkpoint](#)
- [checkpoint\\_subject](#)
- [common](#)
- [compute\\_scores](#)
- [compute\\_search](#)
- [create](#)
- [delete](#)
- [finalize](#)
- [info](#)
- [list](#)
- [list\\_entries](#)
- [load](#)
- [merge](#)
- [move\\_entry](#)
- [refresh](#)
- [rename](#)
- [retrieve](#)

## 6.9 briar.cli.database.checkpoint Namespace Reference

### Functions

- def [database\\_checkpoint](#) (options=None, args=None, input\_command=None, ret=False)

### 6.9.1 Function Documentation

#### 6.9.1.1 database\_checkpoint()

```
def briar.cli.database.checkpoint.database_checkpoint (
    options = None,
    args = None,
    input_command = None,
    ret = False )
```

Checkpoints a database without finalizing it

## 6.10 briar.cli.database.checkpoint\_subject Namespace Reference

### Functions

- def [database\\_checkpoint\\_subject](#) (options=None, args=None, input\_command=None, ret=False)
- def [parseDatabaseCheckpointSubjectOptions](#) (inputCommand=None)
 

*Generate options for getting information about a pre-existing database and parse command line arguments into an API call.*

## 6.10.1 Function Documentation

### 6.10.1.1 database\_checkpoint\_subject()

```
def briar.cli.database.checkpoint_subject.database_checkpoint_subject (
    options = None,
    args = None,
    input_command = None,
    ret = False )
```

### 6.10.1.2 parseDatabaseCheckpointSubjectOptions()

```
def briar.cli.database.checkpoint_subject.parseDatabaseCheckpointSubjectOptions (
    inputCommand = None )
```

Generate options for getting information about a pre-existing database and parse command line arguments into an API call.

#### Returns

: 2 element Tuple of (optparse.Values, list) containing the parsed options and parameters respectively

## 6.11 briar.cli.database.common Namespace Reference

### Functions

- def [db\\_no\\_exist](#)(name)

### 6.11.1 Function Documentation

#### 6.11.1.1 db\_no\_exist()

```
def briar.cli.database.common.db_no_exist (
    name )
```

The `db_no_exist` function is called when the user attempts to access a database that does not exist. It prints an error message and returns `None`.

```
:param name: Print the name of the database that doesn't exist
:return: A string
:doc-author: Joel Brogan
```

## 6.12 briar.cli.database.compute\_scores Namespace Reference

### Functions

- def [addDatabaseComputeScore\\_options2proto](#) (options)
- def [addDatabaseComputeScoreOptions](#) (parser)  
*Add options for search of a database using a database.*
- def [database\\_compute\\_verify](#) (options=None, args=None, input\_command=None, ret=False)  
*Using the options specified in the command line, runs a batch verification between the specified databases using specified probe database.*
- def [parseDatabaseComputeScoreOptions](#) (inputCommand=None)  
*Generate options for matching databases against other databases, and parse command line arguments into the API call.*

### 6.12.1 Function Documentation

#### 6.12.1.1 addDatabaseComputeScore\_options2proto()

```
def briar.cli.database.compute_scores.addDatabaseComputeScore_options2proto (
    options )
```

Parse command line options and populate a proto object for grpc

#### 6.12.1.2 addDatabaseComputeScoreOptions()

```
def briar.cli.database.compute_scores.addDatabaseComputeScoreOptions (
    parser )
```

Add options for search of a database using a database.

##### Parameters

<i>parser</i>	optparse.OptionParser: A parser to modify in place by adding options
---------------	--

#### 6.12.1.3 database\_compute\_verify()

```
def briar.cli.database.compute_scores.database_compute_verify (
    options = None,
    args = None,
```

```

    input_command = None,
    ret = False )

```

Using the options specified in the command line, runs a batch verification between the specified databases using specified probe database.

Writes results to disk to a location specified by the cmd arguments

#### Returns

: No return - Function writes results to disk

#### 6.12.1.4 parseDatabaseComputeScoreOptions()

```

def briar.cli.database.compute_scores.parseDatabaseComputeScoreOptions (
    inputCommand = None )

```

Generate options for matching databases against other databases, and parse command line arguments into the API call.

#### Returns

: 2 element Tuple of (optparse.Values, list) containing the parsed options and parameters respectively

## 6.13 briar.cli.database.compute\_search Namespace Reference

### Functions

- def [database\\_compute\\_search](#) (options=None, args=None, input\_command=None, ret=False)  
*Using the options specified in the command line, runs a search within the specified database using specified probe database.*

#### 6.13.1 Function Documentation

##### 6.13.1.1 database\_compute\_search()

```

def briar.cli.database.compute_search.database_compute_search (
    options = None,
    args = None,
    input_command = None,
    ret = False )

```

Using the options specified in the command line, runs a search within the specified database using specified probe database.

Writes results to disk to a location specified by the cmd arguments

#### Returns

: No return - Function writes results to disk

## 6.14 briar.cli.database.create Namespace Reference

### Functions

- def [database\\_create](#) (options=None, args=None, input\_command=None, ret=False)

#### 6.14.1 Function Documentation

##### 6.14.1.1 database\_create()

```
def briar.cli.database.create.database_create (
    options = None,
    args = None,
    input_command = None,
    ret = False )
```

Checkpoints a database without finalizing it

## 6.15 briar.cli.database.delete Namespace Reference

### Functions

- def [database\\_delete](#) (options=None, args=None, input\_command=None, ret=False)
- def [parseDatabaseDeleteOptions](#) (inputCommand=None)  
*Generate options for Deleting a pre-existing database and parse command line arguments into API call.*

#### 6.15.1 Function Documentation

##### 6.15.1.1 database\_delete()

```
def briar.cli.database.delete.database_delete (
    options = None,
    args = None,
    input_command = None,
    ret = False )
```

delete a database



### 6.15.1.2 parseDatabaseDeleteOptions()

```
def briar.cli.database.delete.parseDatabaseDeleteOptions (
    inputCommand = None )
```

Generate options for Deleting a pre-existing database and parse command line arguments into API call.

#### Returns

: 2 element Tuple of (optparse.Values, list) containing the parsed options and parameters respectively

## 6.16 briar.cli.database.finalize Namespace Reference

### Functions

- def [database\\_finalize](#) (options=None, args=None, input\_command=None, ret=False)  
*Parses the command line options and saves the database to disk.*
- def [finalizeParseOptions](#) (inputCommand=None)  
*Generate options for running 'finalize' (saving the loaded databases) and parse command line arguments into them.*

### 6.16.1 Function Documentation

#### 6.16.1.1 database\_finalize()

```
def briar.cli.database.finalize.database_finalize (
    options = None,
    args = None,
    input_command = None,
    ret = False )
```

Parses the command line options and saves the database to disk.

#### Returns

: None - results are written to disk to a location specified by options

#### 6.16.1.2 finalizeParseOptions()

```
def briar.cli.database.finalize.finalizeParseOptions (
    inputCommand = None )
```

Generate options for running 'finalize' (saving the loaded databases) and parse command line arguments into them.

#### Returns

: 2 element Tuple of (optparse.Values, list) containing the parsed options and parameters respectively

## 6.17 briar.cli.database.info Namespace Reference

### Functions

- def [database\\_info](#) (options=None, args=None, input\_command=None, ret=False)
- def [parseDatabaseInfoOptions](#) (inputCommand=None)  
*Generate options for getting information about a pre-existing database and parse command line arguments into an API call.*

### 6.17.1 Function Documentation

#### 6.17.1.1 database\_info()

```
def briar.cli.database.info.database_info (
    options = None,
    args = None,
    input_command = None,
    ret = False )

list the information pertaining to a database
```

#### 6.17.1.2 parseDatabaseInfoOptions()

```
def briar.cli.database.info.parseDatabaseInfoOptions (
    inputCommand = None )
```

Generate options for getting information about a pre-existing database and parse command line arguments into an API call.

#### Returns

: 2 element Tuple of (optparse.Values, list) containing the parsed options and parameters respectively

## 6.18 briar.cli.database.list Namespace Reference

### Functions

- def [database\\_list](#) (options=None, args=None, input\_command=None, ret=False)
- def [parseDatabaseListOptions](#) (inputCommand=None)  
*Generate options for listing all pre-existing databases and parse command line arguments into them.*

## 6.18.1 Function Documentation

### 6.18.1.1 database\_list()

```
def briar.cli.database.list.database_list (
    options = None,
    args = None,
    input_command = None,
    ret = False )
```

list the names of the databases

### 6.18.1.2 parseDatabaseListOptions()

```
def briar.cli.database.list.parseDatabaseListOptions (
    inputCommand = None )
```

Generate options for listing all pre-existing databases and parse command line arguments into them.

#### Returns

: 2 element Tuple of (optparse.Values, list) containing the parsed options and parameters respectively

## 6.19 briar.cli.database.list\_entries Namespace Reference

### Functions

- def [database\\_list\\_entries](#) (options=None, args=None, input\_command=None, ret=False)
- def [parseDatabaseListEntriesOptions](#) (inputCommand=None)  
*Generate options for Listing entries within a pre-existing database and parse command line arguments into an API call.*

## 6.19.1 Function Documentation

### 6.19.1.1 database\_list\_entries()

```
def briar.cli.database.list_entries.database_list_entries (
    options = None,
    args = None,
    input_command = None,
    ret = False )
```

list the entries in a database

### 6.19.1.2 `parseDatabaseListEntriesOptions()`

```
def briar.cli.database.list_entries.parseDatabaseListEntriesOptions (
    inputCommand = None )
```

Generate options for Listing entries within a pre-existing database and parse command line arguments into an API call.

#### Returns

: 2 element Tuple of (optparse.Values, list) containing the parsed options and parameters respectively

## 6.20 `briar.cli.database.load` Namespace Reference

### Functions

- def [database\\_load](#) ()

#### 6.20.1 Function Documentation

##### 6.20.1.1 `database_load()`

```
def briar.cli.database.load.database_load ( )
```

Loads a database from storage (in case the database needs a procedure for loading from disk into memory)

## 6.21 `briar.cli.database.merge` Namespace Reference

### Functions

- def [database\\_merge](#) (options=None, args=None, input\_command=None, ret=False)
- def [parseDatabaseMergeOptions](#) (inputCommand=None)  
*Generate options for merging databases and parse command line arguments into the API call.*

#### 6.21.1 Function Documentation

### 6.21.1.1 database\_merge()

```
def briar.cli.database.merge.database_merge (
    options = None,
    args = None,
    input_command = None,
    ret = False )
```

Merge a set of databases

### 6.21.1.2 parseDatabaseMergeOptions()

```
def briar.cli.database.merge.parseDatabaseMergeOptions (
    inputCommand = None )
```

Generate options for merging databases and parse command line arguments into the API call.

#### Returns

: 2 element Tuple of (optparse.Values, list) containing the parsed options and parameters respectively

## 6.22 briar.cli.database.move\_entry Namespace Reference

### Functions

- def [database\\_move\\_entry](#) (options=None, args=None, input\_command=None, ret=False, client=None)
- def [parseDatabaseMoveEntryOptions](#) (inputCommand=None)

*Generate options for moving database entries and parse command line arguments into the API call.*

### 6.22.1 Function Documentation

#### 6.22.1.1 database\_move\_entry()

```
def briar.cli.database.move_entry.database_move_entry (
    options = None,
    args = None,
    input_command = None,
    ret = False,
    client = None )
```

Merge a set of databases

### 6.22.1.2 `parseDatabaseMoveEntryOptions()`

```
def briar.cli.database.move_entry.parseDatabaseMoveEntryOptions (
    inputCommand = None )
```

Generate options for moving database entries and parse command line arguments into the API call.

#### Returns

: 2 element Tuple of (optparse.Values, list) containing the parsed options and parameters respectively

## 6.23 `briar.cli.database.refresh` Namespace Reference

### Functions

- def [database\\_checkpoint](#) (options=None, args=None, input\_command=None, ret=False)
- def [database\\_refresh](#) (options=None, args=None, input\_command=None, ret=False)
- def [parseDatabaseRefreshOptions](#) (inputCommand=None)  
*Generate options for getting information about a pre-existing database and parse command line arguments into an API call.*

### 6.23.1 Function Documentation

#### 6.23.1.1 `database_checkpoint()`

```
def briar.cli.database.refresh.database_checkpoint (
    options = None,
    args = None,
    input_command = None,
    ret = False )
```

Checkpoints a database without finalizing it

#### 6.23.1.2 `database_refresh()`

```
def briar.cli.database.refresh.database_refresh (
    options = None,
    args = None,
    input_command = None,
    ret = False )
```

Refresh the list of databases

### 6.23.1.3 parseDatabaseRefreshOptions()

```
def briar.cli.database.refresh.parseDatabaseRefreshOptions (
    inputCommand = None )
```

Generate options for getting information about a pre-existing database and parse command line arguments into an API call.

#### Returns

: 2 element Tuple of (optparse.Values, list) containing the parsed options and parameters respectively

## 6.24 briar.cli.database.rename Namespace Reference

### Functions

- def [database\\_load](#) ()
- def [database\\_rename](#) (options=None, args=None, input\_command=None, ret=False)
- def [parseDatabaseRenameOptions](#) (inputCommand=None)  
*Generate options for Renaming a pre-existing database to a new name and parse command line arguments into API call.*

### 6.24.1 Function Documentation

#### 6.24.1.1 database\_load()

```
def briar.cli.database.rename.database_load ( )
```

Loads a database from storage

#### 6.24.1.2 database\_rename()

```
def briar.cli.database.rename.database_rename (
    options = None,
    args = None,
    input_command = None,
    ret = False )
```

rename a database

### 6.24.1.3 parseDatabaseRenameOptions()

```
def briar.cli.database.rename.parseDatabaseRenameOptions (
    inputCommand = None )
```

Generate options for Renaming a pre-existing database to a new name and parse command line arguments into API call.

#### Returns

: 2 element Tuple of (optparse.Values, list) containing the parsed options and parameters respectively

## 6.25 briar.cli.database.retrieve Namespace Reference

### Functions

- def [database\\_retrieve](#) (options=None, args=None, input\_command=None, ret=False)
- def [parseDatabaseRetrieveOptions](#) (inputCommand=None)  
*Generate options for retrieving a pre-existing database and parse command line arguments API call.*

### 6.25.1 Function Documentation

#### 6.25.1.1 database\_retrieve()

```
def briar.cli.database.retrieve.database_retrieve (
    options = None,
    args = None,
    input_command = None,
    ret = False )
```

list the entries in a database

#### 6.25.1.2 parseDatabaseRetrieveOptions()

```
def briar.cli.database.retrieve.parseDatabaseRetrieveOptions (
    inputCommand = None )
```

Generate options for retrieving a pre-existing database and parse command line arguments API call.

#### Returns

: 2 element Tuple of (optparse.Values, list) containing the parsed options and parameters respectively



## 6.26 briar.cli.detect Namespace Reference

### Functions

- def [addDetectorOptions](#) (parser)  
*Add options for running detections to the parser.*
- def [addTrackingOptions](#) (parser)  
*Add options for running detections to the parser.*
- def [detect](#) (options=None, args=None, input\_command=None, ret=False)  
*Using the options specified in the command line, runs a detection on the specified files.*
- def [detect\\_options2proto](#) (options)
- def [detectParseOptions](#) (inputCommand=None)  
*Generate options for running detections and parse command line arguments into them.*
- def [detectRequestConstructor](#) (briar\_pb2.BriarMedia media, briar\_pb2.BriarDurations durations, options\_dict={}, det\_list\_list=None, str database\_name=None)
- def [get\\_detection\\_path](#) (media\_file, options, i, modality=None, media\_id=None)
- def [isFinalReply](#) (srv\_pb2.DetectReply reply)
- def [save\\_detections](#) (media\_file, reply, options, i, modality=None, media\_id=None)
- def [tracking\\_options2proto](#) (options)

### Variables

- string [DETECTION\\_FILE\\_EXT](#) = ".detection"

### 6.26.1 Function Documentation

#### 6.26.1.1 addDetectorOptions()

```
def briar.cli.detect.addDetectorOptions (
    parser )
```

Add options for running detections to the parser.

Modifies the parser in place

#### Parameters

<i>parser</i>	optparse.OptionParser: A parser to modify in place by adding options
---------------	--

#### 6.26.1.2 addTrackingOptions()

```
def briar.cli.detect.addTrackingOptions (
    parser )
```

Add options for running detections to the parser.

Modifies the parser in place

#### Parameters

<i>parser</i>	optparse.OptionParser: A parser to modify in place by adding options
---------------	--

### 6.26.1.3 detect()

```
def briar.cli.detect.detect (
    options = None,
    args = None,
    input_command = None,
    ret = False )
```

Using the options specified in the command line, runs a detection on the specified files.

Writes results to disk to a location specified by the cmd arguments

#### Returns

: No return - Function writes results to disk

### 6.26.1.4 detect\_options2proto()

```
def briar.cli.detect.detect_options2proto (
    options )
```

Parse command line options and populate a proto object for grpc

### 6.26.1.5 detectParseOptions()

```
def briar.cli.detect.detectParseOptions (
    inputCommand = None )
```

Generate options for running detections and parse command line arguments into them.

#### Returns

: 2 element Tuple of (optparse.Values, list) containing the parsed options and parameters respectively

#### 6.26.1.6 detectRequestConstructor()

```
def briar.cli.detect.detectRequestConstructor (
    briar_pb2.BriarMedia media,
    briar_pb2.BriarDurations durations,
    options_dict = {},
    det_list_list = None,
    str database_name = None )
```

#### 6.26.1.7 get\_detection\_path()

```
def briar.cli.detect.get_detection_path (
    media_file,
    options,
    i,
    modality = None,
    media_id = None )
```

#### 6.26.1.8 isFinalReply()

```
def briar.cli.detect.isFinalReply (
    srv_pb2.DetectReply reply )
```

#### 6.26.1.9 save\_detections()

```
def briar.cli.detect.save_detections (
    media_file,
    reply,
    options,
    i,
    modality = None,
    media_id = None )
```

#### 6.26.1.10 tracking\_options2proto()

```
def briar.cli.detect.tracking_options2proto (
    options )
```

Parse command line options and populate a proto object for grpc

## 6.26.2 Variable Documentation

### 6.26.2.1 DETECTION\_FILE\_EXT

```
string DETECTION_FILE_EXT = ".detection"
```

## 6.27 briar.cli.enhance Namespace Reference

### Functions

- def [addEnhanceOptions](#) (parser)  
*Add options for running detections to the parser.*
- def [enhance](#) (options=None, args=None, input\_command=None, ret=False)  
*Using the options specified in the command line, runs a detection on the specified files.*
- def [enhance\\_options2proto](#) (options)
- def [enhanceParseOptions](#) (inputCommand=None)  
*Generate options for running enhancement and parse command line arguments into them.*
- def [enhanceRequestConstructor](#) (briar\_pb2.BriarMedia media, briar\_pb2.BriarDurations durations, options←\_dict={}, det\_list\_list=None, str database\_name=None)
- def [save\\_Enhancement](#) (media\_file, reply, options, i, modality=None)

### Variables

- string [ENHANCE\\_FILE\\_EXT](#) = '.enh'

## 6.27.1 Function Documentation

### 6.27.1.1 addEnhanceOptions()

```
def briar.cli.enhance.addEnhanceOptions (
    parser )
```

Add options for running detections to the parser.

Modifies the parser in place

#### Parameters

<i>parser</i>	optparse.OptionParser: A parser to modify in place by adding options
---------------	--

### 6.27.1.2 enhance()

```
def briar.cli.enhance.enhance (
    options = None,
    args = None,
    input_command = None,
    ret = False )
```

Using the options specified in the command line, runs a detection on the specified files.

Writes results to disk to a location specified by the cmd arguments

#### Returns

: No return - Function writes results to disk

### 6.27.1.3 enhance\_options2proto()

```
def briar.cli.enhance.enhance_options2proto (
    options )
```

Parse command line options and populate a proto object for grpc

### 6.27.1.4 enhanceParseOptions()

```
def briar.cli.enhance.enhanceParseOptions (
    inputCommand = None )
```

Generate options for running enhancement and parse command line arguments into them.

#### Returns

: 2 element Tuple of (optparse.Values, list) containing the parsed options and parameters respectively

### 6.27.1.5 enhanceRequestConstructor()

```
def briar.cli.enhance.enhanceRequestConstructor (
    briar_pb2.BriarMedia media,
    briar_pb2.BriarDurations durations,
    options_dict = {},
    det_list_list = None,
    str database_name = None )
```

### 6.27.1.6 save\_Enhancement()

```
def briar.cli.enhance.save_Enhancement (
    media_file,
    reply,
    options,
    i,
    modality = None )
```

## 6.27.2 Variable Documentation

### 6.27.2.1 ENHANCE\_FILE\_EXT

```
string ENHANCE_FILE_EXT = '.enh'
```

## 6.28 briar.cli.enroll Namespace Reference

### Functions

- def [addEnrollOptions](#) (parser)  
*Add options for enrollment into a database.*
- def [enroll](#) (options=None, args=None, input\_command=None, ret=False)  
*Using the options specified in the command line, runs an enroll on the specified files.*
- def [enroll\\_options2proto](#) (options)
- def [enrollParseOptions](#) (inputCommand=None)  
*Generate options for running enrollments and parse command line arguments into them.*
- def [enrollRequestConstructor](#) (briar\_pb2.BriarMedia media, briar\_pb2.BriarDurations durations, options\_↔ dict={}, det\_list\_list=None, str database\_name=None)

### 6.28.1 Function Documentation

#### 6.28.1.1 addEnrollOptions()

```
def briar.cli.enroll.addEnrollOptions (
    parser )
```

Add options for enrollment into a database.

#### Parameters

<i>parser</i>	optparse.OptionParser: A parser to modify in place by adding options
---------------	--

### 6.28.1.2 enroll()

```
def briar.cli.enroll.enroll (
    options = None,
    args = None,
    input_command = None,
    ret = False )
```

Using the options specified in the command line, runs an enroll on the specified files.

Can enroll media files (runs auto detection), detections (auto extracts ROI defined by detects) or templates (skips detect/extract)

Writes results to disk to a location specified by the cmd arguments.

#### Returns

: No return - Function writes results to disk

### 6.28.1.3 enroll\_options2proto()

```
def briar.cli.enroll.enroll_options2proto (
    options )
```

The enroll\_options2proto function takes an EnrollOptions object and converts it to a protobuf message.

:param options: Pass in the options for enrolling a template  
:return: A proto object that can be sent to the server  
:doc-author: Joel Brogan

### 6.28.1.4 enrollParseOptions()

```
def briar.cli.enroll.enrollParseOptions (
    inputCommand = None )
```

Generate options for running enrollments and parse command line arguments into them.

#### Returns

: 2 element Tuple of (optparse.Values, list) containing the parsed options and parameters respectively

### 6.28.1.5 enrollRequestConstructor()

```
def briar.cli.enroll.enrollRequestConstructor (
    briar_pb2.BriarMedia media,
    briar_pb2.BriarDurations durations,
    options_dict = {},
    det_list_list = None,
    str database_name = None )
```

The enrollRequestConstructor function is used to construct an EnrollRequest object.

The EnrollRequest object contains the following fields:

database - The name of the database that will be used for enrollment. If no name is provided, then a default database will be created and used for enrollment.

```
:param media: briar_pb2.BriarMedia: Specify the media type
:param durations: briar_pb2.BriarDurations: Record the time it takes to perform each step of the
:param options_dict: Pass the options for each of the three stages of processing
:param det_list_list: Pass the detections from the previous frame to the next frame
:param database_name: str: Specify the name of the database to enroll into
:return: A enrollrequest object
:doc-author: Joel Brogan
```

## 6.29 briar.cli.extract Namespace Reference

### Functions

- def [addExtractOptions](#) (parser)  
*Add options for extractions to the parser.*
- def [extract](#) (options=None, args=None, inputCommand=None, ret=False)  
*Using the options specified in the command line, runs an extract on the specified files.*
- def [extract\\_options2proto](#) (options)
- def [extractParseOptions](#) (inputCommand=None)  
*Generate options for running extracts and parse command line arguments into them.*
- def [extractRequestConstructor](#) (briar\_pb2.BriarMedia media, briar\_pb2.BriarDurations durations, options\_dict={}, det\_list\_list=None, str database\_name=None)
- def [save\\_extractions](#) (media\_file, templates, options, i, modality=None, media\_id=None)

### Variables

- string [TEMPLATE\\_FILE\\_EXT](#) = '.template'

## 6.29.1 Function Documentation

### 6.29.1.1 addExtractOptions()

```
def briar.cli.extract.addExtractOptions (
    parser )
```

Add options for extractions to the parser.

@type parser: optparse.OptionParser



**Parameters**

<i>parser</i>	A parser to modify in place by adding options
---------------	---

**6.29.1.2 extract()**

```
def briar.cli.extract.extract (
    options = None,
    args = None,
    inputCommand = None,
    ret = False )
```

Using the options specified in the command line, runs an extract on the specified files.

Writes results to disk to a location specified by the cmd arguments

**Returns**

: No return - Function writes results to disk

**6.29.1.3 extract\_options2proto()**

```
def briar.cli.extract.extract_options2proto (
    options )
```

**6.29.1.4 extractParseOptions()**

```
def briar.cli.extract.extractParseOptions (
    inputCommand = None )
```

Generate options for running extracts and parse command line arguments into them.

**Returns**

: 2 element Tuple of (optparse.Values, list) containing the parsed options and parameters respectively

#### 6.29.1.5 extractRequestConstructor()

```
def briar.cli.extract.extractRequestConstructor (
    briar_pb2.BriarMedia media,
    briar_pb2.BriarDurations durations,
    options_dict = {},
    det_list_list = None,
    str database_name = None )
```

#### 6.29.1.6 save\_extractions()

```
def briar.cli.extract.save_extractions (
    media_file,
    templates,
    options,
    i,
    modality = None,
    media_id = None )
```

### 6.29.2 Variable Documentation

#### 6.29.2.1 TEMPLATE\_FILE\_EXT

```
string TEMPLATE_FILE_EXT = '.template'
```

## 6.30 briar.cli.finalize Namespace Reference

### Functions

- def [database\\_finalize](#) (options=None, args=None, input\_command=None, ret=False)  
*Parses the command line options and saves the database to disk.*
- def [finalizeParseOptions](#) (inputCommand=None)  
*Generate options for running 'finalize' (saving the loaded databases) and parse command line arguments into them.*

#### 6.30.1 Function Documentation

### 6.30.1.1 database\_finalize()

```
def briar.cli.finalize.database_finalize (
    options = None,
    args = None,
    input_command = None,
    ret = False )
```

Parses the command line options and saves the database to disk.

#### Returns

: None - results are written to disk to a location specified by options

### 6.30.1.2 finalizeParseOptions()

```
def briar.cli.finalize.finalizeParseOptions (
    inputCommand = None )
```

Generate options for running 'finalize' (saving the loaded databases) and parse command line arguments into them.

#### Returns

: 2 element Tuple of (optparse.Values, list) containing the parsed options and parameters respectively

## 6.31 briar.cli.media Namespace Reference

### Functions

- def [addMediaOptions](#) (parser)  
*Add options for running detections to the parser.*
- def [collect\\_files](#) (args, options, extension=None)  
*Take the paths specified by 'args' and find all the media files that they define: folders will be searched for all media files contained inside.*
- def [hasExtension](#) (f, extension)

### Variables

- int [DEFAULT\\_MAX\\_SIZE](#) = 1920

### 6.31.1 Function Documentation

#### 6.31.1.1 addMediaOptions()

```
def briar.cli.media.addMediaOptions (
    parser )
```

Add options for running detections to the parser.

Modifies the parser in place

## Parameters

<i>parser</i>	optparse.OptionParser: A parser to modify in place by adding options
---------------	--

**6.31.1.2 collect\_files()**

```
def briar.cli.media.collect_files (
    args,
    options,
    extension = None )
```

Take the paths specified by 'args' and find all the media files that they define: folders will be searched for all media files contained inside.

## Parameters

<i>args</i>	list(str): List of paths to add as/search for media files
<i>options</i>	optparse.Values: Command line options which dictate collect behavior
<i>extension</i>	str: A specific extension which defines the csv files associated with media.

## Returns

: Return value depends on 'extension' If 'extension' is None, Tuple will be two elements (list of str, list of str) representing lists of image paths and video paths respectively

If 'extension' is not None, returns a single list of csv files with extensions matching 'extension'

**6.31.1.3 hasExtension()**

```
def briar.cli.media.hasExtension (
    f,
    extension )
```

**6.31.2 Variable Documentation****6.31.2.1 DEFAULT\_MAX\_SIZE**

```
int DEFAULT_MAX_SIZE = 1920
```

## 6.32 briar.cli.search Namespace Reference

### Functions

- def [addSearchOptions](#) (parser)  
*Add options for search of a database.*
- def [search](#) (options=None, args=None, input\_command=None, ret=False)  
*Using the options specified in the command line, runs a search within the specified database using specified probe template(s).*
- def [search\\_options2proto](#) (options)
- def [searchParseOptions](#) (inputCommand=None)  
*Generate options for running searches and parse command line arguments into them.*
- def [searchRequestConstructor](#) (briar\_pb2.BriarMedia media, briar\_pb2.BriarDurations durations, options\_  
dict={}, det\_list\_list=None, str database\_name=None)

### Variables

- string [MATCHES\\_FILE\\_EXT](#) = '.matches'

### 6.32.1 Function Documentation

#### 6.32.1.1 addSearchOptions()

```
def briar.cli.search.addSearchOptions (
    parser )
```

Add options for search of a database.

##### Parameters

<i>parser</i>	optparse.OptionParser: A parser to modify in place by adding options
---------------	--

#### 6.32.1.2 search()

```
def briar.cli.search.search (
    options = None,
    args = None,
    input_command = None,
    ret = False )
```

Using the options specified in the command line, runs a search within the specified database using specified probe template(s).

Writes results to disk to a location specified by the cmd arguments

**Returns**

: No return - Function writes results to disk

**6.32.1.3 search\_options2proto()**

```
def briar.cli.search.search_options2proto (
    options )
```

Parse command line options and populate a proto object for grpc

**6.32.1.4 searchParseOptions()**

```
def briar.cli.search.searchParseOptions (
    inputCommand = None )
```

Generate options for running searches and parse command line arguments into them.

**Returns**

: 2 element Tuple of (optparse.Values, list) containing the parsed options and parameters respectively

**6.32.1.5 searchRequestConstructor()**

```
def briar.cli.search.searchRequestConstructor (
    briar_pb2.BriarMedia media,
    briar_pb2.BriarDurations durations,
    options_dict = {},
    det_list_list = None,
    str database_name = None )
```

**6.32.2 Variable Documentation****6.32.2.1 MATCHES\_FILE\_EXT**

```
string MATCHES_FILE_EXT = '.matches'
```

## 6.33 briar.cli.sigset Namespace Reference

### Functions

- def [checkpoint\\_subject\\_threaded](#) (obj)
- def [df\\_batch\\_consumer](#) (batch\_queue, identifier, port, server\_configuration, id\_queue, progress\_queue)
- def [df\\_batch\\_generator](#) (list\_of\_dfs, start, dataset\_dir, detect\_options, extract\_options, enroll\_options, options)
- def [df\\_batch\\_producer](#) (batch\_queue, identifier, progress\_position, list\_of\_dataframes, start, dataset\_dir, detect\_options, extract\_options, enroll\_options, options)
- def [df\\_row\\_generator](#) (rowiter, start, dataset\_dir, detect\_options, extract\_options, enroll\_options, options)
- def [enroll\\_call\\_threaded](#) (input)
- def [inner\\_pool\\_mapper](#) (batch\_obj, local\_pool, progress\_queue, port)
- def [parseSigsetEnrollOptions](#) (inputCommand=None)
- def [parseSigsetStatsOptions](#) (inputCommand=None)
- def [progress\\_consumer](#) (progress\_queue, progress\_position, total\_len, options)
- def [sigset\\_enroll](#) (input\_command=None)
- def [sigset\\_stats](#) (options=None, args=None)

### Variables

- [my\\_pool](#) = None
- [proc\\_number](#) = None
- [service\\_address\\_number](#) = None

### 6.33.1 Function Documentation

#### 6.33.1.1 [checkpoint\\_subject\\_threaded\(\)](#)

```
def briar.cli.sigset.checkpoint_subject_threaded (
    obj )
```

#### 6.33.1.2 [df\\_batch\\_consumer\(\)](#)

```
def briar.cli.sigset.df_batch_consumer (
    batch_queue,
    identifier,
    port,
    server_configuration,
    id_queue,
    progress_queue )
```

The `df_batch_consumer` function is a function that takes in a batch of dataframe objects, and processes them using the Briar API. The function uses the multiprocessing library to create multiple worker processes, each with their own connection to the Briar server. Each process then creates multiple threads which are used for processing image

```
:param batch_queue: Pass in the queue of batches to be processed
:param identifier: Identify the process
:param port: Determine which port to connect to
:param server_configuration: Specify the number of processes and threads per process
:param id_queue: Pass the connection id to the worker process
:param progress_queue: Send progress information to the main process
:return: A list of results
:doc-author: Joel Brogan
```

### 6.33.1.3 df\_batch\_generator()

```
def briar.cli.sigset.df_batch_generator (
    list_of_dfs,
    start,
    dataset_dir,
    detect_options,
    extract_options,
    enroll_options,
    options )
```

The `df_batch_generator` function is a generator that yields batches of dataframes to be processed by the multiprocessing pool. The function takes in a list of dataframes, and returns each one as an element in the generator. This allows for parallel processing using multiple cores on your machine.

```
:param list_of_dfs: Pass in the list of dataframes that are to be processed
:param start: Keep track of the current index in the list_of_dfs
:param dataset_dir: Specify the directory where the images are stored
:param detect_options: Specify the type of detection algorithm to use
:param extract_options: Specify the extraction algorithm to be used
:param enroll_options: Specify the enrollment options for each batch
:param options: Pass in the number of processes to use
:return: A generator that yields a tuple of the following form:
:doc-author: Joel Brogan
```

### 6.33.1.4 df\_batch\_producer()

```
def briar.cli.sigset.df_batch_producer (
    batch_queue,
    identifier,
    progress_position,
    list_of_dataframes,
    start,
    dataset_dir,
    detect_options,
    extract_options,
    enroll_options,
    options )
```

The `df_batch_producer` function is a generator that takes in a list of dataframes and yields batches of images. The function will yield batches until the end of the `list_of_dataframes` is reached.

```
:param batch_queue: Pass the batches to the main process
:param identifier: Identify the thread
:param progress_position: Keep track of the progress of the batch_producer function
:param list_of_dataframes: Store the list of dataframes that are to be processed
:param start: Determine the starting index of the dataframe
:param dataset_dir: Specify the directory where the dataset is located
:param detect_options: Specify the detection options
:param extract_options: Specify the extraction options
:param enroll_options: Specify the enrollment options
:param options: Pass the options for each of the functions
:return: A batch of dataframes
:doc-author: Joel Brogan
```



### 6.33.1.5 df\_row\_generator()

```
def briar.cli.sigset.df_row_generator (
    rowiter,
    start,
    dataset_dir,
    detect_options,
    extract_options,
    enroll_options,
    options )
```

The `df_row_generator` function is a generator that yields the row of the dataframe, the index of the row, and all other arguments passed to it. This allows us to use `multiprocessing.Pool`'s `map` function with multiple arguments.

```
:param rowiter: Iterate over the rows of a dataframe
:param start: Keep track of the row number
:param dataset_dir: Specify the directory where the dataset is located
:param detect_options: Set the detection options
:param extract_options: Pass the extract_options to the worker function
:param enroll_options: Pass the enroll_options to the function
:param options: Pass the options to the enroll_options function
:return: A generator that yields a list of arguments
:doc-author: Joel Brogan
```

### 6.33.1.6 enroll\_call\_threaded()

```
def briar.cli.sigset.enroll_call_threaded (
    input )
```

Enroll the given input data into the Briar database.

Parameters:

- `input` (tuple): A tuple containing the following elements:
  - `row` (dict): A dictionary containing the information about the input data.
  - `i` (int): Index of the input data.
  - `dataset_dir` (str): Directory path of the dataset.
  - `detect_options` (`briar_pb2.FrameDetectOptions`): Detection options for the enrollment process.
  - `extract_options` (`briar_pb2.TemplateExtractOptions`): Template extraction options for the enrollment process.
  - `enroll_options` (`briar_pb2.TemplateEnrollOptions`): Template enrollment options for the enrollment process.
  - `options` (`argparse.Namespace`): Command-line options for the enrollment process.

Returns:

None

### 6.33.1.7 inner\_pool\_mapper()

```
def briar.cli.sigset.inner_pool_mapper (
    batch_obj,
    local_pool,
    progress_queue,
    port )
```

The `inner_pool_mapper` function is a function that takes in a batch of data, and then maps the `enroll_call_thre`. The inner pool mapper is used to map the enroll call threaded function over batches of data. The inner pool ma

```
:param batch_obj: Pass the following parameters to the inner_pool_mapper function:
:param local_pool: Specify the number of threads to use for processing
:param progress_queue: Communicate progress back to the main process
:return: A list of results
:doc-author: Joel Brogan
```

#### 6.33.1.8 parseSigsetEnrollOptions()

```
def briar.cli.sigset.parseSigsetEnrollOptions (
    inputCommand = None )
```

The `parseSigsetEnrollOptions` function parses command line arguments for the `sigset-enroll` program.

```
:param inputCommand: Pass in the command line arguments
:return: The options and args
:doc-author: Joel Brogan, David Bolme, Trelent
```

#### 6.33.1.9 parseSigsetStatsOptions()

```
def briar.cli.sigset.parseSigsetStatsOptions (
    inputCommand = None )
```

Parse command line arguments.

#### 6.33.1.10 progress\_consumer()

```
def briar.cli.sigset.progress_consumer (
    progress_queue,
    progress_position,
    total_len,
    options )
```

The `progress_consumer` function is a consumer for the `progress_queue`. It takes in a queue, and an integer representing the position of the progress bar on screen. The function then creates a `BriarProgress` object with that position, and starts consuming from the queue until it receives `None` as an item in the queue (which indicates that all items have been consumed). For each item received from the queue, it increments its counter by 1.

```
:param progress_queue: Communicate with the progress_consumer function
:param progress_position: Determine where the progress bar should be placed on the screen
:param total_len: Set the maximum value of the progress bar
:param options: Pass the options to the progress bar
:return: A function that takes a progress queue,
:doc-author: Joel Brogan
```

### 6.33.1.11 sigset\_enroll()

```
def briar.cli.sigset.sigset_enroll (
    input_command = None )
```

The sigset\_enroll function is used to enroll a signature set into the Briar system.

```
:return: A list of enrollments
:doc-author: Joel Brogan, David Bolme, Trelent
```

### 6.33.1.12 sigset\_stats()

```
def briar.cli.sigset.sigset_stats (
    options = None,
    args = None )
```

The sigset\_stats function parses a sigset file and prints out some statistics about the contents.

```
:param options: Pass in the command line arguments
:param args: Pass in the command line arguments
:return: The number of unique names, the total media count,
:doc-author: Joel Brogan, David Bolme, Trelent
```

## 6.33.2 Variable Documentation

### 6.33.2.1 my\_pool

```
my_pool = None
```

### 6.33.2.2 proc\_number

```
proc_number = None
```

### 6.33.2.3 service\_address\_number

```
service_address_number = None
```

## 6.34 briar.cli.status Namespace Reference

### Functions

- def [get\\_service\\_configuration](#) (options=None, args=None, input\_command=None, ret=False)
- def [print\\_service\\_configuration](#) (options=None, args=None)
- def [status](#) (options=None, args=None, input\_command=None, ret=False)  
*Conects to the server and gets status information.*
- def [statusParseOptions](#) (inputCommand=None)  
*Generate options for getting status and parse command line arguments into them.*

### 6.34.1 Function Documentation

#### 6.34.1.1 [get\\_service\\_configuration\(\)](#)

```
def briar.cli.status.get_service_configuration (
    options = None,
    args = None,
    input_command = None,
    ret = False )
```

#### 6.34.1.2 [print\\_service\\_configuration\(\)](#)

```
def briar.cli.status.print_service_configuration (
    options = None,
    args = None )
```

#### 6.34.1.3 [status\(\)](#)

```
def briar.cli.status.status (
    options = None,
    args = None,
    input_command = None,
    ret = False )
```

Conects to the server and gets status information.

Print results.

#### Returns

: None - results are printed

#### 6.34.1.4 statusParseOptions()

```
def briar.cli.status.statusParseOptions (
    inputCommand = None )
```

Generate options for getting status and parse command line arguments into them.

##### Returns

: 2 element Tuple of (optparse.Values, list) containing the parsed options and parameters respectively

## 6.35 briar.cli.test Namespace Reference

### Classes

- class [BriarTest](#)
- class [BriarTestResult](#)
- class [DatabaseTest](#)
- class [DetectTest](#)
- class [EnrollTest](#)
- class [ExtractTest](#)
- class [Fore](#)

### Functions

- def [detection\\_output\\_tests](#) (detection\_obj\_loaded, testimage, return\_media)
- def [extraction\\_output\\_tests](#) (template\_obj\_loaded, testimage, return\_media)

### Variables

- bool [use\\_colorama](#) = False

### 6.35.1 Function Documentation

#### 6.35.1.1 detection\_output\_tests()

```
def briar.cli.test.detection_output_tests (
    detection_obj_loaded,
    testimage,
    return_media )
```

### 6.35.1.2 extraction\_output\_tests()

```
def briar.cli.test.extraction_output_tests (
    template_obj_loaded,
    testimage,
    return_media )
```

## 6.35.2 Variable Documentation

### 6.35.2.1 use\_colorama

```
bool use_colorama = False
```

## 6.36 briar.cli.track Namespace Reference

### Functions

- def [get\\_tracklet\\_path](#) (media\_file, options, i, modality=None, media\_id=None)
- def [save\\_tracklets](#) (media\_file, tracklets, options, i, verbose=False, modality=None, media\_id=None)
- def [track](#) (options=None, args=None, input\_command=None, ret=False)  
*Using the options specified in the command line, runs a detection on the specified files.*
- def [trackRequestConstructor](#) (briar\_pb2.BriarMedia media, briar\_pb2.BriarDurations durations, options\_↔ dict={}, det\_list\_list=None, str database\_name=None)

### Variables

- string [TRACKLET\\_FILE\\_EXT](#) = ".tracklet"

## 6.36.1 Function Documentation

### 6.36.1.1 get\_tracklet\_path()

```
def briar.cli.track.get_tracklet_path (
    media_file,
    options,
    i,
    modality = None,
    media_id = None )
```

### 6.36.1.2 save\_tracklets()

```
def briar.cli.track.save_tracklets (
    media_file,
    tracklets,
    options,
    i,
    verbose = False,
    modality = None,
    media_id = None )
```

### 6.36.1.3 track()

```
def briar.cli.track.track (
    options = None,
    args = None,
    input_command = None,
    ret = False )
```

Using the options specified in the command line, runs a detection on the specified files.

Writes results to disk to a location specified by the cmd arguments

#### Returns

: No return - Function writes results to disk

### 6.36.1.4 trackRequestConstructor()

```
def briar.cli.track.trackRequestConstructor (
    briar_pb2.BriarMedia media,
    briar_pb2.BriarDurations durations,
    options_dict = {},
    det_list_list = None,
    str database_name = None )
```

## 6.36.2 Variable Documentation

### 6.36.2.1 TRACKLET\_FILE\_EXT

```
string TRACKLET_FILE_EXT = ".tracklet"
```

## 6.37 briar.cli.verify Namespace Reference

### Functions

- def [addVerifyOptions](#) (parser)  
*Add options for verification to the parser.*
- def [save\\_verifications](#) (media\_file, reply, options, i, modality=None, media\_id=None)
- def [verify](#) (options=None, args=None, input\_command=None, ret=False)  
*Using the options specified in the command line, runs an extract on the specified files.*
- def [verify\\_options2proto](#) (options)
- def [verifyParseOptions](#) (inputCommand=None)  
*Generate options for running verifications and parse command line arguments into them.*

### Variables

- string [VERIFICATION\\_FILE\\_EXT](#) = ".verification"

### 6.37.1 Function Documentation

#### 6.37.1.1 addVerifyOptions()

```
def briar.cli.verify.addVerifyOptions (
    parser )
```

Add options for verification to the parser.

@type parser: optparse.OptionParser

#### Parameters

<i>parser</i>	A parser to modify in place by adding options
---------------	---

#### 6.37.1.2 save\_verifications()

```
def briar.cli.verify.save_verifications (
    media_file,
    reply,
    options,
    i,
    modality = None,
    media_id = None )
```



### 6.37.1.3 verify()

```
def briar.cli.verify.verify (
    options = None,
    args = None,
    input_command = None,
    ret = False )
```

Using the options specified in the command line, runs an extract on the specified files.

Writes results to disk to a location specified by the cmd arguments

#### Returns

: No return - Function writes results to disk

### 6.37.1.4 verify\_options2proto()

```
def briar.cli.verify.verify_options2proto (
    options )
```

### 6.37.1.5 verifyParseOptions()

```
def briar.cli.verify.verifyParseOptions (
    inputCommand = None )
```

Generate options for running verifications and parse command line arguments into them.

#### Returns

: 2 element Tuple of (optparse.Values, list) containing the parsed options and parameters respectively

## 6.37.2 Variable Documentation

### 6.37.2.1 VERIFICATION\_FILE\_EXT

```
string VERIFICATION_FILE_EXT = ".verification"
```

## 6.38 briar.cli.viz Namespace Reference

### Functions

- def [viz](#) ()  
*Using the options specified in the command line, runs visualization on the specified files.*
- def [vizParseOptions](#) ()  
*Generate options for running detections and parse command line arguments into them.*

### 6.38.1 Function Documentation

#### 6.38.1.1 viz()

```
def briar.cli.viz.viz ( )
```

Using the options specified in the command line, runs visualization on the specified files.

#### Returns

: No return - Function writes results to disk

#### 6.38.1.2 vizParseOptions()

```
def briar.cli.viz.vizParseOptions ( )
```

Generate options for running detections and parse command line arguments into them.

#### Returns

: 2 element Tuple of (optparse.Values, list) containing the parsed options and parameters respectively

## 6.39 briar.evaluation Namespace Reference

### Namespaces

- [full\\_evaluation](#)
- [stage1\\_probe\\_enroll](#)
- [stage2](#)
- [stage3\\_result\\_scoring](#)

### Functions

- def [runStages](#) (stages, report=False)

### 6.39.1 Function Documentation

#### 6.39.1.1 runStages()

```
def briar.evaluation.runStages (
    stages,
    report = False )
```

## 6.40 briar.evaluation.full\_evaluation Namespace Reference

### Classes

- class [Test000InitialConfig](#)
- class [Test001SigsetEnrollProbe](#)
- class [Test002ProbeDatabaseMerge](#)
- class [Test003SigsetEnrollMultiProbe](#)
- class [Test004MultiProbeDatabaseMerge](#)
- class [Test005SigsetEnrollGalleries](#)
- class [Test006GalleryDatabaseMerge](#)
- class [Test007SigsetScoreG1](#)
- class [Test008SigsetScoreGaitG1](#)
- class [Test009SigsetScoreFaceG1](#)
- class [Test010SigsetScoreWholeBodyG1](#)
- class [Test011SigsetScoreG2](#)
- class [Test012SigsetScoreGaitG2](#)
- class [Test013SigsetScoreFaceG2](#)
- class [Test014SigsetScoreWholeBodyG2](#)
- class [Test015SigsetBlendedScoreG1](#)
- class [Test016SigsetScoreBlendedGaitG1](#)
- class [Test017SigsetScoreBlendedFaceG1](#)
- class [Test018SigsetScoreBlendedWholeBodyG1](#)
- class [Test019SigsetBlendedScoreG2](#)
- class [Test020SigsetScoreBlendedGaitG2](#)
- class [Test021SigsetScoreBlendedFaceG2](#)
- class [Test022SigsetScoreBlendedWholeBodyG2](#)
- class [Test023MultiSigsetScoreG1](#)
- class [Test023SigsetSearchOutputFormatting](#)
- class [Test024MultiSigsetScoreGaitG1](#)
- class [Test024SigsetVerifyOutputFormatting](#)
- class [Test025MultiSigsetScoreFaceG1](#)
- class [Test026MultiSigsetScoreWholeBodyG1](#)
- class [Test027MultiSigsetScoreG2](#)
- class [Test028MultiSigsetScoreGaitG2](#)
- class [Test029MultiSigsetScoreFaceG2](#)
- class [Test030MultiSigsetScoreWholeBodyG2](#)
- class [Test031MultiSigsetBlendedScoreG1](#)
- class [Test032MultiSigsetScoreBlendedGaitG1](#)
- class [Test033MultiSigsetScoreBlendedFaceG1](#)
- class [Test034MultiSigsetScoreBlendedWholeBodyG1](#)
- class [Test035MultiSigsetBlendedScoreG2](#)
- class [Test036MultiSigsetScoreBlendedGaitG2](#)
- class [Test037MultiSigsetScoreBlendedFaceG2](#)
- class [Test038MultiSigsetScoreBlendedWholeBodyG2](#)

## Functions

- def `compute_search` (probe\_db\_name, gal\_db\_name, `probe_sigset_path`, output\_path, modality=None, blended=False)
- def `compute_verify` (probe\_db\_name, gal\_db\_name, `probe_sigset_path`, gal\_sigset\_path, output\_path, csv←\_path, modality=None, blended=False)
- def `get_info` (self, db\_name)
- def `get_multi_info` (self, base\_db\_name)
- def `merge_dbs` (self, db\_name)
- def `run_on_multi` (self, base\_db\_name, mapped\_function)
- def `runall` ()
- None `setUpClass_main` (cls)
- def `setUpModule` ()

## Variables

- string `args_string` = "--progress "
- string `blended_gallery_1_filename` = "sigsets\_gallery/Blended\_Gallery\_1.xml"
- string `blended_gallery_2_filename` = "sigsets\_gallery/Blended\_Gallery\_2.xml"
- `category`
- string `database_blended_gallery_1_name` = 'db\_eval\_phase2\_blended\_gallery\_1'
- string `database_blended_gallery_2_name` = 'db\_eval\_phase2\_blended\_gallery\_2'
- string `database_gallery_1_name` = 'db\_eval\_phase2\_gallery\_1'
- string `database_gallery_2_name` = 'db\_eval\_phase2\_gallery\_2'
- string `database_multi_probe_name` = 'db\_eval\_phase2\_multisubject\_probe'
- string `database_probe_name` = 'db\_eval\_phase2\_probe'
- `DATABASE_SUFFIX_FLAG` = os.getenv('BRIAR\_DATABASE\_SUFFIX\_FLAG')
- `DATASET_DIR` = os.getenv('BRIAR\_DATASET\_DIR')
- string `enroll_args` = "--auto-create-database "
- `EVAL_PHASE` = os.getenv('BRIAR\_EVAL\_PHASE')
- `EVALUATION_DIR` = os.getenv('BRIAR\_EVALUATION\_DIR')
- `EVALUATION_MULTISUBJECT_DIR` = os.getenv('BRIAR\_MULTISUBJECT\_EVALUATION\_DIR')
- `gallery1_blended_sigset`
- `gallery1_sigset`
- `gallery2_blended_sigset`
- `gallery2_sigset`
- `gallery_1_blended_sigset_path`
- string `gallery_1_filename` = "sigsets\_gallery/Gallery\_1.xml"
- `gallery_1_sigset_path`
- `gallery_2_blended_sigset_path`
- string `gallery_2_filename` = "sigsets\_gallery/Gallery\_2.xml"
- `gallery_2_sigset_path`
- `generate_report` = os.environ.get('REPORT', False)
- string `media_args` = "--no-save "
- `module`
- string `multisubject_probe_filename` = "sigsets\_multiperson/Probe\_BTS\_briar-rd\_multi.xml"
- int `number_of_partitions` = 1
- `OUTPUT_DIR` = os.getenv('BRIAR\_EVALUATION\_OUTPUT\_DIR')
- list `port_list` = []
- string `probe_filename` = "sigsets\_main/Probe\_BTS\_briar-rd\_ALL.xml"
- `probe_multisubject_sigset`
- `probe_multisubject_sigset_path`
- `probe_sigset`
- `probe_sigset_path`

- bool `requires_database_merge` = False
- `run_multisubject_evaluation`
- `RUN_STAGES` = `os.getenv('RUN_STAGES',None)`
- `sortTestMethodsUsing`
- list `stages_temp` = []
- `USE_SINGLE_SUBJECT` = `os.getenv('BRIAR_USE_SINGLE_SUBJECT')`
- `USES_FRONTEND_MERGING` = `os.getenv('BRIAR_USE_FRONTEND_MERGING')`

## 6.40.1 Function Documentation

### 6.40.1.1 `compute_search()`

```
def briar.evaluation.full_evaluation.compute_search (
    probe_db_name,
    gal_db_name,
    probe_sigset_path,
    output_path,
    modality = None,
    blended = False )
```

### 6.40.1.2 `compute_verify()`

```
def briar.evaluation.full_evaluation.compute_verify (
    probe_db_name,
    gal_db_name,
    probe_sigset_path,
    gal_sigset_path,
    output_path,
    csv_path,
    modality = None,
    blended = False )
```

### 6.40.1.3 `get_info()`

```
def briar.evaluation.full_evaluation.get_info (
    self,
    db_name )
```

#### 6.40.1.4 get\_multi\_info()

```
def briar.evaluation.full_evaluation.get_multi_info (
    self,
    base_db_name )
```

#### 6.40.1.5 merge\_dbs()

```
def briar.evaluation.full_evaluation.merge_dbs (
    self,
    db_name )
```

#### 6.40.1.6 run\_on\_multi()

```
def briar.evaluation.full_evaluation.run_on_multi (
    self,
    base_db_name,
    mapped_function )
```

#### 6.40.1.7 runall()

```
def briar.evaluation.full_evaluation.runall ( )
```

#### 6.40.1.8 setUpClass\_main()

```
None briar.evaluation.full_evaluation.setUpClass_main (
    cls )
```

#### 6.40.1.9 setUpModule()

```
def briar.evaluation.full_evaluation.setUpModule ( )
```

### 6.40.2 Variable Documentation

#### 6.40.2.1 args\_string

```
string args_string = " --progress "
```

#### 6.40.2.2 blended\_gallery\_1\_filename

```
string blended_gallery_1_filename = "sigsets_gallery/Blended_Gallery_1.xml"
```

#### 6.40.2.3 blended\_gallery\_2\_filename

```
string blended_gallery_2_filename = "sigsets_gallery/Blended_Gallery_2.xml"
```

#### 6.40.2.4 category

```
category
```

#### 6.40.2.5 database\_blended\_gallery\_1\_name

```
string database_blended_gallery_1_name = 'db_eval_phase2_blended_gallery_1'
```

#### 6.40.2.6 database\_blended\_gallery\_2\_name

```
string database_blended_gallery_2_name = 'db_eval_phase2_blended_gallery_2'
```

#### 6.40.2.7 database\_gallery\_1\_name

```
string database_gallery_1_name = 'db_eval_phase2_gallery_1'
```

#### 6.40.2.8 database\_gallery\_2\_name

```
string database_gallery_2_name = 'db_eval_phase2_gallery_2'
```

#### 6.40.2.9 database\_multi\_probe\_name

```
string database_multi_probe_name = 'db_eval_phase2_multisubject_probe'
```

#### 6.40.2.10 database\_probe\_name

```
string database_probe_name = 'db_eval_phase2_probe'
```

#### 6.40.2.11 DATABASE\_SUFFIX\_FLAG

```
string DATABASE_SUFFIX_FLAG = os.getenv('BRIAR_DATABASE_SUFFIX_FLAG')
```

#### 6.40.2.12 DATASET\_DIR

```
DATASET_DIR = os.getenv('BRIAR_DATASET_DIR')
```

#### 6.40.2.13 enroll\_args

```
string enroll_args = " --auto-create-database "
```

#### 6.40.2.14 EVAL\_PHASE

```
EVAL_PHASE = os.getenv('BRIAR_EVAL_PHASE')
```

#### 6.40.2.15 EVALUATION\_DIR

```
EVALUATION_DIR = os.getenv('BRIAR_EVALUATION_DIR')
```

#### 6.40.2.16 EVALUATION\_MULTISUBJECT\_DIR

```
EVALUATION_MULTISUBJECT_DIR = os.getenv('BRIAR_MULTISUBJECT_EVALUATION_DIR')
```



**6.40.2.17 gallery1\_blended\_sigset**

```
gallery1_blended_sigset
```

**6.40.2.18 gallery1\_sigset**

```
gallery1_sigset
```

**6.40.2.19 gallery2\_blended\_sigset**

```
gallery2_blended_sigset
```

**6.40.2.20 gallery2\_sigset**

```
gallery2_sigset
```

**6.40.2.21 gallery\_1\_blended\_sigset\_path**

```
gallery_1_blended_sigset_path
```

**6.40.2.22 gallery\_1\_filename**

```
string gallery_1_filename = "sigsets_gallery/Gallery_1.xml"
```

**6.40.2.23 gallery\_1\_sigset\_path**

```
gallery_1_sigset_path
```

**6.40.2.24 gallery\_2\_blended\_sigset\_path**

```
gallery_2_blended_sigset_path
```

#### 6.40.2.25 gallery\_2\_filename

```
string gallery_2_filename = "sigsets_gallery/Gallery_2.xml"
```

#### 6.40.2.26 gallery\_2\_sigset\_path

```
gallery_2_sigset_path
```

#### 6.40.2.27 generate\_report

```
generate_report = os.environ.get('REPORT', False)
```

#### 6.40.2.28 media\_args

```
string media_args = " --no-save "
```

#### 6.40.2.29 module

```
module
```

#### 6.40.2.30 multisubject\_probe\_filename

```
string multisubject_probe_filename = "sigsets_multiperson/Probe_BTS_briar-rd_multi.xml"
```

#### 6.40.2.31 number\_of\_partitions

```
int number_of_partitions = 1
```

#### 6.40.2.32 OUTPUT\_DIR

```
OUTPUT_DIR = os.getenv('BRIAR_EVALUATION_OUTPUT_DIR')
```

#### 6.40.2.33 port\_list

```
list port_list = []
```

#### 6.40.2.34 probe\_filename

```
string probe_filename = "sigsets_main/Probe_BTS_briar-rd_ALL.xml"
```

#### 6.40.2.35 probe\_multisubject\_sigset

```
probe_multisubject_sigset
```

#### 6.40.2.36 probe\_multisubject\_sigset\_path

```
probe_multisubject_sigset_path
```

#### 6.40.2.37 probe\_sigset

```
probe_sigset
```

#### 6.40.2.38 probe\_sigset\_path

```
probe_sigset_path
```

#### 6.40.2.39 requires\_database\_merge

```
bool requires_database_merge = False
```

#### 6.40.2.40 run\_multisubject\_evaluation

```
run_multisubject_evaluation
```

#### 6.40.2.41 RUN\_STAGES

```
list RUN_STAGES = os.getenv('RUN_STAGES',None)
```

#### 6.40.2.42 sortTestMethodsUsing

```
sortTestMethodsUsing
```

#### 6.40.2.43 stages\_temp

```
list stages_temp = []
```

#### 6.40.2.44 USE\_SINGLE\_SUBJECT

```
string USE_SINGLE_SUBJECT = os.getenv('BRIAR_USE_SINGLE_SUBJECT')
```

#### 6.40.2.45 USES\_FRONTEND\_MERGING

```
bool USES_FRONTEND_MERGING = os.getenv('BRIAR_USE_FRONTEND_MERGING')
```

### 6.41 briar.evaluation.stage1\_probe\_enroll Namespace Reference

#### Variables

- [generate\\_report](#) = os.environ.get('REPORT',False)
- [main](#) = unittest.TestProgram

#### 6.41.1 Variable Documentation

##### 6.41.1.1 generate\_report

```
generate_report = os.environ.get('REPORT',False)
```

### 6.41.1.2 main

```
main = unittest.TestProgram
```

## 6.42 briar.evaluation.stage2 Namespace Reference

### Variables

- `generate_report` = `os.environ.get('REPORT', False)`
- `main` = `unittest.TestProgram`

### 6.42.1 Variable Documentation

#### 6.42.1.1 generate\_report

```
generate_report = os.environ.get('REPORT', False)
```

#### 6.42.1.2 main

```
main = unittest.TestProgram
```

## 6.43 briar.evaluation.stage3\_result\_scoring Namespace Reference

### Variables

- `generate_report` = `os.environ.get('REPORT', False)`
- `main` = `unittest.TestProgram`

### 6.43.1 Variable Documentation

#### 6.43.1.1 generate\_report

```
generate_report = os.environ.get('REPORT', False)
```

### 6.43.1.2 main

```
main = unittest.TestProgram
```

## 6.44 briar.grpc\_json Namespace Reference

I couldn't find a good library to write gRPC/protobuf objects to disk so this is my own implementation of a generalized json converter for said objects.

### Classes

- class [GrpcDecoder](#)  
*Object which extends the JSONDecoded to allow it to read saved gRPC files.*
- class [GrpcEncoder](#)  
*Encoder class which extends the normal JSON encoder to allow the encoding of gRPC objects.*

### Functions

- def [dict\\_to\\_proto\\_obj](#) (obj\_dict, options=None)  
*Take the object dictionary, read the dict which is saved in in the 'class' key, and initialize it with values stored in the dictionary's key/value pairs.*
- def [load](#) (load\_path, options=None)  
*Load the json file at the given directory, reloading dictionaries with "\_\_class\_\_" fields into the specified objects and initializing them with values defined by key/value pairs within the dictionary.*
- def [proto\\_obj\\_to\\_dict](#) (obj, options=None)  
*Takes a general gRPC/protobuf object, eliminates the unnecessary fields, and stores the data in a dict.*
- def [save](#) (json\_obj, save\_path, options=None)  
*Save a list or dictionary containing protobuf classes to a json file.*

### Variables

- list [ATTRIB\\_IGNORE](#)

### 6.44.1 Detailed Description

I couldn't find a good library to write gRPC/protobuf objects to disk so this is my own implementation of a generalized json converter for said objects.

Good for writing the objects themselves but be aware that it will write any vectors/matrixes/images stored within the objects to disk so it may be a good idea to remove said data before converting to json if you care about performance in your implementations.

### 6.44.2 Function Documentation

#### 6.44.2.1 dict\_to\_proto\_obj()

```
def briar.grpc_json.dict_to_proto_obj (
    obj_dict,
    options = None )
```

Take the object dictionary, read the dict which is saved in in the 'class' key, and initialize it with values stored in the dictionary's key/value pairs.

## Parameters

<i>obj_dict</i>	dict: A dictionary specifically containing a ' <b>class</b> ' key/value pair storing the full module path to the object.
-----------------	--

## Returns

: A gRPC object defined by '**class**'

### 6.44.2.2 load()

```
def briar.grpc_json.load (
    load_path,
    options = None )
```

Load the json file at the given directory, reloading dictionaries with "`__class__`" fields into the specified objects and initializing them with values defined by key/value pairs within the dictionary.

## Parameters

<i>load_path</i>	str: Path to the json file to load
------------------	------------------------------------

## Returns

: The contents of the json file deserialized into the appropriate objects

### 6.44.2.3 proto\_obj\_to\_dict()

```
def briar.grpc_json.proto_obj_to_dict (
    obj,
    options = None )
```

Takes a general gRPC/protobuf object, eliminates the unnecessary fields, and stores the data in a dict.

Classes will be saved as dictionaries with a "`__class__`" attribute. This should be the full import path to the class within its module.

## Parameters

<i>obj</i>	Any gRPC object generated by protobuf files
------------	---

## Returns

: A dictionary representing the object

#### 6.44.2.4 save()

```
def briar.grpc_json.save (
    json_obj,
    save_path,
    options = None )
```

Save a list or dictionary containing protobuf classes to a json file.

##### Parameters

<i>json_obj</i>	list  dict: List or dict containing data to save
<i>save_path</i>	str: Path to the file to save

Returns: None

### 6.44.3 Variable Documentation

#### 6.44.3.1 ATTRIB\_IGNORE

```
list ATTRIB_IGNORE
```

##### Initial value:

```
1 = ['ByteSize', 'Clear', 'ClearExtension', 'ClearField', 'CopyFrom',
2     'DESCRIPTOR', 'DiscardUnknownFields', 'Extensions',
3     'FindInitializationErrors', 'FromString', 'HasExtension',
4     'HasField', 'IsInitialized', 'ListFields', 'MergeFrom',
5     'MergeFromString', 'ParseFromString', 'RegisterExtension',
6     'SerializePartialToString', 'SerializeToString',
7     'SetInParent', 'UnknownFields', 'WhichOneof',
8     '_CheckCalledFromGeneratedFile', '_SetListener',
9     '__deepcopy__', '__delattr__', '__dir__', '__doc__',
10    '_extensions_by_name', '_extensions_by_number', 'EnumTypeWrapper']
```

## 6.45 briar.media Namespace Reference

### Namespaces

- [VideoStream](#)
- [visualize](#)

### Classes

- class [BriarProgress](#)
- class [BriarVideoIterator](#)
- class [ImageIterator](#)
- class [MediaSetIterator](#)
- class [ThreadedVideoIterator](#)
- class [VideoIterator](#)



## Functions

- def [aenumerate](#) (asequence, start=0)
- def [decodeMedia](#) (media\_pb, newsource=None)  
*Convert protobuf media into a numpy array.*
- def [enroll\\_frames\\_iter](#) (database\_name, video, detect\_options=None, extract\_options=None, enroll\_options=None, det\_list\_list=None, whole\_image=False, request\_start=-1)  
*Iterates the paths in the media file list, loading them one by one and yielding grpc enroll requests.*
- def [enroll\\_frames\\_iter\\_async](#) (database\_name, video, detect\_options=None, extract\_options=None, enroll\_options=None, det\_list\_list=None, whole\_image=False, request\_start=-1)  
*Asynchronously iterates the paths in the media file list, loading them one by one and yielding grpc enroll requests.*
- def [file\\_iter](#) (list[str] media\_files, briar\_pb2.DetectionOptions clientoptions=None, dict options\_dict=None, str database\_name=None, bool verbose=False, int request\_start=-1, callable requestConstructor=None)
- def [frame\\_iter](#) (media\_iterator, clientoptions=None, dict options\_dict={}, database\_name=None, det\_list\_list=None, whole\_image=False, request\_start=-1, requestConstructor=None)
- def [ImageGenerator](#) (filepath, start=None, stop=None, unit=None, options=None)
- def [isFinalFrame](#) (request)
- def [single\\_frame\\_generate](#) (frame, frame\_num, itype, filepath, start\_frame, stop\_frame, fps, video\_length, clientoptions=None, dict options\_dict={}, database\_name=None, det\_list\_list=None, whole\_image=False, request\_start=-1, frame\_load\_time\_start=-1, file\_level\_client\_time\_end=-1, requestConstructor=None)

### 6.45.1 Function Documentation

#### 6.45.1.1 [aenumerate\(\)](#)

```
def briar.media.aenumerate (
    asequence,
    start = 0 )
```

Asynchronously enumerate an async iterator from a given start value

#### 6.45.1.2 [decodeMedia\(\)](#)

```
def briar.media.decodeMedia (
    media_pb,
    newsource = None )
```

Convert protobuf media into a numpy array.

##### Parameters

<code>media_pb</code>	<code>briar_pb2.BriarMedia</code>
-----------------------	-----------------------------------

return: `numpy.array`

### 6.45.1.3 enroll\_frames\_iter()

```
def briar.media.enroll_frames_iter (
    database_name,
    video,
    detect_options = None,
    extract_options = None,
    enroll_options = None,
    det_list_list = None,
    whole_image = False,
    request_start = -1 )
```

Iterates the paths in the media file list, loading them one by one and yielding grpc enroll requests.

@type database\_name: str

#### Parameters

<i>database_name</i>	Name of the database to enroll templates in
----------------------	---

@type video: an iterator that generates cv2 frames

#### Parameters

<i>media_files</i>	Paths to the media files to enroll from
--------------------	---

@type detect\_options: briar\_pb2.DetectionOptions

#### Parameters

<i>options</i>	Command line options in protobuf format which control detection functionality
----------------	---

@type options: briar\_pb2.ExtractOptions

#### Parameters

<i>options</i>	Command line options in protobuf format which control extraction functionality
----------------	--

@type options: briar\_pb2.EnrollOptions

#### Parameters

<i>options</i>	Command line options in protobuf format which control enrollment functionality
----------------	--

@type det\_list\_list: List of list of briar\_pb2.Detection

#### Parameters

<i>det_list_list</i>	If not None, it will contains 1 list of detections per media file
----------------------	---

@type whole\_image: boolean

#### Parameters

	Ignore detections and run an extract on the whole image
--	---

@type request\_start: int

#### Parameters

	timestamp of when the request started
--	---------------------------------------

@yield: briar\_service\_pb2.EnrollRequest

### 6.45.1.4 enroll\_frames\_iter\_async()

```
def briar.media.enroll_frames_iter_async (
    database_name,
    video,
    detect_options = None,
    extract_options = None,
    enroll_options = None,
    det_list_list = None,
    whole_image = False,
    request_start = -1 )
```

Asynchronously iterates the paths in the media file list, loading them one by one and yielding grpc enroll requests.

This method should provide better video load performance and better request generation performance.

@type database\_name: str

#### Parameters

<i>database_name</i>	Name of the database to enroll templates in
----------------------	---

@type video: an iterator that generates cv2 frames

#### Parameters

<i>media_files</i>	Paths to the media files to enroll from
--------------------	---

@type detect\_options: briar\_pb2.DetectionOptions

#### Parameters

<i>options</i>	Command line options in protobuf format which control detection functionality
----------------	---

@type options: briar\_pb2.ExtractOptions

**Parameters**

<i>options</i>	Command line options in protobuf format which control extraction functionality
----------------	--

@type options: briar\_pb2.EnrollOptions

**Parameters**

<i>options</i>	Command line options in protobuf format which control enrollment functionality
----------------	--

@type det\_list\_list: List of list of briar\_pb2.Detection

**Parameters**

<i>det_list_list</i>	If not None, it will contains 1 list of detections per media file
----------------------	---

@type whole\_image: boolean

**Parameters**

	Ignore detections and run an extract on the whole image
--	---

@type request\_start: int

**Parameters**

	timestamp of when the request started
--	---------------------------------------

@yield: briar\_service\_pb2.EnrollRequest

**6.45.1.5 file\_iter()**

```
def briar.media.file_iter (
    list[str] media_files,
    briar_pb2.DetectionOptions clientoptions = None,
    dict options_dict = None,
    str database_name = None,
    bool verbose = False,
    int request_start = -1,
    callable requestConstructor = None )
```

Iterates the paths in the media file list, loading them one by one and yielding grpc requests generated by <requestConstructor>

**Args:**

media\_files (list of str): Paths to the media files to enroll from.  
 clientoptions (briar\_pb2.DetectionOptions, optional): Command line options in protobuf format which control extraction functionality.  
 options\_dict (dict, optional): Additional options in dictionary format.  
 database\_name (str, optional): Name of the database.  
 verbose (bool, optional): If True, enables verbose output.  
 request\_start (int, optional): Starting index for the request.  
 requestConstructor (callable, optional): Function to construct the request.

**Yields:**

briar\_service\_pb2.DetectRequest: gRPC request generated by requestConstructor.

### 6.45.1.6 frame\_iter()

```
def briar.media.frame_iter (
    media_iterator,
    clientoptions = None,
    dict options_dict = {},
    database_name = None,
    det_list_list = None,
    whole_image = False,
    request_start = -1,
    requestConstructor = None )
```

The frame\_iter function is a generator that takes in an iterator of frames, and returns a request object for each frame. The request object contains the following information:

- The frame number (int)
- The type of media\_iterator (str)
- The filepath to the video/image sequence (str)
- A list containing all detections from previous iterations, if applicable. If not applicable, this will be None

```
:param media_iterator: Iterate through the frames of a video
:param clientoptions: Pass the client options to the request
:param options_dict : dict: Pass in the options dictionary from the client
:param database_name: Specify the database name
:param det_list_list: Pass in a list of detection lists
:param whole_image: Determine whether or not to send the whole image
:param request_start: Keep track of when the request was begun
:param requestConstructor: Create a request object
:return: A generator that yields a single frame request
:doc-author: Joel Brogan, BRIAR team, Trelent
```

### 6.45.1.7 ImageGenerator()

```
def briar.media.ImageGenerator (
    filepath,
    start = None,
    stop = None,
    unit = None,
    options = None )
```

The ImageGenerator function is a generator that yields frames from an image file.

```
:param filepath: Specify the location of the image
:param start: Set the starting frame of the video
:param stop: Stop the generator at a certain frame
:param unit: Determine the unit of time that is used for the start and stop parameters
:return: A single frame, so we need to wrap it in a loop
:doc-author: Joel Brogan, BRIAR team, Trelent
```

### 6.45.1.8 isFinalFrame()

```
def briar.media.isFinalFrame (
    request )
```

### 6.45.1.9 single\_frame\_generate()

```
def briar.media.single_frame_generate (
    frame,
    frame_num,
    ittype,
    filepath,
    start_frame,
    stop_frame,
    fps,
    video_length,
    clientoptions = None,
    dict options_dict = {},
    database_name = None,
    det_list_list = None,
    whole_image = False,
    request_start = -1,
    frame_load_time_start = -1,
    file_level_client_time_end = -1,
    requestConstructor = None )
```

The single\_frame\_generate function is used to generate a single frame of BriarMedia.

It takes in the following parameters:

- frame: The image or video frame that will be converted into BriarMedia.
- ittype: The type of iterator being used (ImageIterator, VideoIterator, MediaSetIterator). This is needed because the type of iterator determines the type of frame that is returned.
- filepath: A string containing the path and filename for the media being processed by BRIAR.

```
:param frame: Pass in the frame to be processed
:param frame_num: Keep track of the frame number
:param ittype: Determine the type of iterator used to generate frames
:param filepath: Indicate the path to the file that is being processed
:param start_frame: Indicate the starting frame of a video, and stop_frame is used to indicate the ending frame
:param stop_frame: Determine the last frame to be processed
:param video_length: Set the frame_count in the briarmedia
:param clientoptions: Pass the options to the client
:param options_dict : dict: Pass the options dictionary to the server
:param database_name: Specify the database to which the request should be sent
:param det_list_list: Pass the detection list from one frame to another
:param whole_image: Determine if the whole image is sent to briar or just a cropped version
:param request_start: Record the time at which the client api was called
:param file_level_client_time_end: Record the time at which the file-level operations of the briar client api
:param requestConstructor: Create a request object
:return: A request object, which is a briarrequest message
:doc-author: Joel Brogan, BRIAR team, Trellent
```

## 6.46 briar.media.VideoStream Namespace Reference

### Classes

- class [FileVideoStream\\_cv2](#)
- class [FileVideoStream\\_imageio](#)

## 6.47 briar.media.visualize Namespace Reference

### Classes

- class [match\\_matrix\\_visualizer](#)

### Functions

- def [decode\\_track](#) (tracklet, framenum=None, newsource=None)
- def [get\\_frame](#) (vidfile)
- def [playVideo](#) (vidfiles, titles=None, attributes=None, isvideo=True)
- def [update\\_annot](#) (ind, visualizer, pltloc, playvid=False)
- def [update\\_annot\\_filename\\_only](#) (ind, visualizer, pltloc)
- def [visualize\\_detection](#) (detection\_path)
- def [visualize\\_matches](#) (matches\_path)
- def [visualize\\_track](#) (track\_path, options)
- def [windowclick](#) (event, visualizer)
- def [windowhover](#) (event, visualizer, playvid=False)
- def [windowhover\\_filename\\_only](#) (event, visualizer)

### Variables

- string [fdir](#) = "/Users/2r6/Projects/briar/briar-api/media/test\_probe/clinton3.matches"
- list [files](#) = [os.path.join([fdir](#), f) for f in os.listdir([fdir](#))]

### 6.47.1 Function Documentation

#### 6.47.1.1 [decode\\_track\(\)](#)

```
def briar.media.visualize.decode_track (
    tracklet,
    framenum = None,
    newsource = None )
```

#### 6.47.1.2 [get\\_frame\(\)](#)

```
def briar.media.visualize.get_frame (
    vidfile )
```

#### 6.47.1.3 playVideo()

```
def briar.media.visualize.playVideo (
    vidfiles,
    titles = None,
    attributes = None,
    isvideo = True )
```

#### 6.47.1.4 update\_annot()

```
def briar.media.visualize.update_annot (
    ind,
    visualizer,
    pltloc,
    playvid = False )
```

#### 6.47.1.5 update\_annot\_filename\_only()

```
def briar.media.visualize.update_annot_filename_only (
    ind,
    visualizer,
    pltloc )
```

#### 6.47.1.6 visualize\_detection()

```
def briar.media.visualize.visualize_detection (
    detection_path )
```

#### 6.47.1.7 visualize\_matches()

```
def briar.media.visualize.visualize_matches (
    matches_path )
```

#### 6.47.1.8 visualize\_track()

```
def briar.media.visualize.visualize_track (
    track_path,
    options )
```



#### 6.47.1.9 windowclick()

```
def briar.media.visualize.windowclick (
    event,
    visualizer )
```

#### 6.47.1.10 windowhover()

```
def briar.media.visualize.windowhover (
    event,
    visualizer,
    playvid = False )
```

#### 6.47.1.11 windowhover\_filename\_only()

```
def briar.media.visualize.windowhover_filename_only (
    event,
    visualizer )
```

### 6.47.2 Variable Documentation

#### 6.47.2.1 fdir

```
string fdir = "/Users/2r6/Projects/briar/briar-api/media/test_probe/clinton3.matches"
```

#### 6.47.2.2 files

```
list files = [os.path.join(fdir, f) for f in os.listdir(fdir)]
```

## 6.48 briar.media\_converters Namespace Reference

Contained in this are functions for converting numpy arrays into various protobuf objects and back again since numpy arrays cannot be sent directly over gRPC.

## Functions

- def [attribute\\_find](#) (key, List[briar\_pb2.Attribute] attributes)
- def [attribute\\_proto2val](#) (briar\_pb2.Attribute attribute)
- def [attribute\\_retrieve](#) (briar\_pb2.Attribute attribute, return\_type=False)
- def [attribute\\_val2proto](#) (key, val, briar\_pb2.BriarDataType override\_type=None)
- def [check\\_if\\_delete\\_request](#) (srvc\_pb2.DatabaseInsertRequest request)
- def [check\\_if\\_delete\\_request\\_due\\_to\\_error](#) (srvc\_pb2.DatabaseInsertRequest request)
- def [get\\_entry\\_id\\_list](#) (srvc\_pb2.DatabaseInsertRequest request)
- def [image\\_cv2proto](#) (im, compression='uint8', quality=99, flip\_channels=True)  
*Convert a cv2 numpy array to a protobuf format.*
- def [image\\_file2proto](#) (imfile, path\_map={})
- def [image\\_np2proto](#) (im, compression='uint8', quality=99, flip\_channels=True)  
*Convert a numpy array to a protobuf format.*
- def [image\\_proto2cv](#) (pb\_data, flip\_channels=False)  
*Convert a protobuf BriarMedia image to a cv2 numpy array.*
- def [image\\_proto2np](#) (pb\_data, flip\_channels=True)  
*Convert a protobuf image to a numpy array.*
- def [matrix\\_np2proto](#) (mat)  
*Convert a numpy matrix into a BriarMatrix.*
- def [matrix\\_proto2np](#) (protomat)  
*Convert a protobuf matrix into a numpy matrix.*
- def [modality\\_proto2string](#) (modality)
- def [modality\\_string2proto](#) (modality)
- def [pathmap\\_path2remotepath](#) (path, path\_map, exclude\_cases\_containing\_folder=['mugshots'])
- def [pathmap\\_str2dict](#) (path\_map)
- def [subjectID\\_int2str](#) (subjectid)
- def [subjectID\\_str2int](#) (subjectid)
- def [subjectList\\_list2string](#) (subject\_list\_str, chomp=True)
- def [subjectList\\_string2list](#) (subject\_list)
- def [tracklet\\_list2proto](#) (track\_list)
- def [vector\\_np2proto](#) (vec)  
*Convert a 1 dimensional np array into a BriarVector.*
- def [vector\\_proto2np](#) (protovec)  
*Convert a protobuf vector into a numpy array.*
- def [video\\_file2proto](#) (vidfile, start, end, path\_map={})

## Variables

- dictionary [attribute\\_type\\_name\\_map](#) = {'int': 'ivalue', 'float': 'fvalue', 'string': 'text'}
- dictionary [modalityDict](#)
- dictionary [reverseModalityDict](#) = {[modalityDict](#)[k]: k for k in [modalityDict](#)}

### 6.48.1 Detailed Description

Contained in this are functions for converting numpy arrays into various protobuf objects and back again since numpy arrays cannot be sent directly over gRPC.

### 6.48.2 Function Documentation

### 6.48.2.1 attribute\_find()

```
def briar.media_converters.attribute_find (
    key,
    List[briar_pb2.Attribute] attributes )
```

Find an attribute by its key in a list of attributes.

Parameters:

key (Any): The key of the attribute to find.  
attributes (list[briar\_pb2.Attribute]): The list of attributes to search through.

Returns:

Any: The value of the attribute if found, otherwise None.

### 6.48.2.2 attribute\_proto2val()

```
def briar.media_converters.attribute_proto2val (
    briar_pb2.Attribute attribute )
```

The attribute\_proto2val function takes a briar\_pb2.Attribute object and returns the value of that attribute in

:param attribute: briar\_pb2.Attribute: Store the attribute  
:return: The value of the attribute  
:doc-author: Joel Brogan, BRIAR team, Trelent

### 6.48.2.3 attribute\_retrieve()

```
def briar.media_converters.attribute_retrieve (
    briar_pb2.Attribute attribute,
    return_type = False )
```

The attribute\_retrieve function takes in a briar\_pb2.Attribute object and returns the value of that attribute as a python object. The function also has an optional parameter, return\_type, which if set to True will return both the value of the attribute and its type as a tuple.

:param attribute: briar\_pb2.Attribute: Specify the attribute to retrieve  
:param return\_type: Determine if the type of the attribute should be returned as well  
:return: A tuple of the attribute value and type  
:doc-author: Joel Brogan, BRIAR team, Trelent

#### 6.48.2.4 attribute\_val2proto()

```
def briar.media_converters.attribute_val2proto (
    key,
    val,
    briar_pb2.BriarDataType  override_type = None )
```

The attribute\_val2proto function takes a key, value pair and converts it to a BriarAttribute protobuf. The function will attempt to determine the type of the value automatically, but you can override this by passing an override\_type parameter, then the val must be of that type or else it will raise an exception.

```
:param key: Identify the attribute
:param val: Set the value of the attribute
:param override_type :briar_pb2.BriarDataType: Force the type of the attribute
:return: A briar_pb2
:doc-author: Joel Brogan, BRIAR team, Trelent
```

#### 6.48.2.5 check\_if\_delete\_request()

```
def briar.media_converters.check_if_delete_request (
    srv_pb2.DatabaseInsertRequest request )
```

Checks if the delete request is valid.

Parameters:

request (srv\_pb2.DatabaseInsertRequest): The request containing the ids.

Returns:

bool: True if the delete request is valid, False otherwise.

#### 6.48.2.6 check\_if\_delete\_request\_due\_to\_error()

```
def briar.media_converters.check_if_delete_request_due_to_error (
    srv_pb2.DatabaseInsertRequest request )
```

Check if a delete request is due to an error.

Args:

request (srv\_pb2.DatabaseInsertRequest): The delete request to be checked.

Returns:

bool: True if the delete request is due to an error, False otherwise.

### 6.48.2.7 get\_entry\_id\_list()

```
def briar.media_converters.get_entry_id_list (
    srvc_pb2.DatabaseInsertRequest request )
```

Get Entry ID List

This method accepts a request object of type `srvc_pb2.DatabaseInsertRequest` and returns a list of entry IDs.

Parameters:

`request` (`srvc_pb2.DatabaseInsertRequest`): The request object containing entry IDs

Returns:

`list`: A list of entry IDs

### 6.48.2.8 image\_cv2proto()

```
def briar.media_converters.image_cv2proto (
    im,
    compression = 'uint8',
    quality = 99,
    flip_channels = True )
```

Convert a cv2 numpy array to a protobuf format.

Parameters

<i>img</i>	numpy.array: array containing the image to convert to BriarMedia
<i>compression</i>	str: What compression to use. Can be 'uint8', 'png', 'jpg'
<i>quality</i>	int: 0-100 How much do you want to mutilate the image in the name of saving memory?
<i>flip_channels</i>	boolean : Flips the channels dimension of a cv2-type numpy image. This could translate an image from RGB->BGR or vice-versa. Default is True.

return: `briar_pb2.BriarMedia`

### 6.48.2.9 image\_file2proto()

```
def briar.media_converters.image_file2proto (
    imfile,
    path_map = {} )
```

### 6.48.2.10 image\_np2proto()

```
def briar.media_converters.image_np2proto (
    im,
```

```
compression = 'uint8',  
quality = 99,  
flip_channels = True )
```

Convert a numpy array to a protobuf format.

## Parameters

<i>img</i>	numpy.array: array containing the image to convert to BriarMedia
<i>compression</i>	str: What compression to use. Can be 'uint8', 'png', 'jpg'
<i>quality</i>	int: 0-100 How much do you want to mutilate the image in the name of saving memory?
<i>flip_channels</i>	boolean : Flips the channels dimension of a cv2-type numpy image. This could translate an image from RGB->BGR or vice-versa. Default is True.

return: briar\_pb2.BriarMedia

**6.48.2.11 image\_proto2cv()**

```
def briar.media_converters.image_proto2cv (
    pb_data,
    flip_channels = False )
```

Convert a protobuf BriarMedia image to a cv2 numpy array.

## Parameters

<i>pb_data</i>	briar_pb2.BriarMedia: Protobuf object containing image data
<i>flip_channels</i>	boolean : Flips the channels dimension of a cv2-type numpy image. This could translate an image from RGB->BGR or vice-versa. Default is False.

## Returns

: numpy.array cv2 formatted np array containing image

**6.48.2.12 image\_proto2np()**

```
def briar.media_converters.image_proto2np (
    pb_data,
    flip_channels = True )
```

Convert a protobuf image to a numpy array.

## Parameters

<i>pb_data</i>	briar_pb2.BriarMedia: Protobuf object containing image data
----------------	---

## Returns

: np.array

#### 6.48.2.13 matrix\_np2proto()

```
def briar.media_converters.matrix_np2proto (
    mat )
```

Convert a numpy matrix into a BriarMatrix.

##### Parameters

<i>mat</i>	numpy.array: Matrix to convert
------------	--------------------------------

##### Returns

: briar\_pb2.BriarMatrix

#### 6.48.2.14 matrix\_proto2np()

```
def briar.media_converters.matrix_proto2np (
    protomat )
```

Convert a protobuf matrix into a numpy matrix.

##### Parameters

<i>protomat</i>	briar_pb2.BriarMatrix: Protobuf matrix to convert
-----------------	---

##### Returns

: numpy.array

#### 6.48.2.15 modality\_proto2string()

```
def briar.media_converters.modality_proto2string (
    modality )
```

The modality\_proto2string function takes a modality and returns the string representation of that modality.

```
:param modality: Determine which modality the data is from
:return: The string representation of the modality
:doc-author: Joel Brogan, BRIAR team, Trelent
```



### 6.48.2.16 modality\_string2proto()

```
def briar.media_converters.modality_string2proto (  
    modality )
```

The `modality_string2proto` function takes a string and returns the corresponding modality enum value.

```
:param modality: Determine the type of data that is being used  
:return: The modality of the image  
:doc-author: Joel Brogan, BRIAR team, Trelent
```

### 6.48.2.17 pathmap\_path2remotepath()

```
def briar.media_converters.pathmap_path2remotepath (  
    path,  
    path_map,  
    exclude_cases_containing_folder = ['mugshots'] )
```

Maps a local file path to a remote file path based on a given path map and excludes specific folders.

Args:

```
    path (str): The local file path to be mapped.  
    path_map (dict): A dictionary containing mappings from local folders to remote folders.  
    exclude_cases_containing_folder (list, optional): A list of folder names. If any of these folder names are  
    *shots'].
```

Returns:

```
    str: The mapped remote file path.
```

Raises:

```
    None
```

Example:

```
path = '/Users/johndoe/Documents/pictures/mugshots/2020/abc.jpg'  
path_map = {  
    'pictures': 'photo',  
    'mugshots': 'headshots'  
}  
exclude_cases_containing_folder = ['mugshots']  
  
remote_path = pathmap_path2remotepath(path, path_map, exclude_cases_containing_folder)  
print(remote_path)  
# Output: '/Users/johndoe/Documents/photo/headshots/2020/abc.jpg'
```

### 6.48.2.18 pathmap\_str2dict()

```
def briar.media_converters.pathmap_str2dict (  
    path_map )
```

The `pathmap_str2dict` function takes a string of the form  
'key:value,key2:value2'  
and returns a dictionary with keys and values as follows:  
{ key : value, key2 : value2 }.

```
:param path_map: Map the path of a file to another path  
:return: A dictionary mapping the paths in the  
:doc-author: Joel Brogan, BRIAR team, Trelent
```

#### 6.48.2.19 subjectID\_int2str()

```
def briar.media_converters.subjectID_int2str (
    subjectid )
```

The subjectID\_int2str function takes an integer subject ID and converts it to a string. The function is used in the process of creating a new subject ID for each participant. It ensures that all IDs are 5 digits long, with leading zeros if necessary.

```
:param subjectid: Convert the subjectid from an integer to a string
:return: A string of the subject id with a g in front and zeros to fill out the rest
:doc-author: Joel Brogan, BRIAR team, Trelent
```

#### 6.48.2.20 subjectID\_str2int()

```
def briar.media_converters.subjectID_str2int (
    subjectid )
```

The subjectID\_str2int function is for legacy systems that utilize only numbers as gallery entries. This function takes a string of the form &quot;G####&quot; and returns an integer of the form ####.

```
:param subjectid: Create a subjectid that is only numbers
:return: An integer value of the subjectid
:doc-author: Joel Brogan, BRIAR team, Trelent
```

#### 6.48.2.21 subjectList\_list2string()

```
def briar.media_converters.subjectList_list2string (
    subject_list_str,
    chomp = True )
```

Convert a list of subjects to a string representation.

Parameters:

subject\_list\_str (list): A list containing the subjects as strings.

Returns:

str: A string representation of the list of subjects, where each subject is separated by a comma.

Example:

```
>>> subjects = ['Math', 'Science', 'English']
>>> subjectList_list2string(subjects)
'Math,Science,English'
```

### 6.48.2.22 subjectList\_string2list()

```
def briar.media_converters.subjectList_string2list (
    subject_list )
```

Convert a comma-separated string of subjects to a list of strings.

Parameters:

subject\_list (str): A comma-separated string of subjects.

Returns:

List[str]: A list of subjects.

Example:

```
>>> subject_string2list("Math,Science,English")
['Math', 'Science', 'English']
```

### 6.48.2.23 tracklet\_list2proto()

```
def briar.media_converters.tracklet_list2proto (
    track_list )
```

The tracklet\_list2proto function takes a list of dictionaries and converts them into a tracklet proto. The dictionary must have the following keys:

x, y, width, height (all floats)

frame (int)

confidence (float)

:param track\_list: Create a tracklet proto

:return: A protobuf object

:doc-author: Joel Brogan, BRIAR team, Trelent

### 6.48.2.24 vector\_np2proto()

```
def briar.media_converters.vector_np2proto (
    vec )
```

Convert a 1 dimensional np array into a BriarVector.

Parameters

vec	numpy.array: Numpy array containing vector data
-----	---

Returns

: briar\_pb2.BriarVector

### 6.48.2.25 vector\_proto2np()

```
def briar.media_converters.vector_proto2np (
    protovec )
```

Convert a protobuf vector into a numpy array.

#### Parameters

<i>protovec</i>	briar_pb2.BriarVector: Protobuf object containing vector info
-----------------	---

#### Returns

: numpy.array

### 6.48.2.26 video\_file2proto()

```
def briar.media_converters.video_file2proto (
    vidfile,
    start,
    end,
    path_map = {} )
```

The video\_file2proto function takes in a video file, start frame, end frame and path\_map. The path\_map is used to map the local paths of the video files to their server side paths. This function returns a BriarMedia proto object with all of its fields filled out.

```
:param vidfile: Get the video file name
:param start: Specify the frame number of the first frame in a video
:param end: Determine the last frame of a video
:param path_map: Map the local path to the server path
:return: A briar_pb2
:doc-author: Joel Brogan, BRIAR team, Trelent
```

## 6.48.3 Variable Documentation

### 6.48.3.1 attribute\_type\_name\_map

```
dictionary attribute_type_name_map = {'int': 'ivalue', 'float': 'fvalue','string':'text'}
```

### 6.48.3.2 modalityDict

dictionary modalityDict

#### Initial value:

```
1 = {'whole_body': briar_pb2.WHOLE_BODY,
2     'wholeBody': briar_pb2.WHOLE_BODY,
3     'wholebody': briar_pb2.WHOLE_BODY,
4     'face': briar_pb2.FACE,
5     'gait': briar_pb2.GAIT,
6     'unspecified': briar_pb2.UNSPECIFIED}
```

### 6.48.3.3 reverseModalityDict

dictionary reverseModalityDict = {modalityDict[k]: k for k in modalityDict}

## 6.49 briar.sigset Namespace Reference

### Namespaces

- [parse](#)

## 6.50 briar.sigset.parse Namespace Reference

### Functions

- def [create\\_test\\_sigset](#) (str sigset\_probe\_file, str sigset\_gallery\_file, str base\_dir, output\_dir)
- def [expandTree](#) (root, level=0, spaces=3)
- def [parseBriarSigset](#) (filename)

### Variables

- [args](#) = sys.argv

### 6.50.1 Function Documentation

#### 6.50.1.1 create\_test\_sigset()

```
def briar.sigset.parse.create_test_sigset (
    str sigset_probe_file,
    str sigset_gallery_file,
    str base_dir,
    output_dir )
```

### 6.50.1.2 expandTree()

```
def briar.sigset.parse.expandTree (
    root,
    level = 0,
    spaces = 3 )
```

The expandTree function takes a root element and prints out the tag, length of padding, and level for each element. The function is recursive, so it will print out all elements in the tree.

```
:param root: Pass the root of the tree to be expanded
:param level: Keep track of the depth of the tree
:param spaces: Control the number of spaces used for indentation
:return: The name of each tag and the number of spaces between tags
:doc-author: Joel Brogan, BRIAR team, Trelent
```

### 6.50.1.3 parseBriarSigset()

```
def briar.sigset.parse.parseBriarSigset (
    filename )
```

The parseBriarSigset function parses a Briar XML sigset file and returns a pandas dataframe with the following columns:

- name - The name of the signature.
- subjectId - The ID of the subject who signed this signature.
- filepath - The path to where this sigmember is stored on disk. This may be relative or absolute, depending on the modality.
- modality - What type of media is used for this sigmember (e.g., 'online', 'offline'). See <https://briarsigset.org>

```
:param filename: Specify the path to the xml file
:return: A dataframe with the following columns:
:doc-author: Joel Brogan, BRIAR team, Joel Brogan, BRIAR team, Trelent
```

## 6.50.2 Variable Documentation

### 6.50.2.1 args

```
args = sys.argv
```

## 6.51 briar.timing Namespace Reference

### Functions

- def [end\\_duration](#) (reply)
- def [generate\\_progress](#) (frame\_id, media)
- def [loadDurationsFolder](#) (durations\_directory)
- def [parseDurations](#) (durationsperfile\_dictionary)
- def [print\\_duration](#) (name, duration)
- def [print\\_durations](#) (durations)
- def [save\\_durations](#) (media\_file, durations\_list, options, operation, modality=None)
- def [start\\_duration](#) (request, reply)
- def [timeElapsed](#) (duration)
- def [timestamp](#) ()

## Variables

- string `DURATION_FILE_EXT` = ".durations"

## 6.51.1 Function Documentation

### 6.51.1.1 `end_duration()`

```
def briar.timing.end_duration (
    reply )
```

### 6.51.1.2 `generate_progress()`

```
def briar.timing.generate_progress (
    frame_id,
    media )
```

### 6.51.1.3 `loadDurationsFolder()`

```
def briar.timing.loadDurationsFolder (
    durations_directory )
```

### 6.51.1.4 `parseDurations()`

```
def briar.timing.parseDurations (
    durationsperfile_dictionary )
```

### 6.51.1.5 `print_duration()`

```
def briar.timing.print_duration (
    name,
    duration )
```

#### 6.51.1.6 `print_durations()`

```
def briar.timing.print_durations (
    durations )
```

#### 6.51.1.7 `save_durations()`

```
def briar.timing.save_durations (
    media_file,
    durations_list,
    options,
    operation,
    modality = None )
```

#### 6.51.1.8 `start_duration()`

```
def briar.timing.start_duration (
    request,
    reply )
```

#### 6.51.1.9 `timeElapsed()`

```
def briar.timing.timeElapsed (
    duration )
```

#### 6.51.1.10 `timestamp()`

```
def briar.timing.timestamp ( )
```

### 6.51.2 Variable Documentation

#### 6.51.2.1 `DURATION_FILE_EXT`

```
string DURATION_FILE_EXT = ".durations"
```



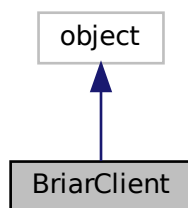
## Chapter 7

# Class Documentation

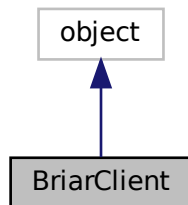
### 7.1 BriarClient Class Reference

Provide a client to a BRIAR service.

Inheritance diagram for BriarClient:



Collaboration diagram for BriarClient:



## Public Member Functions

- def `__init__` (self, `options`=None, reused\_channel=None, reused\_stub=None)  
*Initialize the client and connect it to the specified server.*
- def `database_create` (self, database\_name)  
*Creates an empty database of the given name.*
- def `database_insert` (self, database\_name, template\_list, template\_ids)  
*Database functions: Insert.*
- def `database_list_templates` (self, database\_name)  
*Lists the templates stored inside the given database name.*
- def `database_refresh` (self)
- def `database_remove_templates` (self, database\_name, template\_ids)  
*Remove templates matching the ids from the database.*
- def `database_retrieve` (self, database\_name, template\_ids)  
*Iteratively grab and return templates matching template\_ids from the database.*
- def `detect` (self, detect\_requests, `options`=None)  
*Detection functions.*
- def `enhance` (self, enhance\_requests, `options`=None)  
*Enhancement Functions.*
- def `enroll` (self, enroll\_iter)  
*Enroll images contained in the enroll iterator.*
- def `enroll_frames_iter` (self, database\_name, video, clientoptions=None, detect\_options=None, extract\_options=None, enroll\_options=None, det\_list\_list=None, whole\_image=False, request\_start=-1, as\_↵  
async=True, constructor=None)
- def `enroll_frames_iter_async` (self, database\_name, video, detect\_options=None, extract\_options=None, enroll\_options=None, det\_list\_list=None, whole\_image=False, request\_start=-1, yieldextra=False)
- def `extract` (self, extract\_iter)  
*Extract images contained in the extract iterator.*
- def `finalize` (self, database\_name)  
*Write the given database to disk on the server on which it is running.*
- def `get_database_names` (self)  
*Database functions.*
- def `get_service_configuration` (self)  
*Utility Functions:*
- def `get_status` (self, `options`=None)  
*Service Functions.*
- def `iter_over_async` (self, ait, loop)  
*Enroll Functions.*
- def `load_database` (self, database\_name)  
*Database functions: Load/Create.*
- def `print_verbose` (self, \*args)  
*Simple helper function to print only when the verbose client is given the verbose flag.*
- def `retrieve_req_iter` (self, database\_name, template\_ids)  
*Database functions: Retrieve.*
- def `search` (self, search\_iter)  
*Search Functions.*
- def `sync_enroll_frames_iter` (self, database\_name, video, detect\_options=None, extract\_options=None, enroll\_options=None, det\_list\_list=None, whole\_image=False, request\_start=-1)
- def `track` (self, track\_iter)  
*Tracking Functions.*
- def `verify` (self, flag, reference\_media=None, verification\_media=None, reference\_dets=None, verification\_↵  
\_dets=None, reference\_tmpls=None, verification\_tmpls=None)

*Verify Functions.*

- def `verify_file_iter` (self, reference\_media\_file, verify\_media\_file, client\_options, detect\_options=None, extract\_options=None, verify\_options=None, verify\_templates=None, det\_list\_list=None, whole\_image=False, request\_start=-1)

*Iterates the paths in the media file list, loading them one by one and yielding grpc verification requests.*

- def `verify_files` (self, reference\_file, verify\_file, client\_options, detect\_options=None, extract\_options=None, verify\_options=None, verify\_templates=None, det\_list\_list=None, whole\_image=False, request\_start=-1)

*Iterator which verifies images or videos from the given file paths, automatically creating and yielding enroll requests initialized from the read images.*

## Public Attributes

- `channel`
- `options`
- `port`
- `stub`

## Static Public Attributes

- string `DEFAULT_PORT` = "0.0.0.0:50051"

### 7.1.1 Detailed Description

Provide a client to a BRIAR service.

It defines and sends the messages which are sent to the connected server

### 7.1.2 Constructor & Destructor Documentation

#### 7.1.2.1 `__init__()`

```
def __init__ (
    self,
    options = None,
    reused_channel = None,
    reused_stub = None )
```

Initialize the client and connect it to the specified server.

Attempts a connection to localhost by default

#### Parameters

<code>options</code>	optparse.Values: Options which define the connection being established
----------------------	--

### 7.1.3 Member Function Documentation

#### 7.1.3.1 database\_create()

```
def database_create (
    self,
    database_name )
```

Creates an empty database of the given name.

##### Parameters

<i>database_name</i>	str: Name of database to create
----------------------	---------------------------------

##### Returns

: briar\_pb2.BriarDurations

#### 7.1.3.2 database\_insert()

```
def database_insert (
    self,
    database_name,
    template_list,
    template_ids )
```

Database functions: Insert.

Insert the given templates and ids into the database

**TODO remove template\_ids: is superfluous - template\_list is all that is needed**

**TODO Insert should automatically generate template ids for templates with no ids and return new ids**

##### Parameters

<i>database_name</i>	str: Name of the database to insert into
<i>template_list</i>	List of briar_pb2.Template
<i>template_list</i>	Templates to insert into database
<i>template_ids</i>	List of str
<i>template_ids</i>	IDs of templates being inserted

**Returns**

: 2 element Tuple (list of str, briar\_pb2.BriarDurations

**7.1.3.3 database\_list\_templates()**

```
def database_list_templates (
    self,
    database_name )
```

Lists the templates stored inside the given database name.

**Parameters**

<i>database_name</i>	str: Name of the database to get the templates from
----------------------	---

**Returns**

: 2 element tuple (list of str, briar\_pb2.Durations)

**7.1.3.4 database\_refresh()**

```
def database_refresh (
    self )
```

**7.1.3.5 database\_remove\_templates()**

```
def database_remove_templates (
    self,
    database_name,
    template_ids )
```

Remove templates matching the ids from the database.

**Parameters**

<i>database_name</i>	str: Name of the database to remove from
<i>template_ids</i>	list(str): Ids of the templates to remove

**Returns**

: briar\_pb2.BriarDurations

### 7.1.3.6 database\_retrieve()

```
def database_retrieve (
    self,
    database_name,
    template_ids )
```

Iteratively grab and return templates matching template\_ids from the database.

#### Parameters

<i>database_name</i>	str: Name of the database to retrieve from
<i>template_ids</i>	list(str): List of ids to retrieve from the database

#### Returns

: 2 element Tuple (briar\_pb2.Template, briar\_pb2.BriarDurations

### 7.1.3.7 detect()

```
def detect (
    self,
    detect_requests,
    options = None )
```

Detection functions.

Run detection on media contained in detect\_requests.

#### Parameters

<i>detect_requests</i>	Iterator yielding briar_service_pb2.DetectRequest: gRPC communication packet containing the data to run the detections on along with any additional options
<i>options</i>	optparse.Values
<i>options</i>	Additional options to feed to control the detect functions

yield: briar\_service\_pb2.DetectReply containing results

### 7.1.3.8 enhance()

```
def enhance (
    self,
    enhance_requests,
    options = None )
```

Enhancement Functions.

Run enhancement on media contained in enhance\_requests.

## Parameters

<i>enhance_requests</i>	Iterator yielding <code>briar_service_pb2.EnhanceRequest</code> : gRPC communication packet containing the data to run the detections on along with any additional options
<i>options</i>	<code>optparse.Values</code>
<i>options</i>	Additional options to feed to control the enhance functions

yield: `briar_service_pb2.EnhanceReply` containing results

**7.1.3.9 enroll()**

```
def enroll (
    self,
    enroll_iter )
```

Enroll images contained in the enroll iterator.

## Parameters

<i>enroll_iter</i>	Generator: Generator object which yields enroll requests
--------------------	--

## Returns

: `briar_service_pb2.ExtractReply`

**7.1.3.10 enroll\_frames\_iter()**

```
def enroll_frames_iter (
    self,
    database_name,
    video,
    clientoptions = None,
    detect_options = None,
    extract_options = None,
    enroll_options = None,
    det_list_list = None,
    whole_image = False,
    request_start = -1,
    as_async = True,
    constructor = None )
```

#### 7.1.3.11 enroll\_frames\_iter\_async()

```
def enroll_frames_iter_async (
    self,
    database_name,
    video,
    detect_options = None,
    extract_options = None,
    enroll_options = None,
    det_list_list = None,
    whole_image = False,
    request_start = -1,
    yieldextra = False )
```

#### 7.1.3.12 extract()

```
def extract (
    self,
    extract_iter )
```

Extract images contained in the extract iterator.

##### Parameters

<i>extract_iter</i>	Generator: Generator object which yields extract requests
---------------------	---

##### Returns

: `briar_service_pb2.ExtractReply`

#### 7.1.3.13 finalize()

```
def finalize (
    self,
    database_name )
```

Write the given database to disk on the server on which it is running.

##### Parameters

<i>database_name</i>	str: Name of the database to write to disk
----------------------	--

##### Returns

: `briar_pb2.durations`



**7.1.3.14 get\_database\_names()**

```
def get_database_names (
    self )
```

Database functions.

Gets a list of names from the service representing the databases human readable names

**Returns**

: List of str

**7.1.3.15 get\_service\_configuration()**

```
def get_service_configuration (
    self )
```

Utility Functions:

**7.1.3.16 get\_status()**

```
def get_status (
    self,
    options = None )
```

Service Functions.

Initialize the client and connect it to the specified server. Attempts a connection to localhost by default

**Parameters**

<i>options</i>	optparse.↔ Values:
----------------	-----------------------

**Returns**

: 5 element Tuple of str

**7.1.3.17 iter\_over\_async()**

```
def iter_over_async (
    self,
```

```
ait,  
loop )
```

Enroll Functions.

#### 7.1.3.18 load\_database()

```
def load_database (  
    self,  
    database_name )
```

Database functions: Load/Create.

Load the database from disk into memory

##### Parameters

<i>database_name</i>	str: Name of database to load
----------------------	-------------------------------

##### Returns

: 3 element Tuple(list of str, list of records, briar\_pb2.BriarDurations)

#### 7.1.3.19 print\_verbose()

```
def print_verbose (  
    self,  
    * args )
```

Simple helper function to print only when the verbose client is given the verbose flag.

##### Parameters

<i>args</i>	tuple(object): Arguments to get passed to print
-------------	---

##### Returns

: None - outputs to screen

#### 7.1.3.20 retrieve\_req\_iter()

```
def retrieve_req_iter (  
    self,
```

```
database_name,  
template_ids )
```

Database functions: Retrieve.

Generator which yields retrieve requests

#### Parameters

<i>database_name</i>	str: Name of the database to retrieve from
<i>template_ids</i>	list(str): The templates to retrieve

#### Returns

: briar\_service\_pb2.DatabaseRetrieveRequest

### 7.1.3.21 search()

```
def search (  
    self,  
    search_iter )
```

Search Functions.

Given a probe, search the database and return matches

#### Parameters

<i>database_name</i>	str: Name of the database to search
<i>media</i>	briar_pb2.BriarMedia: Media to pull probes from
<i>search_templates</i>	briar_pb2.Template: Probe Template
<i>detections</i>	briar_pb2.Detection: Detections to extract probe templates from
<i>flag</i>	int, briar_pb2.SearchFlags: Tells search whether to use auto-detect, extract detections, or provided templates,

#### Returns

: briar\_service\_pb2.SearchReply

### 7.1.3.22 sync\_enroll\_frames\_iter()

```
def sync_enroll_frames_iter (  
    self,  
    database_name,  
    video,
```

```

detect_options = None,
extract_options = None,
enroll_options = None,
det_list_list = None,
whole_image = False,
request_start = -1 )

```

### 7.1.3.23 track()

```

def track (
    self,
    track_iter )

```

Tracking Functions.

Track person instances contained in the tracking iterator

#### Parameters

<i>extract_iter</i>	Generator: Generator object which yields extract requests
---------------------	---

#### Returns

: briar\_service\_pb2.ExtractReply

### 7.1.3.24 verify()

```

def verify (
    self,
    flag,
    reference_media = None,
    verification_media = None,
    reference_dets = None,
    verification_dets = None,
    reference_tmpls = None,
    verification_tmpls = None )

```

Verify Functions.

Either takes probe templates or generates them from provided media and compares them against the 'verification' variables of a matching type. I.e. templates<->templates, media<->media, etc.

#### Parameters

<i>flag</i>	int briar_pb2.VerifyFlags: Tells the service details about the media it is going to verify perform detections on the images then extract and verify, should it use existing dets, or should it use the provided templates to verify
<i>reference_media</i>	briar_pb2.BriarMedia: This media is where the the probe templates will be pulled from

## Parameters

<i>verification_media</i>	briar_pb2.BriarMedia: This media is where the comparison templates will be pulled from
<i>reference_dets</i>	List of briar_pb2.Detection: These are the detections to extract probe templates from
<i>verification_dets</i>	List of briar_pb2.Detection: These are the detections to extract comparison templates from
<i>reference_tmpls</i>	list(briar_pb2.Template): Probe templates
<i>verification_tmpls</i>	list(briar_pb2.Template): Comparison Templates

## Returns

: 2 element Tuple (briar\_pb2.MatchSimilarities, briar\_pb2.BriarDurations)

TODO Currently, this can only compare templates<->templates, detections<->detections, media<->media  
 TODO Ideally any combination should be able to be compared, i.e. template<->media, detection<->template, etc.

## 7.1.3.25 verify\_file\_iter()

```
def verify_file_iter (
    self,
    reference_media_file,
    verify_media_file,
    client_options,
    detect_options = None,
    extract_options = None,
    verify_options = None,
    verify_templates = None,
    det_list_list = None,
    whole_image = False,
    request_start = -1 )
```

Iterates the paths in the media file list, loading them one by one and yielding grpc verification requests.

@type reference\_media\_file: File path as string

## Parameters

<i>reference_media_file</i>	File path as string to the media file that acts as the reference media to be verify against
-----------------------------	---

@type verify\_media\_file: File path as string

## Parameters

<i>verify_media_file</i>	File path as string to the media file that requires verification, should be of same length as reference_media_files
--------------------------	---

@type detect\_options: briar\_pb2.DetectionOptions

**Parameters**

<i>options</i>	Command line options in protobuf format which control detection functionality
----------------	---

@type extract\_options: briar\_pb2.ExtractOptions

**Parameters**

<i>options</i>	Command line options in protobuf format which control extraction functionality
----------------	--

@type earch\_options: briar\_pb2.SearchOptions

**Parameters**

<i>options</i>	Command line options in protobuf format which control search functionality
----------------	--

@type det\_list\_list: List of list of briar\_pb2.Detection

**Parameters**

<i>det_list_list</i>	If not None, it will contains 1 list of detections per media file
----------------------	---

@type whole\_image: boolean

**Parameters**

	Ignore detections and run an extract on the whole image
--	---

@yield: briar\_service\_pb2.EnrollRequest

**7.1.3.26 verify\_files()**

```
def verify_files (
    self,
    reference_file,
    verify_file,
    client_options,
    detect_options = None,
    extract_options = None,
    verify_options = None,
    verify_templates = None,
    det_list_list = None,
    whole_image = False,
    request_start = -1 )
```

Iterator which verifies images or videos from the given file paths, automatically creating and yielding enroll requests initialized from the read images.

@type reference\_media\_files: File path string of reference media

## Parameters

<i>reference_media_files</i>	File path string to the media that act as the reference media to be verify against
------------------------------	--

@type verify\_media\_files: File path string

## Parameters

<i>verify_media_files</i>	File path string of the media file that requires verification
---------------------------	---

@type options: optparse.Values

## Parameters

<i>options</i>	Command line options which control enrollment functionality
----------------	---

@type det\_list\_list: List of list of briar\_pb2.Detection

## Parameters

<i>det_list_list</i>	If not None, it will contains 1 list of detections per media file
----------------------	---

@type whole\_image: boolean

## Parameters

	Ignore detections and run an extract on the whole image
--	---

yield: briar\_service\_pb2.VerifyReply containing results

## 7.1.4 Member Data Documentation

### 7.1.4.1 channel

channel

### 7.1.4.2 DEFAULT\_PORT

```
string DEFAULT_PORT = "0.0.0.0:50051" [static]
```

#### 7.1.4.3 options

options

#### 7.1.4.4 port

port

#### 7.1.4.5 stub

stub

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

- [briar\\_client.py](#)

## 7.2 BriarMedia Class Reference

### Public Member Functions

- def [\\_\\_init\\_\\_](#) (self, media\_input="", [description](#)="", [datetime](#)=None, [metadata](#)=None)

### Public Attributes

- [channels](#)
- [datetime](#)
- [description](#)
- [fps](#)
- [height](#)
- [len](#)
- [metadata](#)
- [source](#)
- [width](#)

### Static Public Attributes

- [DATA\\_TYPES](#)
- list [IMAGE\\_FORMATS](#)
- list [VIDEO\\_FORMATS](#) = ['.avi', '.mp4', '.mov', '.m4v', '.ts']

#### 7.2.1 Constructor & Destructor Documentation



### 7.2.1.1 `__init__()`

```
def __init__ (
    self,
    media_input = "",
    description = "",
    datetime = None,
    metadata = None )
```

## 7.2.2 Member Data Documentation

### 7.2.2.1 `channels`

`channels`

### 7.2.2.2 `DATA_TYPES`

`DATA_TYPES` [static]

#### Initial value:

```
= dict(UINT8=0, UINT16=1, FLOAT32=2, URL=3, PNG=4, JPG=5, MJPG=6,
      H264=7, H265=8)
```

### 7.2.2.3 `datetime`

`datetime`

### 7.2.2.4 `description`

`description`

### 7.2.2.5 `fps`

`fps`

### 7.2.2.6 height

height

### 7.2.2.7 IMAGE\_FORMATS

```
list IMAGE_FORMATS [static]
```

**Initial value:**

```
= ['.bmp', 'dib', '.jpeg', '.jpg', '.jpe', '.jp2', '.png',  
   '.webp', '.pbm', '.pgm', '.ppm', '.pnm', '.sr',  
   '.ras', '.tiff', '.tif', '.exr', '.hdr', '.pic']
```

### 7.2.2.8 len

len

### 7.2.2.9 metadata

metadata

### 7.2.2.10 source

source

### 7.2.2.11 VIDEO\_FORMATS

```
list VIDEO_FORMATS = ['.avi', '.mp4', '.mov', '.m4v', '.ts'] [static]
```

### 7.2.2.12 width

width

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

- [briar\\_media.py](#)

## 7.3 BriarProgress Class Reference

### Public Member Functions

- def `__init__` (self, options, `desc`=None, `name`=None, `leave`=True, `position`=None)
- def `close` (self)
- def `refresh` (self)
- def `update` (self, current=1, total=-1)

### Public Attributes

- `desc`
- `enabled`
- `leave`
- `name`
- `pbar`
- `position`
- `prevstep`
- `tqdm`

### 7.3.1 Constructor & Destructor Documentation

#### 7.3.1.1 `__init__()`

```
def __init__ (
    self,
    options,
    desc = None,
    name = None,
    leave = True,
    position = None )
```

The `__init__` function is called when the class is instantiated.  
It sets up the progress bar and initializes some variables.

```
:param self: Represent the instance of the class
:param options: Determine if progress bars are enabled
:param desc: Set the description of the progress bar
:param name: Name the progress bar
:param leave: Determine whether the progress bar should be left on screen after completion
:param position: Set the position of the progress bar
:return: Nothing
:doc-author: Joel Brogan, BRIAR team, Trelent
```

### 7.3.2 Member Function Documentation

#### 7.3.2.1 close()

```
def close (
    self )
```

#### 7.3.2.2 refresh()

```
def refresh (
    self )
```

#### 7.3.2.3 update()

```
def update (
    self,
    current = 1,
    total = -1 )
```

### 7.3.3 Member Data Documentation

#### 7.3.3.1 desc

desc

#### 7.3.3.2 enabled

enabled

#### 7.3.3.3 leave

leave

#### 7.3.3.4 name

name

#### 7.3.3.5 pbar

pbar

#### 7.3.3.6 position

position

#### 7.3.3.7 prevstep

prevstep

#### 7.3.3.8 tqdm

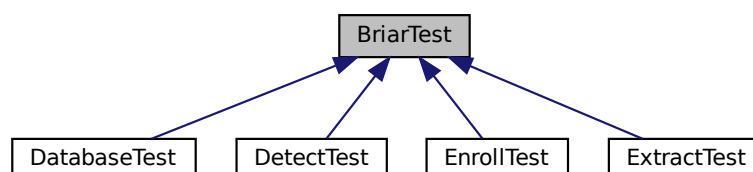
tqdm

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

- [media/\\_\\_init\\_\\_.py](#)

## 7.4 BriarTest Class Reference

Inheritance diagram for BriarTest:



## Public Member Functions

- def [\\_\\_init\\_\\_](#) (self)
- def [description](#) (self)
- def [run](#) (self)
- def [test](#) (self)

### 7.4.1 Constructor & Destructor Documentation

#### 7.4.1.1 [\\_\\_init\\_\\_\(\)](#)

```
def __init__ (  
    self )
```

### 7.4.2 Member Function Documentation

#### 7.4.2.1 [description\(\)](#)

```
def description (  
    self )
```

Reimplemented in [ExtractTest](#), and [DetectTest](#).

#### 7.4.2.2 [run\(\)](#)

```
def run (  
    self )
```

#### 7.4.2.3 [test\(\)](#)

```
def test (  
    self )
```

Reimplemented in [DatabaseTest](#), and [EnrollTest](#).

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

- [cli/test.py](#)

## 7.5 BriarTestResult Class Reference

### Public Member Functions

- `def __init__` (self, name, passed, reason=None, level=0)

### Public Attributes

- `level`
- `name`
- `passed`
- `reason`

### 7.5.1 Constructor & Destructor Documentation

#### 7.5.1.1 `__init__()`

```
def __init__ (
    self,
    name,
    passed,
    reason = None,
    level = 0 )
```

### 7.5.2 Member Data Documentation

#### 7.5.2.1 `level`

`level`

#### 7.5.2.2 `name`

`name`

#### 7.5.2.3 `passed`

`passed`

#### 7.5.2.4 reason

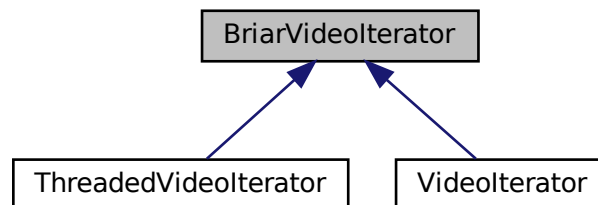
reason

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

- [cli/test.py](#)

## 7.6 BriarVideolterator Class Reference

Inheritance diagram for BriarVideolterator:



### Public Member Functions

- `def __aiter__(self)`
- `def __anext__(self)`
- `def __init__(self, filepath, start=None, stop=None, unit=None, debug_empty=False, options=None)`
- `def __iter__(self)`
- `def __len__(self)`
- `def __next__(self)`

#### 7.6.1 Constructor & Destructor Documentation



### 7.6.1.1 `__init__()`

```
def __init__ (
    self,
    filepath,
    start = None,
    stop = None,
    unit = None,
    debug_empty = False,
    options = None )
```

The `__init__` function is called when the class is instantiated. It sets up the instance of the class, and defines all its attributes. The `__init__` function takes in arguments that are passed to it by whoever creates an instance of this class, and assigns these arguments to self variables so they can be used throughout this object.

```
:param self: Represent the instance of the class
:param filepath: Specify the path to the video file
:param start: Specify the start frame of the video
:param stop: Set the last frame to be read from the video
:param unit: Specify the unit of start and stop, choices: frame, time in seconds, NA (defaults to full video)
:param debug_empty: specified for creating a debug video iterator object that passes empty frames for testing
:return: Nothing
:doc-author: Joel Brogan, BRIAR team, Trelent
```

Reimplemented in [ThreadedVideoIterator](#), and [VideoIterator](#).

## 7.6.2 Member Function Documentation

### 7.6.2.1 `__aiter__()`

```
def __aiter__ (
    self )
```

The `__aiter__` function is used to define an asynchronous iterator.

```
:param self: Refer to the current instance of a class
:return: The __iter__ function
:doc-author: Joel Brogan, BRIAR team, Trelent
```

Reimplemented in [ThreadedVideoIterator](#), and [VideoIterator](#).

### 7.6.2.2 `__anext__()`

```
def __anext__ (
    self )
```

The `__anext__` function is the asynchronous iterator protocol. It allows you to use `async` for loops, which are a lot more efficient than regular for loops. The `__anext__` function should return an awaitable object that resolves to the next item in your sequence.

```
:param self: Represent the instance of the class
:return: A frame, which is a numpy array
:doc-author: Joel Brogan, BRIAR team, Trelent
```

Reimplemented in [ThreadedVideoIterator](#), and [VideoIterator](#).

### 7.6.2.3 `__iter__()`

```
def __iter__ (
    self )
```

The `__iter__` function is called when an iterator is required for a container. This function should return a new iterator object that can iterate over all the objects in the container. For mappings, it should iterate over the keys of the container, and should also be made available as the method

```
:param self: Represent the instance of the class
:return: Self
:doc-author: Joel Brogan, BRIAR team, Trelent
```

Reimplemented in [ThreadedVideoIterator](#), and [VideoIterator](#).

### 7.6.2.4 `__len__()`

```
def __len__ (
    self )
```

The `__len__` function is used to determine the length of an object. For example, if you have a list with 5 items in it, calling `len(my_list)` will return 5. The `__len__` function is called when using the built-in `len()` function.

```
:param self: Allow an object to refer to itself
:return: The length of the iterator
:doc-author: Joel Brogan, BRIAR team, Trelent
```

Reimplemented in [ThreadedVideoIterator](#), and [VideoIterator](#).

### 7.6.2.5 `__next__()`

```
def __next__ (
    self )
```

The `__next__` function is called by the for loop to get each item from the iterator. The `__next__` function should raise a `StopIteration` exception when there are no more items in the container.

```
:param self: Represent the instance of the class
:return: A frame from the video
:doc-author: Joel Brogan, BRIAR team, Trelent
```

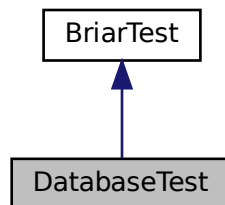
Reimplemented in [ThreadedVideoIterator](#), and [VideoIterator](#).

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

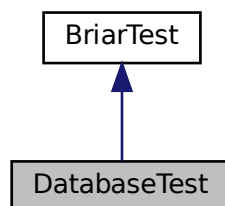
- [media/\\_\\_init\\_\\_.py](#)

## 7.7 DatabaseTest Class Reference

Inheritance diagram for DatabaseTest:



Collaboration diagram for DatabaseTest:



### Public Member Functions

- def [test](#) (self)

### 7.7.1 Member Function Documentation

#### 7.7.1.1 test()

```
def test (
    self )
```

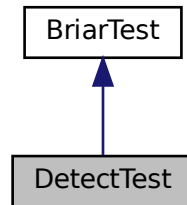
Reimplemented from [BriarTest](#).

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

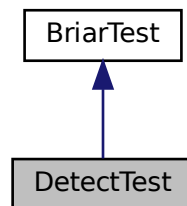
- cli/[test.py](#)

## 7.8 DetectTest Class Reference

Inheritance diagram for DetectTest:



Collaboration diagram for DetectTest:



### Public Member Functions

- def [description](#) (self)
- def [test\\_1\\_detection\\_image](#) (self, [testim\\_path](#)=None, [output\\_path](#)=None, return\_media=False)
- def [test\\_2\\_detection\\_image\\_output](#) (self, [testim\\_path](#)=None, [output\\_path](#)=None, return\_media=False)
- def [test\\_3\\_detection\\_image\\_withreturn](#) (self)
- def [test\\_4\\_detection\\_image\\_output\\_withreturn](#) (self)

### Public Attributes

- [detection\\_file\\_path](#)

### Static Public Attributes

- string [output\\_path](#) = `"/briar-integration-test-results"`
- string [testim\\_path](#) = `"testdata/BTS1/distractors/G00038/controlled/images_jpg/face/G00038_set2_face0_03_45_662fb70a.jpg"`

## 7.8.1 Member Function Documentation

### 7.8.1.1 description()

```
def description (
    self )
```

Reimplemented from [BriarTest](#).

### 7.8.1.2 test\_1\_detection\_image()

```
def test_1_detection_image (
    self,
    testim_path = None,
    output_path = None,
    return_media = False )
```

### 7.8.1.3 test\_2\_detection\_image\_output()

```
def test_2_detection_image_output (
    self,
    testim_path = None,
    output_path = None,
    return_media = False )
```

### 7.8.1.4 test\_3\_detection\_image\_withreturn()

```
def test_3_detection_image_withreturn (
    self )
```

### 7.8.1.5 test\_4\_detection\_image\_output\_withreturn()

```
def test_4_detection_image_output_withreturn (
    self )
```

## 7.8.2 Member Data Documentation

### 7.8.2.1 detection\_file\_path

```
detection_file_path
```

### 7.8.2.2 output\_path

```
string output_path = "./briar-integration-test-results" [static]
```

### 7.8.2.3 testim\_path

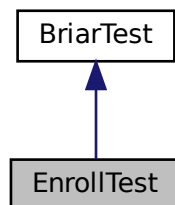
```
string testim_path = "testdata/BTS1/distractors/G00038/controlled/images_jpg/face/G00038_↵  
set2_face0_03_45_662fb70a.jpg" [static]
```

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

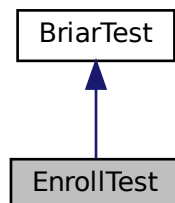
- cli/[test.py](#)

## 7.9 EnrollTest Class Reference

Inheritance diagram for EnrollTest:



Collaboration diagram for EnrollTest:



## Public Member Functions

- def [test](#) (self)

### 7.9.1 Member Function Documentation

#### 7.9.1.1 test()

```
def test (
    self )
```

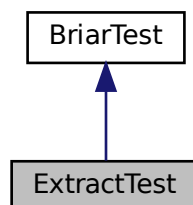
Reimplemented from [BriarTest](#).

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

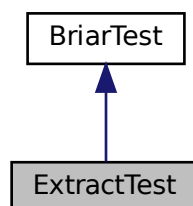
- cli/[test.py](#)

## 7.10 ExtractTest Class Reference

Inheritance diagram for ExtractTest:



Collaboration diagram for ExtractTest:



## Public Member Functions

- def [description](#) (self)
- def [test\\_1\\_extraction\\_image](#) (self, [testim\\_path](#)=None, [output\\_path](#)=None, return\_media=False)
- def [test\\_2\\_extraction\\_image\\_output](#) (self, [testim\\_path](#)=None, [output\\_path](#)=None, return\_media=False)

## Public Attributes

- [detection\\_file\\_path](#)
- [template\\_file\\_path](#)

## Static Public Attributes

- string [output\\_path](#) = `"/briar-integration-test-results"`
- string [testim\\_path](#) = `"testdata/BTS1/distractors/G00038/controlled/images_jpg/face/G00038_set2_face0_↵03_45_662fb70a.jpg"`

### 7.10.1 Member Function Documentation

#### 7.10.1.1 [description\(\)](#)

```
def description (  
    self )
```

Reimplemented from [BriarTest](#).

#### 7.10.1.2 [test\\_1\\_extraction\\_image\(\)](#)

```
def test_1_extraction_image (  
    self,  
    testim_path = None,  
    output_path = None,  
    return_media = False )
```

#### 7.10.1.3 [test\\_2\\_extraction\\_image\\_output\(\)](#)

```
def test_2_extraction_image_output (  
    self,  
    testim_path = None,  
    output_path = None,  
    return_media = False )
```



## 7.10.2 Member Data Documentation

### 7.10.2.1 detection\_file\_path

detection\_file\_path

### 7.10.2.2 output\_path

```
string output_path = "./briar-integration-test-results" [static]
```

### 7.10.2.3 template\_file\_path

template\_file\_path

### 7.10.2.4 testim\_path

```
string testim_path = "testdata/BTS1/distractors/G00038/controlled/images_jpg/face/G00038_↵  
set2_face0_03_45_662fb70a.jpg" [static]
```

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

- cli/[test.py](#)

## 7.11 FileVideoStream\_cv2 Class Reference

### Public Member Functions

- def [\\_\\_init\\_\\_](#) (self, path, [transform](#)=None, queue\_size=60 \*3, [options](#)=None)
- def [get\\_fps](#) (self)
- def [get\\_height](#) (self)
- def [get\\_length](#) (self)
- def [get\\_position](#) (self)
- def [get\\_width](#) (self)
- def [is\\_open](#) (self)
- def [more](#) (self)
- def [read](#) (self)
- def [running](#) (self)
- def [scrub\\_to](#) (self, index)
- def [start](#) (self)
- def [stop](#) (self)
- def [update](#) (self)

## Public Attributes

- [options](#)
- [Q](#)
- [stopped](#)
- [stream](#)
- [thread](#)
- [transform](#)

## 7.11.1 Constructor & Destructor Documentation

### 7.11.1.1 `__init__()`

```
def __init__ (
    self,
    path,
    transform = None,
    queue_size = 60*3,
    options = None )
```

## 7.11.2 Member Function Documentation

### 7.11.2.1 `get_fps()`

```
def get_fps (
    self )
```

### 7.11.2.2 `get_height()`

```
def get_height (
    self )
```

### 7.11.2.3 `get_length()`

```
def get_length (
    self )
```

#### 7.11.2.4 get\_position()

```
def get_position (
    self )
```

#### 7.11.2.5 get\_width()

```
def get_width (
    self )
```

#### 7.11.2.6 is\_open()

```
def is_open (
    self )
```

#### 7.11.2.7 more()

```
def more (
    self )
```

#### 7.11.2.8 read()

```
def read (
    self )
```

#### 7.11.2.9 running()

```
def running (
    self )
```

#### 7.11.2.10 scrub\_to()

```
def scrub_to (
    self,
    index )
```

#### 7.11.2.11 start()

```
def start (
    self )
```

#### 7.11.2.12 stop()

```
def stop (
    self )
```

#### 7.11.2.13 update()

```
def update (
    self )
```

### 7.11.3 Member Data Documentation

#### 7.11.3.1 options

options

#### 7.11.3.2 Q

Q

#### 7.11.3.3 stopped

stopped

#### 7.11.3.4 stream

stream

### 7.11.3.5 thread

thread

### 7.11.3.6 transform

transform

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

- [media/VideoStream.py](#)

## 7.12 FileVideoStream\_imageio Class Reference

### Public Member Functions

- def `__init__` (self, path, [transform](#)=None, queue\_size=60 \*3, [options](#)=None)
- def [get\\_fps](#) (self)
- def [get\\_height](#) (self)
- def [get\\_length](#) (self)
- def [get\\_position](#) (self)
- def [get\\_width](#) (self)
- def [is\\_open](#) (self)
- def [more](#) (self)
- def [read](#) (self)
- def [running](#) (self)
- def [scrub\\_to](#) (self, index)
- def [start](#) (self)
- def [stop](#) (self)
- def [update](#) (self)

### Public Attributes

- [backend](#)
- [fps](#)
- [options](#)
- [Q](#)
- [stopped](#)
- [stream](#)
- [thread](#)
- [transform](#)

### 7.12.1 Constructor & Destructor Documentation

#### 7.12.1.1 `__init__()`

```
def __init__ (
    self,
    path,
    transform = None,
    queue_size = 60 * 3,
    options = None )
```

### 7.12.2 Member Function Documentation

#### 7.12.2.1 `get_fps()`

```
def get_fps (
    self )
```

#### 7.12.2.2 `get_height()`

```
def get_height (
    self )
```

#### 7.12.2.3 `get_length()`

```
def get_length (
    self )
```

#### 7.12.2.4 `get_position()`

```
def get_position (
    self )
```

#### 7.12.2.5 `get_width()`

```
def get_width (
    self )
```

### 7.12.2.6 is\_open()

```
def is_open (
    self )
```

### 7.12.2.7 more()

```
def more (
    self )
```

### 7.12.2.8 read()

```
def read (
    self )
```

### 7.12.2.9 running()

```
def running (
    self )
```

### 7.12.2.10 scrub\_to()

```
def scrub_to (
    self,
    index )
```

### 7.12.2.11 start()

```
def start (
    self )
```

### 7.12.2.12 stop()

```
def stop (
    self )
```

#### 7.12.2.13 update()

```
def update (
    self )
```

### 7.12.3 Member Data Documentation

#### 7.12.3.1 backend

backend

#### 7.12.3.2 fps

fps

#### 7.12.3.3 options

options

#### 7.12.3.4 Q

Q

#### 7.12.3.5 stopped

stopped

#### 7.12.3.6 stream

stream



### 7.12.3.7 thread

```
thread
```

### 7.12.3.8 transform

```
transform
```

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

- [media/VideoStream.py](#)

## 7.13 Fore Class Reference

### Static Public Attributes

- string [BLUE](#) = "
- string [GREEN](#) = "
- string [RED](#) = "
- string [RESET](#) = "
- string [YELLOW](#) = "

### 7.13.1 Member Data Documentation

#### 7.13.1.1 BLUE

```
string BLUE = '' [static]
```

#### 7.13.1.2 GREEN

```
string GREEN = '' [static]
```

#### 7.13.1.3 RED

```
string RED = '' [static]
```

#### 7.13.1.4 RESET

```
string RESET = '' [static]
```

#### 7.13.1.5 YELLOW

```
string YELLOW = '' [static]
```

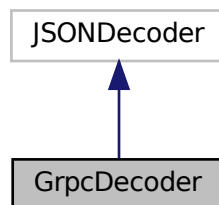
The documentation for this class was generated from the following file:

- [cli/test.py](#)

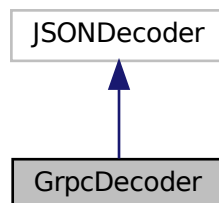
## 7.14 GrpcDecoder Class Reference

Object which extends the JSONDecoded to allow it to read saved gRPC files.

Inheritance diagram for GrpcDecoder:



Collaboration diagram for GrpcDecoder:



## Public Member Functions

- def `__init__` (self, `options`)
- def `default` (self, `obj`)

*Takes the given object and convert it into a gRPC object if its a dictionary.*

## Public Attributes

- `options`

### 7.14.1 Detailed Description

Object which extends the JSONDecoded to allow it to read saved gRPC files.

Applied as a hook in the json load function. Inherits from json.JSONDecoder

### 7.14.2 Constructor & Destructor Documentation

#### 7.14.2.1 `__init__()`

```
def __init__ (
    self,
    options )
```

### 7.14.3 Member Function Documentation

#### 7.14.3.1 `default()`

```
def default (
    self,
    obj )
```

Takes the given object and convert it into a gRPC object if its a dictionary.

#### Parameters

<code>obj</code>	object  dict: Dictionary which represents an object.
------------------	--

Returns

: object

## 7.14.4 Member Data Documentation

### 7.14.4.1 options

`options`

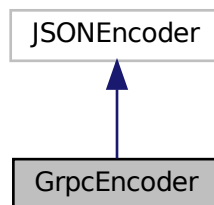
The documentation for this class was generated from the following file:

- [grpc\\_json.py](#)

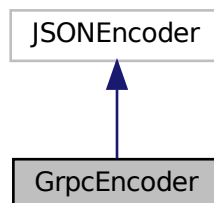
## 7.15 GrpcEncoder Class Reference

Encoder class which extends the normal JSON encoder to allow the encoding of gRPC objects.

Inheritance diagram for GrpcEncoder:



Collaboration diagram for GrpcEncoder:



## Public Member Functions

- def `__init__` (self, `options`=None)
- def `default` (self, obj)

*Json hook function to convert gRPC objects into a json-serializable object.*

## Public Attributes

- `options`

### 7.15.1 Detailed Description

Encoder class which extends the normal JSON encoder to allow the encoding of gRPC objects.

Inherits from `json.JSONEncoder`

### 7.15.2 Constructor & Destructor Documentation

#### 7.15.2.1 `__init__()`

```
def __init__ (
    self,
    options = None )
```

### 7.15.3 Member Function Documentation

#### 7.15.3.1 `default()`

```
def default (
    self,
    obj )
```

Json hook function to convert gRPC objects into a json-serializable object.

#### Parameters

<i>obj</i>	Any: General object to convert to a json-serializable object
------------	--

**Returns**

: Json-Serializable object

## 7.15.4 Member Data Documentation

### 7.15.4.1 options

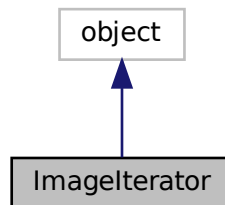
`options`

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

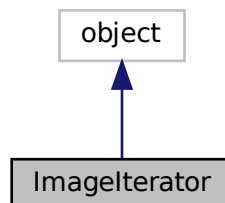
- [grpc\\_json.py](#)

## 7.16 Imageliterator Class Reference

Inheritance diagram for Imageliterator:



Collaboration diagram for Imageliterator:



## Public Member Functions

- def `__init__` (self, `filepath`, start=None, stop=None, unit=None, `debug_empty`=False, options=None)
- def `__iter__` (self)
- def `__len__` (self)
- def `__next__` (self)

## Public Attributes

- `debug_empty`
- `filepath`
- `fps`
- `frame`
- `frame_count`
- `frame_height`
- `frame_width`
- `i`
- `isOpened`
- `length`
- `msec`
- `pos`
- `processed`
- `start_frame`
- `stop_frame`

### 7.16.1 Constructor & Destructor Documentation

#### 7.16.1.1 `__init__()`

```
def __init__ (
    self,
    filepath,
    start = None,
    stop = None,
    unit = None,
    debug_empty = False,
    options = None )
```

The `__init__` function is called when the class is instantiated.  
It sets up the object with all of its initial values.

```
:param self: Represent the instance of the class
:param filepath: Specify the path to the image file
:param start: Set the start frame of the video
:param stop: Specify the last frame to be read
:param unit: Specify the unit of time for the start and stop parameters
:return: The following:
:doc-author: Joel Brogan, BRIAR team, Trelent
```

## 7.16.2 Member Function Documentation

### 7.16.2.1 `__iter__()`

```
def __iter__ (
    self )
```

The `__iter__` function is called when an iterator object is created for the class. This function should return a new iterator object that can iterate over all the objects in the class. For example, list, tuple or string classes have this method defined that allows them to be iterated over with

```
:param self: Access the attributes and methods of the class
:return: The object itself
:doc-author: Joel Brogan, BRIAR team, Trelent
```

### 7.16.2.2 `__len__()`

```
def __len__ (
    self )
```

The `__len__` function is used to determine the length of an object. For example, `len(s)` returns the number of items in `s`. The built-in function `len()` calls `s.__len__()`.

```
:param self: Refer to the instance of the class
:return: The length of the list
:doc-author: Joel Brogan, BRIAR team, Trelent
```

### 7.16.2.3 `__next__()`

```
def __next__ (
    self )
```

The `__next__` function is called by the Python interpreter to fetch the next value from an iterator. It should raise `StopIteration` when there are no more values to fetch.

```
:param self: Represent the instance of the class
:return: The next frame in the video
:doc-author: Joel Brogan, BRIAR team, Trelent
```

## 7.16.3 Member Data Documentation



### 7.16.3.1 debug\_empty

debug\_empty

### 7.16.3.2 filepath

filepath

### 7.16.3.3 fps

fps

### 7.16.3.4 frame

frame

### 7.16.3.5 frame\_count

frame\_count

### 7.16.3.6 frame\_height

frame\_height

### 7.16.3.7 frame\_width

frame\_width

### 7.16.3.8 i

i

**7.16.3.9 isOpened**

isOpened

**7.16.3.10 length**

length

**7.16.3.11 msec**

msec

**7.16.3.12 pos**

pos

**7.16.3.13 processed**

processed

**7.16.3.14 start\_frame**

start\_frame

**7.16.3.15 stop\_frame**

stop\_frame

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

- [media/\\_\\_\\_init\\_\\_\\_py](#)

## 7.17 match\_matrix\_visualizer Class Reference

### Public Member Functions

- `def __init__` (self, [searchReply](#), `probedbname`, `gallerydbname`)
- `def showmat\_interactive` (self)

### Public Attributes

- [annotations](#)
- [ax](#)
- [fig](#)
- [figures](#)
- [gallerydb\\_name](#)
- [gt](#)
- [mat](#)
- [prevx](#)
- [prevy](#)
- [probedb\\_name](#)
- [searchReply](#)
- [xlabs](#)
- [xsources](#)
- [ylabs](#)
- [ysources](#)

### 7.17.1 Constructor & Destructor Documentation

#### 7.17.1.1 \_\_init\_\_()

```
def __init__ (
    self,
    searchReply,
    probedbname,
    gallerydbname )
```

### 7.17.2 Member Function Documentation

#### 7.17.2.1 showmat\_interactive()

```
def showmat_interactive (
    self )
```

### 7.17.3 Member Data Documentation

#### 7.17.3.1 annotations

annotations

#### 7.17.3.2 ax

ax

#### 7.17.3.3 fig

fig

#### 7.17.3.4 figures

figures

#### 7.17.3.5 gallerydb\_name

gallerydb\_name

#### 7.17.3.6 gt

gt

#### 7.17.3.7 mat

mat

### 7.17.3.8 prevx

prevx

### 7.17.3.9 prevy

prevy

### 7.17.3.10 probedb\_name

probedb\_name

### 7.17.3.11 searchReply

searchReply

### 7.17.3.12 xlabs

xlabs

### 7.17.3.13 xsources

xsources

### 7.17.3.14 ylabs

ylabs

### 7.17.3.15 ysources

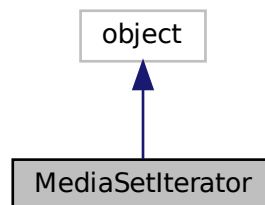
ysources

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

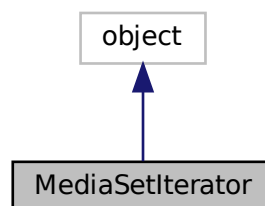
- [media/visualize.py](#)

## 7.18 MediaSetIterator Class Reference

Inheritance diagram for MediaSetIterator:



Collaboration diagram for MediaSetIterator:



### Public Member Functions

- def [\\_\\_init\\_\\_](#) (self, [filepaths](#), [start\\_frames](#), [stop\\_frames](#), unit=None, path\_map={}, options=None)
- def [\\_\\_iter\\_\\_](#) (self)
- def [\\_\\_len\\_\\_](#) (self)
- def [\\_\\_next\\_\\_](#) (self)

## Public Attributes

- [filepaths](#)
- [i](#)
- [isOpen](#)
- [media\\_set](#)
- [processed](#)
- [start\\_frame](#)
- [start\\_frames](#)
- [stop\\_frame](#)
- [stop\\_frames](#)

## 7.18.1 Constructor & Destructor Documentation

### 7.18.1.1 `__init__()`

```
def __init__ (
    self,
    filepaths,
    start_frames,
    stop_frames,
    unit = None,
    path_map = {},
    options = None )
```

The `__init__` function is called when the class is instantiated.  
It sets up the object with all of its initial values and does any other setup that needs to be done.

```
:param self: Refer to the object itself
:param filepaths: Store the filepaths of all the media files that are to be processed
:param start_frames: Specify the starting frame of each video
:param stop_frames: Set the last frame to be read from a video file
:param unit: Specify the unit of the start and stop frames
:param path_map: Map the filepaths to a new location
:return: Nothing
:doc-author: Joel Brogan, BRIAR team, Trelent
```

## 7.18.2 Member Function Documentation

### 7.18.2.1 `__iter__()`

```
def __iter__ (
    self )
```

The `__iter__` function is called when an iterator object is created for the class.  
This function should return an object (usually just `self`) that has a `next()` method defined.  
The `next()` method should return the next value for the iterable, or raise `StopIteration` if there are no more values.

```
:param self: Represent the instance of the class
:return: The object itself
:doc-author: Joel Brogan, BRIAR team, Trelent
```

### 7.18.2.2 `__len__()`

```
def __len__ (
    self )
```

The `__len__` function is a special function that returns the length of an object. In this case, it's returning the number of media objects in the set.

```
:param self: Refer to the class itself
:return: The length of the media_set list
:doc-author: Joel Brogan, BRIAR team, Trelent
```

### 7.18.2.3 `__next__()`

```
def __next__ (
    self )
```

The `__next__` function is called by the for loop to get the next item in the iterable. It should raise `StopIteration` when there are no more items to return.

```
:param self: Represent the instance of the class
:return: The next item in the sequence
:doc-author: Joel Brogan, BRIAR team, Trelent
```

## 7.18.3 Member Data Documentation

### 7.18.3.1 `filepaths`

`filepaths`

### 7.18.3.2 `i`

`i`

### 7.18.3.3 `isOpened`

`isOpened`



#### 7.18.3.4 media\_set

media\_set

#### 7.18.3.5 processed

processed

#### 7.18.3.6 start\_frame

start\_frame

#### 7.18.3.7 start\_frames

start\_frames

#### 7.18.3.8 stop\_frame

stop\_frame

#### 7.18.3.9 stop\_frames

stop\_frames

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

- [media/\\_\\_\\_init\\_\\_.py](#)

## 7.19 MediaStream Class Reference

### Public Member Functions

- [def \\_\\_init\\_\\_](#) (self, briar\_media)
- [def \\_\\_iter\\_\\_](#) (self, request\_type)

## Private Attributes

- [\\_media\\_list](#)

## 7.19.1 Constructor & Destructor Documentation

### 7.19.1.1 `__init__()`

```
def __init__ (
    self,
    briar_media )
```

## 7.19.2 Member Function Documentation

### 7.19.2.1 `__iter__()`

```
def __iter__ (
    self,
    request_type )
```

## 7.19.3 Member Data Documentation

### 7.19.3.1 `_media_list`

```
_media_list [private]
```

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

- [briar\\_media.py](#)

## 7.20 Rect Class Reference

### Public Member Functions

- `def __init__(self, x, y, width, height)`

## Public Attributes

- [height](#)
- [width](#)
- [x](#)
- [y](#)

### 7.20.1 Detailed Description

Basic rectangle for storing ROIs without needing to mess with the gRPC BriarRect

### 7.20.2 Constructor & Destructor Documentation

#### 7.20.2.1 `__init__()`

```
def __init__ (
    self,
    x,
    y,
    width,
    height )
```

The `__init__` function is called when the class is instantiated. It sets up the object with its initial state.

```
:param self: Represent the instance of the class
:param x: Set the x coordinate of the rectangle
:param y: Set the y coordinate of the rectangle
:param width: Set the width of the rectangle
:param height: Set the height of the rectangle
:return: Nothing
:doc-author: Joel Brogan
```

### 7.20.3 Member Data Documentation

#### 7.20.3.1 `height`

`height`

#### 7.20.3.2 `width`

`width`

### 7.20.3.3 x

x

### 7.20.3.4 y

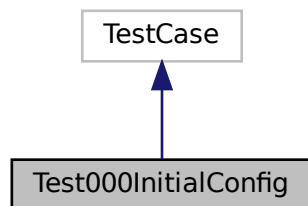
y

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

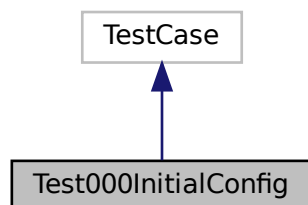
- [\\_\\_init\\_\\_.py](#)

## 7.21 Test000InitialConfig Class Reference

Inheritance diagram for Test000InitialConfig:



Collaboration diagram for Test000InitialConfig:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_config\\_portlist](#) (self)
- def [test\\_02\\_port\\_connections](#) (self)
- def [test\\_03\\_num\\_service\\_ports](#) (self)
- def [test\\_04\\_num\\_procs\\_per\\_port](#) (self)
- def [test\\_05\\_num\\_threads\\_per\\_port](#) (self)
- def [test\\_06\\_correct\\_database\\_creation](#) (self)
- def [testDatasetDir](#) (self)
- def [testOutDir](#) (self)
- def [testValidationDir](#) (self)

## Public Attributes

- [config\\_reply](#)

### 7.21.1 Member Function Documentation

#### 7.21.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

#### 7.21.1.2 test\_01\_config\_portlist()

```
def test_01_config_portlist (  
    self )
```

Testing configuration port list

#### 7.21.1.3 test\_02\_port\_connections()

```
def test_02_port_connections (  
    self )
```

Testing port connections

**7.21.1.4 test\_03\_num\_service\_ports()**

```
def test_03_num_service_ports (
    self )
```

Testing number of ports variable

**7.21.1.5 test\_04\_num\_procs\_per\_port()**

```
def test_04_num_procs_per_port (
    self )
```

**7.21.1.6 test\_05\_num\_threads\_per\_port()**

```
def test_05_num_threads_per_port (
    self )
```

**7.21.1.7 test\_06\_correct\_database\_creation()**

```
def test_06_correct_database_creation (
    self )
```

**7.21.1.8 testDatasetDir()**

```
def testDatasetDir (
    self )
```

**7.21.1.9 testOutDir()**

```
def testOutDir (
    self )
```

### 7.21.1.10 testValidationDir()

```
def testValidationDir (
    self )
```

## 7.21.2 Member Data Documentation

### 7.21.2.1 config\_reply

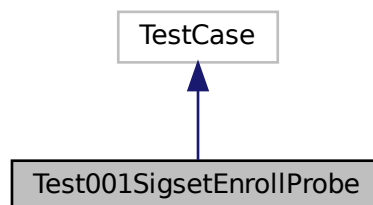
config\_reply

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

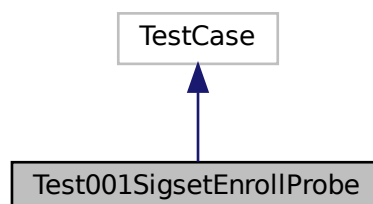
- [evaluation/full\\_evaluation.py](#)

## 7.22 Test001SigsetEnrollProbe Class Reference

Inheritance diagram for Test001SigsetEnrollProbe:



Collaboration diagram for Test001SigsetEnrollProbe:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_enroll\\_probe](#) (self)
- def [test\\_02\\_probe\\_checkpoint](#) (self)

### 7.22.1 Member Function Documentation

#### 7.22.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

#### 7.22.1.2 test\_01\_sigset\_enroll\_probe()

```
def test_01_sigset_enroll_probe (  
    self )
```

#### 7.22.1.3 test\_02\_probe\_checkpoint()

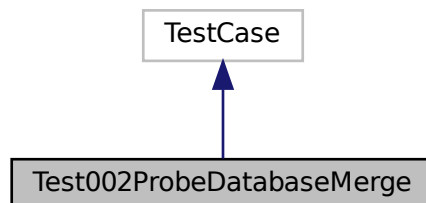
```
def test_02_probe_checkpoint (  
    self )
```

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

- [evaluation/full\\_evaluation.py](#)

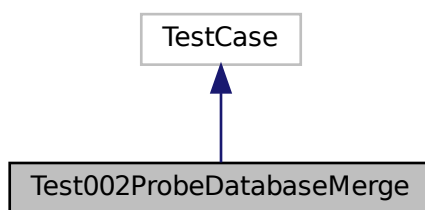
## 7.23 Test002ProbeDatabaseMerge Class Reference

Inheritance diagram for Test002ProbeDatabaseMerge:





Collaboration diagram for Test002ProbeDatabaseMerge:



## Public Member Functions

- None `setUpClass` (cls)
- def `test_02_merge_probe_dbs` (self)
- def `test_03_checkpoint_merged_probe_db` (self)

## Public Attributes

- `merged_dbinfo`
- `total_entries`
- `total_failed`
- `total_templates`

### 7.23.1 Member Function Documentation

#### 7.23.1.1 `setUpClass()`

```
None setUpClass (  
    cls )
```

#### 7.23.1.2 `test_02_merge_probe_dbs()`

```
def test_02_merge_probe_dbs (  
    self )
```

### 7.23.1.3 test\_03\_checkpoint\_merged\_probe\_db()

```
def test_03_checkpoint_merged_probe_db (
    self )
```

## 7.23.2 Member Data Documentation

### 7.23.2.1 merged\_dbinfo

merged\_dbinfo

### 7.23.2.2 total\_entries

total\_entries

### 7.23.2.3 total\_failed

total\_failed

### 7.23.2.4 total\_templates

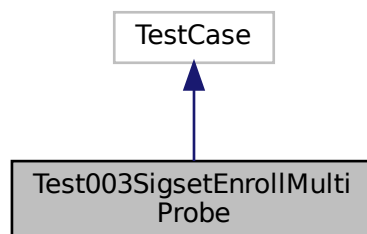
total\_templates

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

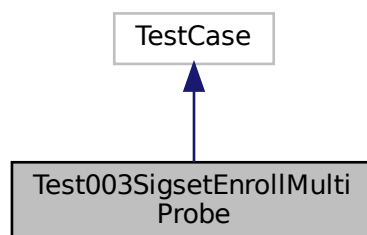
- [evaluation/full\\_evaluation.py](#)

## 7.24 Test003SigsetEnrollMultiProbe Class Reference

Inheritance diagram for Test003SigsetEnrollMultiProbe:



Collaboration diagram for Test003SigsetEnrollMultiProbe:



### Public Member Functions

- None `setUpClass` (cls)
- def `test_01_sigset_enroll_probe` (self)
- def `test_02_probe_checkpoint` (self)

#### 7.24.1 Member Function Documentation

##### 7.24.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

### 7.24.1.2 test\_01\_sigset\_enroll\_probe()

```
def test_01_sigset_enroll_probe (
    self )
```

### 7.24.1.3 test\_02\_probe\_checkpoint()

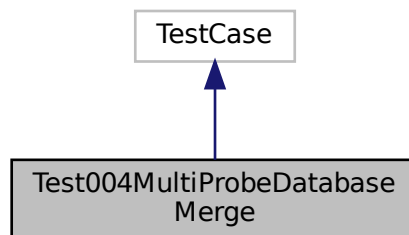
```
def test_02_probe_checkpoint (
    self )
```

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

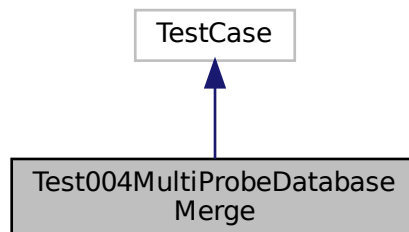
- [evaluation/full\\_evaluation.py](#)

## 7.25 Test004MultiProbeDatabaseMerge Class Reference

Inheritance diagram for Test004MultiProbeDatabaseMerge:



Collaboration diagram for Test004MultiProbeDatabaseMerge:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_02\\_merge\\_probe\\_dbs](#) (self)
- def [test\\_03\\_checkpoint\\_merged\\_probe\\_db](#) (self)

## Public Attributes

- [merged\\_dbinfo](#)
- [total\\_entries](#)
- [total\\_failed](#)
- [total\\_templates](#)

## 7.25.1 Member Function Documentation

### 7.25.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

### 7.25.1.2 test\_02\_merge\_probe\_dbs()

```
def test_02_merge_probe_dbs (  
    self )
```

### 7.25.1.3 test\_03\_checkpoint\_merged\_probe\_db()

```
def test_03_checkpoint_merged_probe_db (  
    self )
```

## 7.25.2 Member Data Documentation

### 7.25.2.1 merged\_dbinfo

```
merged_dbinfo
```

### 7.25.2.2 total\_entries

total\_entries

### 7.25.2.3 total\_failed

total\_failed

### 7.25.2.4 total\_templates

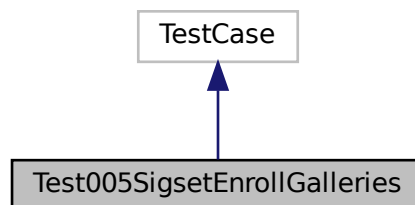
total\_templates

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

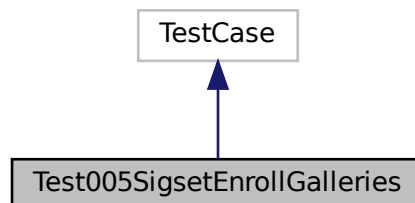
- [evaluation/full\\_evaluation.py](#)

## 7.26 Test005SigsetEnrollGalleries Class Reference

Inheritance diagram for Test005SigsetEnrollGalleries:



Collaboration diagram for Test005SigsetEnrollGalleries:



## Public Member Functions

- def [runGallery](#) (self, gal\_name, sigset\_path)
- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_enroll\\_gallery1](#) (self)
- def [test\\_02\\_gallery1\\_partitioned\\_checkpoint](#) (self)
- def [test\\_03\\_sigset\\_enroll\\_gallery2](#) (self)
- def [test\\_04\\_gallery2\\_partitioned\\_checkpoint](#) (self)
- def [test\\_05\\_sigset\\_enroll\\_gallery1](#) (self)
- def [test\\_06\\_gallery1\\_partitioned\\_checkpoint](#) (self)
- def [test\\_07\\_sigset\\_enroll\\_gallery2](#) (self)
- def [test\\_08\\_gallery2\\_partitioned\\_checkpoint](#) (self)

## 7.26.1 Member Function Documentation

### 7.26.1.1 [runGallery\(\)](#)

```
def runGallery (
    self,
    gal_name,
    sigset_path )
```

### 7.26.1.2 [setUpClass\(\)](#)

```
None setUpClass (
    cls )
```

### 7.26.1.3 [test\\_01\\_sigset\\_enroll\\_gallery1\(\)](#)

```
def test_01_sigset_enroll_gallery1 (
    self )
```

### 7.26.1.4 [test\\_02\\_gallery1\\_partitioned\\_checkpoint\(\)](#)

```
def test_02_gallery1_partitioned_checkpoint (
    self )
```

**7.26.1.5 test\_03\_sigset\_enroll\_gallery2()**

```
def test_03_sigset_enroll_gallery2 (
    self )
```

**7.26.1.6 test\_04\_gallery2\_partitioned\_checkpoint()**

```
def test_04_gallery2_partitioned_checkpoint (
    self )
```

**7.26.1.7 test\_05\_sigset\_enroll\_gallery1()**

```
def test_05_sigset_enroll_gallery1 (
    self )
```

**7.26.1.8 test\_06\_gallery1\_partitioned\_checkpoint()**

```
def test_06_gallery1_partitioned_checkpoint (
    self )
```

**7.26.1.9 test\_07\_sigset\_enroll\_gallery2()**

```
def test_07_sigset_enroll_gallery2 (
    self )
```

**7.26.1.10 test\_08\_gallery2\_partitioned\_checkpoint()**

```
def test_08_gallery2_partitioned_checkpoint (
    self )
```

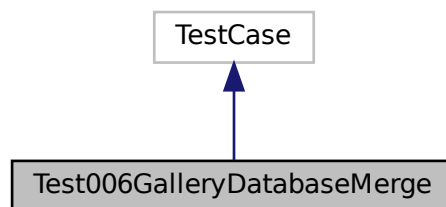
The documentation for this class was generated from the following file:

- [evaluation/full\\_evaluation.py](#)

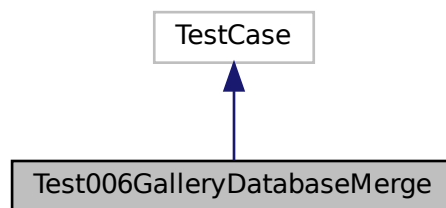


## 7.27 Test006GalleryDatabaseMerge Class Reference

Inheritance diagram for Test006GalleryDatabaseMerge:



Collaboration diagram for Test006GalleryDatabaseMerge:



### Public Member Functions

- def [merge\\_db\\_func](#) (self, gallery\_sigset, gallery\_db\_base\_name)
- None [setUpClass](#) (cls)
- def [test\\_01\\_merge\\_gallery1\\_dbs](#) (self)
- def [test\\_02\\_merge\\_gallery2\\_dbs](#) (self)
- def [test\\_03\\_merge\\_blended\\_gallery2\\_dbs](#) (self)
- def [test\\_04\\_merge\\_blended\\_gallery2\\_dbs](#) (self)
- def [test\\_05\\_finalized\\_merged\\_gallery1\\_db](#) (self)
- def [test\\_06\\_finalized\\_merged\\_gallery1\\_db](#) (self)
- def [test\\_07\\_finalized\\_merged\\_gallery1\\_db](#) (self)
- def [test\\_08\\_finalized\\_merged\\_gallery1\\_db](#) (self)

#### 7.27.1 Member Function Documentation

**7.27.1.1 merge\_db\_func()**

```
def merge_db_func (
    self,
    gallery_sigset,
    gallery_db_base_name )
```

**7.27.1.2 setUpClass()**

```
None setUpClass (
    cls )
```

**7.27.1.3 test\_01\_merge\_gallery1\_dbs()**

```
def test_01_merge_gallery1_dbs (
    self )
```

**7.27.1.4 test\_02\_merge\_gallery2\_dbs()**

```
def test_02_merge_gallery2_dbs (
    self )
```

**7.27.1.5 test\_03\_merge\_blended\_gallery2\_dbs()**

```
def test_03_merge_blended_gallery2_dbs (
    self )
```

**7.27.1.6 test\_04\_merge\_blended\_gallery2\_dbs()**

```
def test_04_merge_blended_gallery2_dbs (
    self )
```

**7.27.1.7 test\_05\_finalized\_merged\_gallery1\_db()**

```
def test_05_finalized_merged_gallery1_db (
    self )
```

### 7.27.1.8 test\_06\_finalized\_merged\_gallery1\_db()

```
def test_06_finalized_merged_gallery1_db (
    self )
```

### 7.27.1.9 test\_07\_finalized\_merged\_gallery1\_db()

```
def test_07_finalized_merged_gallery1_db (
    self )
```

### 7.27.1.10 test\_08\_finalized\_merged\_gallery1\_db()

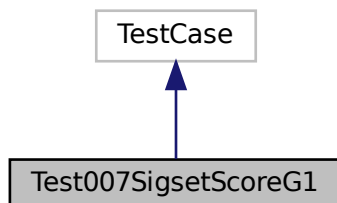
```
def test_08_finalized_merged_gallery1_db (
    self )
```

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

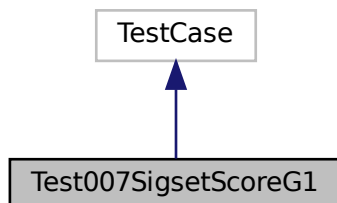
- [evaluation/full\\_evaluation.py](#)

## 7.28 Test007SigsetScoreG1 Class Reference

Inheritance diagram for Test007SigsetScoreG1:



Collaboration diagram for Test007SigsetScoreG1:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_probe\\_gallery1](#) (self)
- def [test\\_02\\_sigset\\_search\\_probe\\_gallery1](#) (self)

### 7.28.1 Member Function Documentation

#### 7.28.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

#### 7.28.1.2 test\_01\_sigset\_verify\_probe\_gallery1()

```
def test_01_sigset_verify_probe_gallery1 (  
    self )
```

#### 7.28.1.3 test\_02\_sigset\_search\_probe\_gallery1()

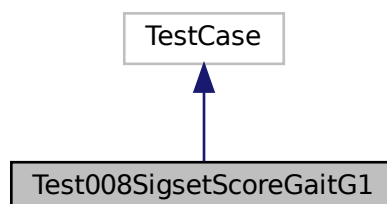
```
def test_02_sigset_search_probe_gallery1 (  
    self )
```

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

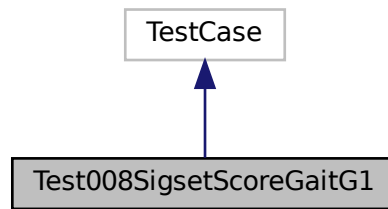
- [evaluation/full\\_evaluation.py](#)

## 7.29 Test008SigsetScoreGaitG1 Class Reference

Inheritance diagram for Test008SigsetScoreGaitG1:



Collaboration diagram for Test008SigsetScoreGaitG1:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_gait\\_probe\\_gallery1](#) (self)
- def [test\\_02\\_sigset\\_search\\_gait\\_probe\\_gallery1](#) (self)

### 7.29.1 Member Function Documentation

#### 7.29.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

#### 7.29.1.2 test\_01\_sigset\_verify\_gait\_probe\_gallery1()

```
def test_01_sigset_verify_gait_probe_gallery1 (  
    self )
```

#### 7.29.1.3 test\_02\_sigset\_search\_gait\_probe\_gallery1()

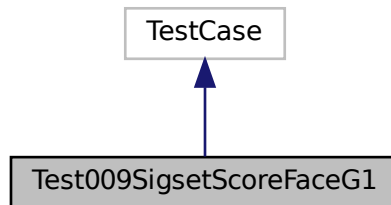
```
def test_02_sigset_search_gait_probe_gallery1 (  
    self )
```

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

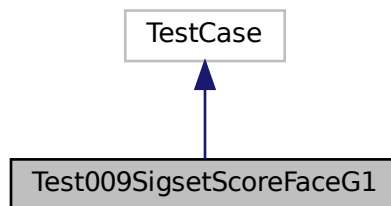
- [evaluation/full\\_evaluation.py](#)

## 7.30 Test009SigsetScoreFaceG1 Class Reference

Inheritance diagram for Test009SigsetScoreFaceG1:



Collaboration diagram for Test009SigsetScoreFaceG1:



### Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_face\\_probe\\_gallery1](#) (self)
- def [test\\_02\\_sigset\\_search\\_face\\_probe\\_gallery1](#) (self)

### 7.30.1 Member Function Documentation

#### 7.30.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

### 7.30.1.2 test\_01\_sigset\_verify\_face\_probe\_gallery1()

```
def test_01_sigset_verify_face_probe_gallery1 (
    self )
```

### 7.30.1.3 test\_02\_sigset\_search\_face\_probe\_gallery1()

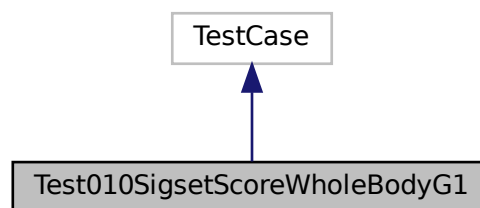
```
def test_02_sigset_search_face_probe_gallery1 (
    self )
```

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

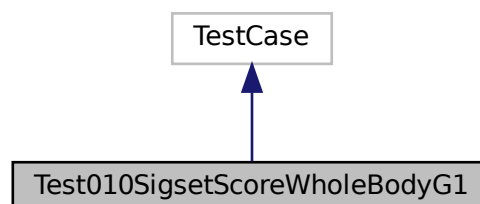
- [evaluation/full\\_evaluation.py](#)

## 7.31 Test010SigsetScoreWholeBodyG1 Class Reference

Inheritance diagram for Test010SigsetScoreWholeBodyG1:



Collaboration diagram for Test010SigsetScoreWholeBodyG1:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_WB\\_probe\\_gallery1](#) (self)
- def [test\\_02\\_sigset\\_search\\_WB\\_probe\\_gallery1](#) (self)

### 7.31.1 Member Function Documentation

#### 7.31.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

#### 7.31.1.2 test\_01\_sigset\_verify\_WB\_probe\_gallery1()

```
def test_01_sigset_verify_WB_probe_gallery1 (  
    self )
```

#### 7.31.1.3 test\_02\_sigset\_search\_WB\_probe\_gallery1()

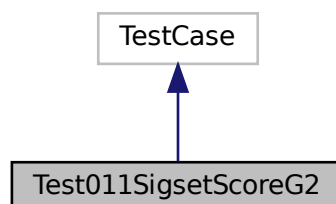
```
def test_02_sigset_search_WB_probe_gallery1 (  
    self )
```

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

- [evaluation/full\\_evaluation.py](#)

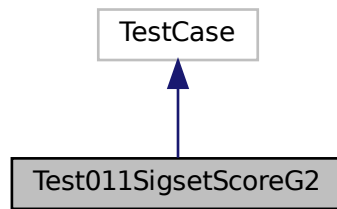
## 7.32 Test011SigsetScoreG2 Class Reference

Inheritance diagram for Test011SigsetScoreG2:





Collaboration diagram for Test011SigsetScoreG2:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_probe\\_gallery2](#) (self)
- def [test\\_02\\_sigset\\_search\\_probe\\_gallery2](#) (self)

### 7.32.1 Member Function Documentation

#### 7.32.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

#### 7.32.1.2 test\_01\_sigset\_verify\_probe\_gallery2()

```
def test_01_sigset_verify_probe_gallery2 (  
    self )
```

#### 7.32.1.3 test\_02\_sigset\_search\_probe\_gallery2()

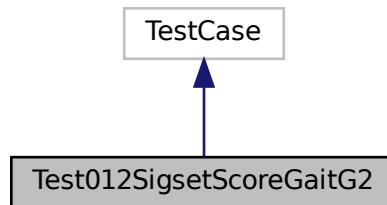
```
def test_02_sigset_search_probe_gallery2 (  
    self )
```

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

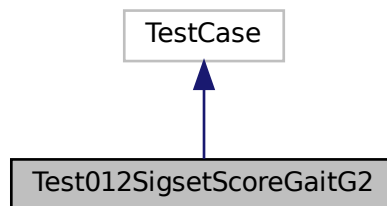
- [evaluation/full\\_evaluation.py](#)

## 7.33 Test012SigsetScoreGaitG2 Class Reference

Inheritance diagram for Test012SigsetScoreGaitG2:



Collaboration diagram for Test012SigsetScoreGaitG2:



### Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_gait\\_probe\\_gallery2](#) (self)
- def [test\\_02\\_sigset\\_search\\_gait\\_probe\\_gallery2](#) (self)

### 7.33.1 Member Function Documentation

#### 7.33.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

### 7.33.1.2 test\_01\_sigset\_verify\_gait\_probe\_gallery2()

```
def test_01_sigset_verify_gait_probe_gallery2 (
    self )
```

### 7.33.1.3 test\_02\_sigset\_search\_gait\_probe\_gallery2()

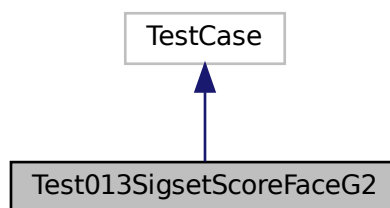
```
def test_02_sigset_search_gait_probe_gallery2 (
    self )
```

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

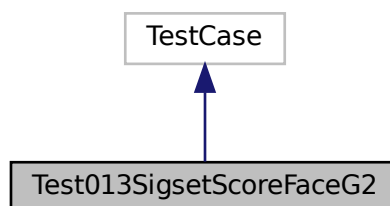
- [evaluation/full\\_evaluation.py](#)

## 7.34 Test013SigsetScoreFaceG2 Class Reference

Inheritance diagram for Test013SigsetScoreFaceG2:



Collaboration diagram for Test013SigsetScoreFaceG2:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_face\\_probe\\_gallery2](#) (self)
- def [test\\_02\\_sigset\\_search\\_face\\_probe\\_gallery2](#) (self)

### 7.34.1 Member Function Documentation

#### 7.34.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

#### 7.34.1.2 test\_01\_sigset\_verify\_face\_probe\_gallery2()

```
def test_01_sigset_verify_face_probe_gallery2 (  
    self )
```

#### 7.34.1.3 test\_02\_sigset\_search\_face\_probe\_gallery2()

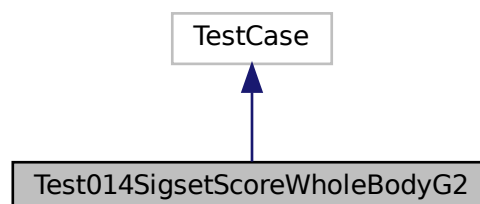
```
def test_02_sigset_search_face_probe_gallery2 (  
    self )
```

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

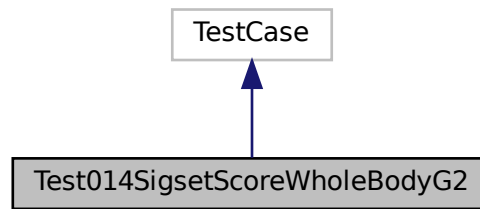
- [evaluation/full\\_evaluation.py](#)

## 7.35 Test014SigsetScoreWholeBodyG2 Class Reference

Inheritance diagram for Test014SigsetScoreWholeBodyG2:



Collaboration diagram for Test014SigsetScoreWholeBodyG2:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_WB\\_probe\\_gallery2](#) (self)
- def [test\\_02\\_sigset\\_search\\_WB\\_probe\\_gallery2](#) (self)

### 7.35.1 Member Function Documentation

#### 7.35.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

#### 7.35.1.2 test\_01\_sigset\_verify\_WB\_probe\_gallery2()

```
def test_01_sigset_verify_WB_probe_gallery2 (  
    self )
```

#### 7.35.1.3 test\_02\_sigset\_search\_WB\_probe\_gallery2()

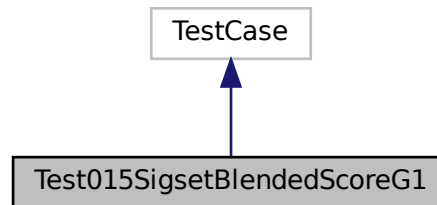
```
def test_02_sigset_search_WB_probe_gallery2 (  
    self )
```

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

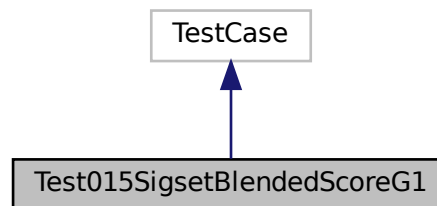
- [evaluation/full\\_evaluation.py](#)

## 7.36 Test015SigsetBlendedScoreG1 Class Reference

Inheritance diagram for Test015SigsetBlendedScoreG1:



Collaboration diagram for Test015SigsetBlendedScoreG1:



### Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_probe\\_blended\\_gallery1](#) (self)
- def [test\\_02\\_sigset\\_search\\_probe\\_blended\\_gallery1](#) (self)

### 7.36.1 Member Function Documentation

#### 7.36.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

### 7.36.1.2 test\_01\_sigset\_verify\_probe\_blended\_gallery1()

```
def test_01_sigset_verify_probe_blended_gallery1 (
    self )
```

### 7.36.1.3 test\_02\_sigset\_search\_probe\_blended\_gallery1()

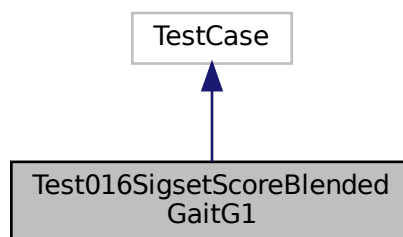
```
def test_02_sigset_search_probe_blended_gallery1 (
    self )
```

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

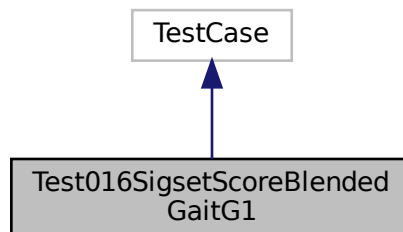
- [evaluation/full\\_evaluation.py](#)

## 7.37 Test016SigsetScoreBlendedGaitG1 Class Reference

Inheritance diagram for Test016SigsetScoreBlendedGaitG1:



Collaboration diagram for Test016SigsetScoreBlendedGaitG1:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_gait\\_probe\\_blended\\_gallery1](#) (self)
- def [test\\_02\\_sigset\\_search\\_gait\\_probe\\_blended\\_gallery1](#) (self)

### 7.37.1 Member Function Documentation

#### 7.37.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

#### 7.37.1.2 test\_01\_sigset\_verify\_gait\_probe\_blended\_gallery1()

```
def test_01_sigset_verify_gait_probe_blended_gallery1 (  
    self )
```

#### 7.37.1.3 test\_02\_sigset\_search\_gait\_probe\_blended\_gallery1()

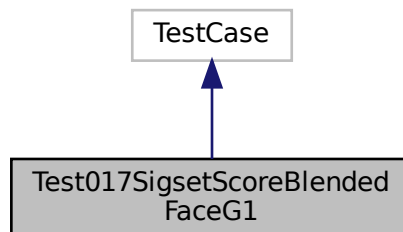
```
def test_02_sigset_search_gait_probe_blended_gallery1 (  
    self )
```

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

- [evaluation/full\\_evaluation.py](#)

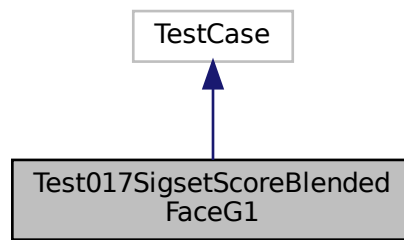
## 7.38 Test017SigsetScoreBlendedFaceG1 Class Reference

Inheritance diagram for Test017SigsetScoreBlendedFaceG1:





Collaboration diagram for Test017SigsetScoreBlendedFaceG1:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_face\\_probe\\_blended\\_gallery1](#) (self)
- def [test\\_02\\_sigset\\_search\\_face\\_probe\\_blended\\_gallery1](#) (self)

### 7.38.1 Member Function Documentation

#### 7.38.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

#### 7.38.1.2 test\_01\_sigset\_verify\_face\_probe\_blended\_gallery1()

```
def test_01_sigset_verify_face_probe_blended_gallery1 (  
    self )
```

#### 7.38.1.3 test\_02\_sigset\_search\_face\_probe\_blended\_gallery1()

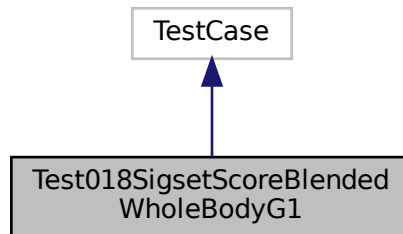
```
def test_02_sigset_search_face_probe_blended_gallery1 (  
    self )
```

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

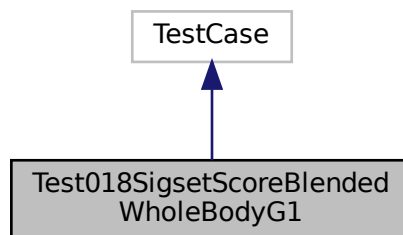
- [evaluation/full\\_evaluation.py](#)

## 7.39 Test018SigsetScoreBlendedWholeBodyG1 Class Reference

Inheritance diagram for Test018SigsetScoreBlendedWholeBodyG1:



Collaboration diagram for Test018SigsetScoreBlendedWholeBodyG1:



### Public Member Functions

- None `setUpClass` (cls)
- def `test_01_sigset_verify_WB_probe_blended_gallery1` (self)
- def `test_02_sigset_search_WB_probe_blended_gallery1` (self)

### 7.39.1 Member Function Documentation

#### 7.39.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

### 7.39.1.2 test\_01\_sigset\_verify\_WB\_probe\_blended\_gallery1()

```
def test_01_sigset_verify_WB_probe_blended_gallery1 (
    self )
```

### 7.39.1.3 test\_02\_sigset\_search\_WB\_probe\_blended\_gallery1()

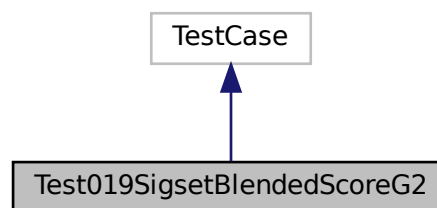
```
def test_02_sigset_search_WB_probe_blended_gallery1 (
    self )
```

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

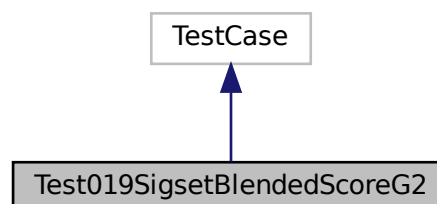
- [evaluation/full\\_evaluation.py](#)

## 7.40 Test019SigsetBlendedScoreG2 Class Reference

Inheritance diagram for Test019SigsetBlendedScoreG2:



Collaboration diagram for Test019SigsetBlendedScoreG2:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_probe\\_blended\\_gallery2](#) (self)
- def [test\\_02\\_sigset\\_search\\_probe\\_blended\\_gallery2](#) (self)

### 7.40.1 Member Function Documentation

#### 7.40.1.1 setUpClass()

```
None setUpClass (
    cls )
```

#### 7.40.1.2 test\_01\_sigset\_verify\_probe\_blended\_gallery2()

```
def test_01_sigset_verify_probe_blended_gallery2 (
    self )
```

#### 7.40.1.3 test\_02\_sigset\_search\_probe\_blended\_gallery2()

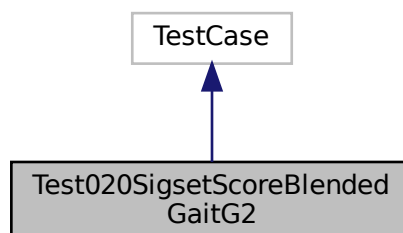
```
def test_02_sigset_search_probe_blended_gallery2 (
    self )
```

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

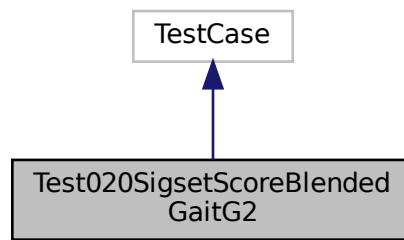
- [evaluation/full\\_evaluation.py](#)

## 7.41 Test020SigsetScoreBlendedGaitG2 Class Reference

Inheritance diagram for Test020SigsetScoreBlendedGaitG2:



Collaboration diagram for Test020SigsetScoreBlendedGaitG2:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_gait\\_probe\\_blended\\_gallery2](#) (self)
- def [test\\_02\\_sigset\\_search\\_gait\\_probe\\_blended\\_gallery2](#) (self)

### 7.41.1 Member Function Documentation

#### 7.41.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

#### 7.41.1.2 test\_01\_sigset\_verify\_gait\_probe\_blended\_gallery2()

```
def test_01_sigset_verify_gait_probe_blended_gallery2 (  
    self )
```

#### 7.41.1.3 test\_02\_sigset\_search\_gait\_probe\_blended\_gallery2()

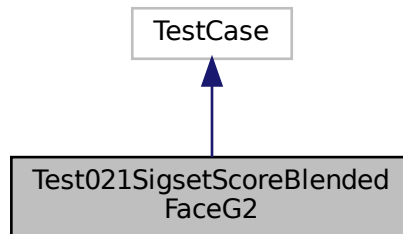
```
def test_02_sigset_search_gait_probe_blended_gallery2 (  
    self )
```

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

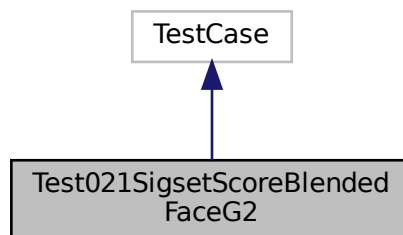
- [evaluation/full\\_evaluation.py](#)

## 7.42 Test021SigsetScoreBlendedFaceG2 Class Reference

Inheritance diagram for Test021SigsetScoreBlendedFaceG2:



Collaboration diagram for Test021SigsetScoreBlendedFaceG2:



### Public Member Functions

- None `setUpClass` (cls)
- def `test_01_sigset_verify_face_probe_blended_gallery2` (self)
- def `test_02_sigset_search_face_probe_blended_gallery2` (self)

### 7.42.1 Member Function Documentation

#### 7.42.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

### 7.42.1.2 test\_01\_sigset\_verify\_face\_probe\_blended\_gallery2()

```
def test_01_sigset_verify_face_probe_blended_gallery2 (
    self )
```

### 7.42.1.3 test\_02\_sigset\_search\_face\_probe\_blended\_gallery2()

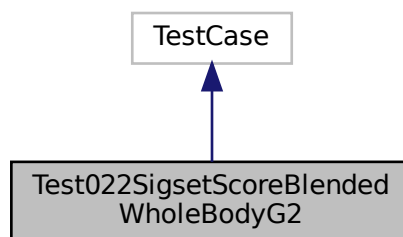
```
def test_02_sigset_search_face_probe_blended_gallery2 (
    self )
```

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

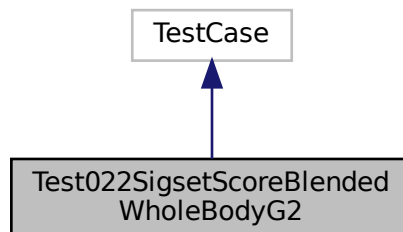
- [evaluation/full\\_evaluation.py](#)

## 7.43 Test022SigsetScoreBlendedWholeBodyG2 Class Reference

Inheritance diagram for Test022SigsetScoreBlendedWholeBodyG2:



Collaboration diagram for Test022SigsetScoreBlendedWholeBodyG2:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_WB\\_probe\\_blended\\_gallery2](#) (self)
- def [test\\_02\\_sigset\\_search\\_WB\\_probe\\_blended\\_gallery2](#) (self)

### 7.43.1 Member Function Documentation

#### 7.43.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

#### 7.43.1.2 test\_01\_sigset\_verify\_WB\_probe\_blended\_gallery2()

```
def test_01_sigset_verify_WB_probe_blended_gallery2 (  
    self )
```

#### 7.43.1.3 test\_02\_sigset\_search\_WB\_probe\_blended\_gallery2()

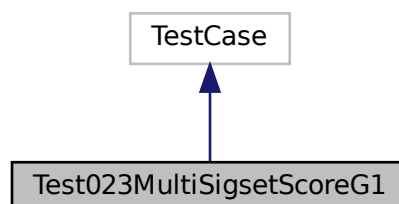
```
def test_02_sigset_search_WB_probe_blended_gallery2 (  
    self )
```

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

- [evaluation/full\\_evaluation.py](#)

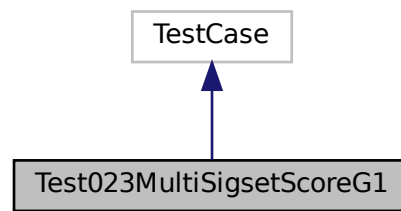
## 7.44 Test023MultiSigsetScoreG1 Class Reference

Inheritance diagram for Test023MultiSigsetScoreG1:





Collaboration diagram for Test023MultiSigsetScoreG1:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_multiprobe\\_gallery1](#) (self)
- def [test\\_02\\_sigset\\_search\\_multiprobe\\_gallery1](#) (self)

### 7.44.1 Member Function Documentation

#### 7.44.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

#### 7.44.1.2 test\_01\_sigset\_verify\_multiprobe\_gallery1()

```
def test_01_sigset_verify_multiprobe_gallery1 (  
    self )
```

#### 7.44.1.3 test\_02\_sigset\_search\_multiprobe\_gallery1()

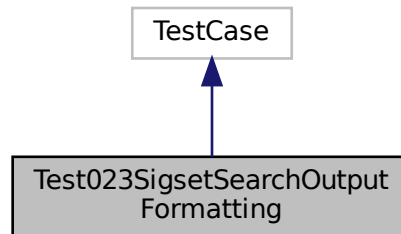
```
def test_02_sigset_search_multiprobe_gallery1 (  
    self )
```

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

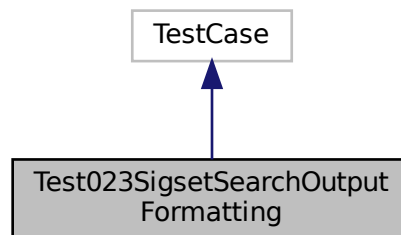
- [evaluation/full\\_evaluation.py](#)

## 7.45 Test023SigsetSearchOutputFormatting Class Reference

Inheritance diagram for Test023SigsetSearchOutputFormatting:



Collaboration diagram for Test023SigsetSearchOutputFormatting:



### Public Member Functions

- def [search\\_file\\_check](#) (self, modality="", blended=False, galnumber=1)
- def [test\\_01\\_sigset\\_search\\_pickle\\_fileG1](#) (self)
- def [test\\_02\\_sigset\\_search\\_gait\\_pickle\\_fileG1](#) (self)
- def [test\\_03\\_sigset\\_search\\_face\\_pickle\\_fileG1](#) (self)
- def [test\\_04\\_sigset\\_search\\_wb\\_pickle\\_fileG1](#) (self)
- def [test\\_05\\_sigset\\_search\\_pickle\\_fileG2](#) (self)
- def [test\\_06\\_sigset\\_search\\_gait\\_pickle\\_fileG2](#) (self)
- def [test\\_07\\_sigset\\_search\\_face\\_pickle\\_fileG2](#) (self)
- def [test\\_08\\_sigset\\_search\\_wb\\_pickle\\_fileG2](#) (self)
- def [test\\_09\\_sigset\\_blended\\_search\\_pickle\\_fileG1](#) (self)
- def [test\\_10\\_sigset\\_blended\\_search\\_gait\\_pickle\\_fileG1](#) (self)
- def [test\\_11\\_sigset\\_blended\\_search\\_face\\_pickle\\_fileG1](#) (self)
- def [test\\_12\\_sigset\\_blended\\_search\\_wb\\_pickle\\_fileG1](#) (self)
- def [test\\_13\\_sigset\\_blended\\_search\\_pickle\\_fileG2](#) (self)
- def [test\\_14\\_sigset\\_blended\\_search\\_gait\\_pickle\\_fileG2](#) (self)
- def [test\\_15\\_sigset\\_blended\\_search\\_face\\_pickle\\_fileG2](#) (self)
- def [test\\_16\\_sigset\\_blended\\_search\\_wb\\_pickle\\_fileG2](#) (self)

## 7.45.1 Member Function Documentation

### 7.45.1.1 `search_file_check()`

```
def search_file_check (
    self,
    modality = '',
    blended = False,
    galnumber = 1 )
```

### 7.45.1.2 `test_01_sigset_search_pickle_fileG1()`

```
def test_01_sigset_search_pickle_fileG1 (
    self )
```

### 7.45.1.3 `test_02_sigset_search_gait_pickle_fileG1()`

```
def test_02_sigset_search_gait_pickle_fileG1 (
    self )
```

### 7.45.1.4 `test_03_sigset_search_face_pickle_fileG1()`

```
def test_03_sigset_search_face_pickle_fileG1 (
    self )
```

### 7.45.1.5 `test_04_sigset_search_wb_pickle_fileG1()`

```
def test_04_sigset_search_wb_pickle_fileG1 (
    self )
```

### 7.45.1.6 `test_05_sigset_search_pickle_fileG2()`

```
def test_05_sigset_search_pickle_fileG2 (
    self )
```

**7.45.1.7 test\_06\_sigset\_search\_gait\_pickle\_fileG2()**

```
def test_06_sigset_search_gait_pickle_fileG2 (
    self )
```

**7.45.1.8 test\_07\_sigset\_search\_face\_pickle\_fileG2()**

```
def test_07_sigset_search_face_pickle_fileG2 (
    self )
```

**7.45.1.9 test\_08\_sigset\_search\_wb\_pickle\_fileG2()**

```
def test_08_sigset_search_wb_pickle_fileG2 (
    self )
```

**7.45.1.10 test\_09\_sigset\_blended\_search\_pickle\_fileG1()**

```
def test_09_sigset_blended_search_pickle_fileG1 (
    self )
```

**7.45.1.11 test\_10\_sigset\_blended\_search\_gait\_pickle\_fileG1()**

```
def test_10_sigset_blended_search_gait_pickle_fileG1 (
    self )
```

**7.45.1.12 test\_11\_sigset\_blended\_search\_face\_pickle\_fileG1()**

```
def test_11_sigset_blended_search_face_pickle_fileG1 (
    self )
```

**7.45.1.13 test\_12\_sigset\_blended\_search\_wb\_pickle\_fileG1()**

```
def test_12_sigset_blended_search_wb_pickle_fileG1 (
    self )
```

**7.45.1.14 test\_13\_sigset\_blended\_search\_pickle\_fileG2()**

```
def test_13_sigset_blended_search_pickle_fileG2 (
    self )
```

**7.45.1.15 test\_14\_sigset\_blended\_search\_gait\_pickle\_fileG2()**

```
def test_14_sigset_blended_search_gait_pickle_fileG2 (
    self )
```

**7.45.1.16 test\_15\_sigset\_blended\_search\_face\_pickle\_fileG2()**

```
def test_15_sigset_blended_search_face_pickle_fileG2 (
    self )
```

**7.45.1.17 test\_16\_sigset\_blended\_search\_wb\_pickle\_fileG2()**

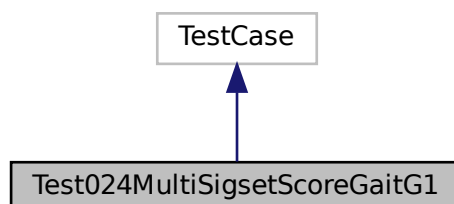
```
def test_16_sigset_blended_search_wb_pickle_fileG2 (
    self )
```

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

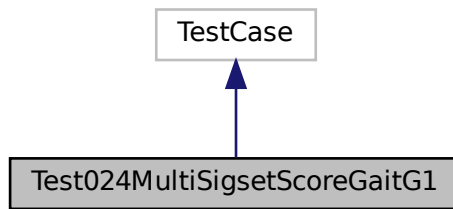
- [evaluation/full\\_evaluation.py](#)

## 7.46 Test024MultiSigsetScoreGaitG1 Class Reference

Inheritance diagram for Test024MultiSigsetScoreGaitG1:



Collaboration diagram for Test024MultiSigsetScoreGaitG1:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_gait\\_multiprobe\\_gallery1](#) (self)
- def [test\\_02\\_sigset\\_search\\_gait\\_multiprobe\\_gallery1](#) (self)

### 7.46.1 Member Function Documentation

#### 7.46.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

#### 7.46.1.2 test\_01\_sigset\_verify\_gait\_multiprobe\_gallery1()

```
def test_01_sigset_verify_gait_multiprobe_gallery1 (  
    self )
```

#### 7.46.1.3 test\_02\_sigset\_search\_gait\_multiprobe\_gallery1()

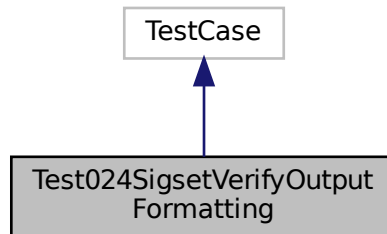
```
def test_02_sigset_search_gait_multiprobe_gallery1 (  
    self )
```

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

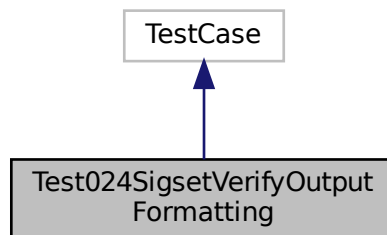
- [evaluation/full\\_evaluation.py](#)

## 7.47 Test024SigsetVerifyOutputFormatting Class Reference

Inheritance diagram for Test024SigsetVerifyOutputFormatting:



Collaboration diagram for Test024SigsetVerifyOutputFormatting:



### Public Member Functions

- def [score\\_file\\_check](#) (self, modality="", blended=False, galnumber=1)
- def [test\\_01\\_sigset\\_verify\\_pickle\\_fileG1](#) (self)
- def [test\\_02\\_sigset\\_verify\\_gait\\_pickle\\_fileG1](#) (self)
- def [test\\_03\\_sigset\\_verify\\_face\\_pickle\\_fileG1](#) (self)
- def [test\\_04\\_sigset\\_verify\\_wb\\_pickle\\_fileG1](#) (self)
- def [test\\_05\\_sigset\\_verify\\_pickle\\_fileG2](#) (self)
- def [test\\_06\\_sigset\\_verify\\_gait\\_pickle\\_fileG2](#) (self)
- def [test\\_07\\_sigset\\_verify\\_face\\_pickle\\_fileG2](#) (self)
- def [test\\_08\\_sigset\\_verify\\_wb\\_pickle\\_fileG2](#) (self)
- def [test\\_09\\_sigset\\_blended\\_verify\\_pickle\\_fileG1](#) (self)
- def [test\\_10\\_sigset\\_blended\\_verify\\_gait\\_pickle\\_fileG1](#) (self)
- def [test\\_11\\_sigset\\_blended\\_verify\\_face\\_pickle\\_fileG1](#) (self)
- def [test\\_12\\_sigset\\_blended\\_verify\\_wb\\_pickle\\_fileG1](#) (self)
- def [test\\_13\\_sigset\\_blended\\_verify\\_pickle\\_fileG2](#) (self)
- def [test\\_14\\_sigset\\_blended\\_verify\\_gait\\_pickle\\_fileG2](#) (self)
- def [test\\_15\\_sigset\\_blended\\_verify\\_face\\_pickle\\_fileG2](#) (self)
- def [test\\_16\\_sigset\\_blended\\_verify\\_wb\\_pickle\\_fileG2](#) (self)

## 7.47.1 Member Function Documentation

### 7.47.1.1 `score_file_check()`

```
def score_file_check (
    self,
    modality = '',
    blended = False,
    galnumber = 1 )
```

### 7.47.1.2 `test_01_sigset_verify_pickle_fileG1()`

```
def test_01_sigset_verify_pickle_fileG1 (
    self )
```

### 7.47.1.3 `test_02_sigset_verify_gait_pickle_fileG1()`

```
def test_02_sigset_verify_gait_pickle_fileG1 (
    self )
```

### 7.47.1.4 `test_03_sigset_verify_face_pickle_fileG1()`

```
def test_03_sigset_verify_face_pickle_fileG1 (
    self )
```

### 7.47.1.5 `test_04_sigset_verify_wb_pickle_fileG1()`

```
def test_04_sigset_verify_wb_pickle_fileG1 (
    self )
```

### 7.47.1.6 `test_05_sigset_verify_pickle_fileG2()`

```
def test_05_sigset_verify_pickle_fileG2 (
    self )
```



**7.47.1.7 test\_06\_sigset\_verify\_gait\_pickle\_fileG2()**

```
def test_06_sigset_verify_gait_pickle_fileG2 (
    self )
```

**7.47.1.8 test\_07\_sigset\_verify\_face\_pickle\_fileG2()**

```
def test_07_sigset_verify_face_pickle_fileG2 (
    self )
```

**7.47.1.9 test\_08\_sigset\_verify\_wb\_pickle\_fileG2()**

```
def test_08_sigset_verify_wb_pickle_fileG2 (
    self )
```

**7.47.1.10 test\_09\_sigset\_blended\_verify\_pickle\_fileG1()**

```
def test_09_sigset_blended_verify_pickle_fileG1 (
    self )
```

**7.47.1.11 test\_10\_sigset\_blended\_verify\_gait\_pickle\_fileG1()**

```
def test_10_sigset_blended_verify_gait_pickle_fileG1 (
    self )
```

**7.47.1.12 test\_11\_sigset\_blended\_verify\_face\_pickle\_fileG1()**

```
def test_11_sigset_blended_verify_face_pickle_fileG1 (
    self )
```

**7.47.1.13 test\_12\_sigset\_blended\_verify\_wb\_pickle\_fileG1()**

```
def test_12_sigset_blended_verify_wb_pickle_fileG1 (
    self )
```

**7.47.1.14 test\_13\_sigset\_blended\_verify\_pickle\_fileG2()**

```
def test_13_sigset_blended_verify_pickle_fileG2 (
    self )
```

**7.47.1.15 test\_14\_sigset\_blended\_verify\_gait\_pickle\_fileG2()**

```
def test_14_sigset_blended_verify_gait_pickle_fileG2 (
    self )
```

**7.47.1.16 test\_15\_sigset\_blended\_verify\_face\_pickle\_fileG2()**

```
def test_15_sigset_blended_verify_face_pickle_fileG2 (
    self )
```

**7.47.1.17 test\_16\_sigset\_blended\_verify\_wb\_pickle\_fileG2()**

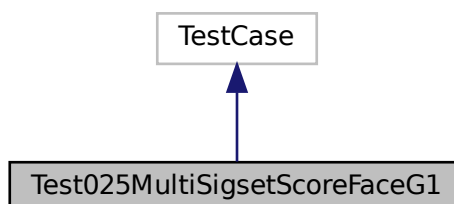
```
def test_16_sigset_blended_verify_wb_pickle_fileG2 (
    self )
```

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

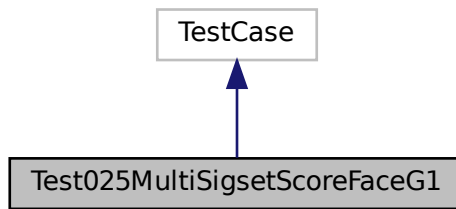
- [evaluation/full\\_evaluation.py](#)

## 7.48 Test025MultiSigsetScoreFaceG1 Class Reference

Inheritance diagram for Test025MultiSigsetScoreFaceG1:



Collaboration diagram for Test025MultiSigsetScoreFaceG1:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_face\\_multiprobe\\_gallery1](#) (self)
- def [test\\_02\\_sigset\\_search\\_face\\_multiprobe\\_gallery1](#) (self)

### 7.48.1 Member Function Documentation

#### 7.48.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

#### 7.48.1.2 test\_01\_sigset\_verify\_face\_multiprobe\_gallery1()

```
def test_01_sigset_verify_face_multiprobe_gallery1 (  
    self )
```

#### 7.48.1.3 test\_02\_sigset\_search\_face\_multiprobe\_gallery1()

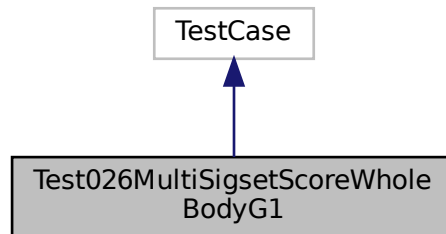
```
def test_02_sigset_search_face_multiprobe_gallery1 (  
    self )
```

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

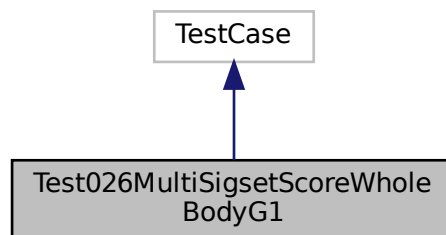
- [evaluation/full\\_evaluation.py](#)

## 7.49 Test026MultiSigsetScoreWholeBodyG1 Class Reference

Inheritance diagram for Test026MultiSigsetScoreWholeBodyG1:



Collaboration diagram for Test026MultiSigsetScoreWholeBodyG1:



### Public Member Functions

- None `setUpClass` (cls)
- def `test_01_sigset_verify_WB_multiprobe_gallery1` (self)
- def `test_02_sigset_search_WB_multiprobe_gallery1` (self)

### 7.49.1 Member Function Documentation

#### 7.49.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

### 7.49.1.2 test\_01\_sigset\_verify\_WB\_multiprobe\_gallery1()

```
def test_01_sigset_verify_WB_multiprobe_gallery1 (
    self )
```

### 7.49.1.3 test\_02\_sigset\_search\_WB\_multiprobe\_gallery1()

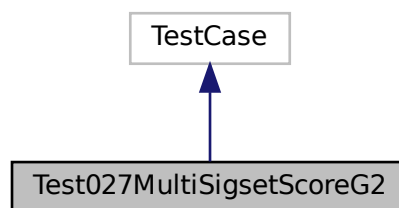
```
def test_02_sigset_search_WB_multiprobe_gallery1 (
    self )
```

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

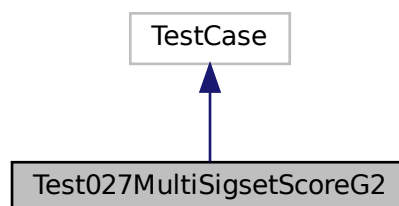
- [evaluation/full\\_evaluation.py](#)

## 7.50 Test027MultiSigsetScoreG2 Class Reference

Inheritance diagram for Test027MultiSigsetScoreG2:



Collaboration diagram for Test027MultiSigsetScoreG2:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_multiprobe\\_gallery2](#) (self)
- def [test\\_02\\_sigset\\_search\\_multiprobe\\_gallery2](#) (self)

### 7.50.1 Member Function Documentation

#### 7.50.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

#### 7.50.1.2 test\_01\_sigset\_verify\_multiprobe\_gallery2()

```
def test_01_sigset_verify_multiprobe_gallery2 (  
    self )
```

#### 7.50.1.3 test\_02\_sigset\_search\_multiprobe\_gallery2()

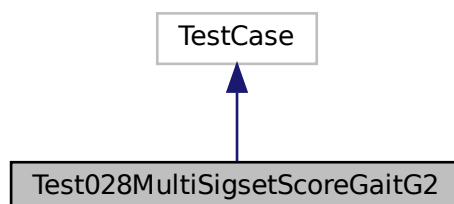
```
def test_02_sigset_search_multiprobe_gallery2 (  
    self )
```

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

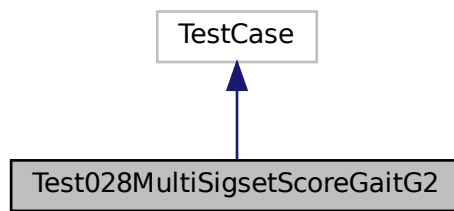
- [evaluation/full\\_evaluation.py](#)

## 7.51 Test028MultiSigsetScoreGaitG2 Class Reference

Inheritance diagram for Test028MultiSigsetScoreGaitG2:



Collaboration diagram for Test028MultiSigsetScoreGaitG2:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_gait\\_multiprobe\\_gallery2](#) (self)
- def [test\\_02\\_sigset\\_search\\_gait\\_multiprobe\\_gallery2](#) (self)

### 7.51.1 Member Function Documentation

#### 7.51.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

#### 7.51.1.2 test\_01\_sigset\_verify\_gait\_multiprobe\_gallery2()

```
def test_01_sigset_verify_gait_multiprobe_gallery2 (  
    self )
```

#### 7.51.1.3 test\_02\_sigset\_search\_gait\_multiprobe\_gallery2()

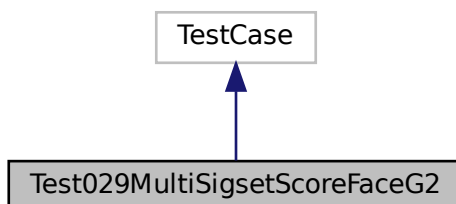
```
def test_02_sigset_search_gait_multiprobe_gallery2 (  
    self )
```

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

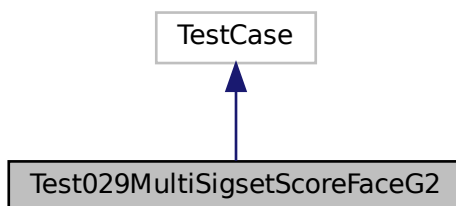
- [evaluation/full\\_evaluation.py](#)

## 7.52 Test029MultiSigsetScoreFaceG2 Class Reference

Inheritance diagram for Test029MultiSigsetScoreFaceG2:



Collaboration diagram for Test029MultiSigsetScoreFaceG2:



### Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_face\\_multiprobe\\_gallery2](#) (self)
- def [test\\_02\\_sigset\\_search\\_face\\_multiprobe\\_gallery2](#) (self)

### 7.52.1 Member Function Documentation

#### 7.52.1.1 setUpClass()

```
None setUpClass (  
    cls )
```



### 7.52.1.2 test\_01\_sigset\_verify\_face\_multiprobe\_gallery2()

```
def test_01_sigset_verify_face_multiprobe_gallery2 (
    self )
```

### 7.52.1.3 test\_02\_sigset\_search\_face\_multiprobe\_gallery2()

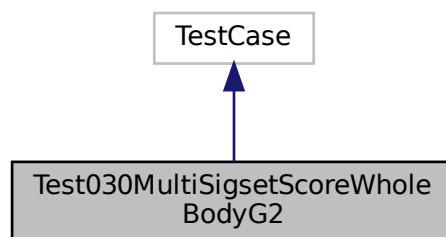
```
def test_02_sigset_search_face_multiprobe_gallery2 (
    self )
```

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

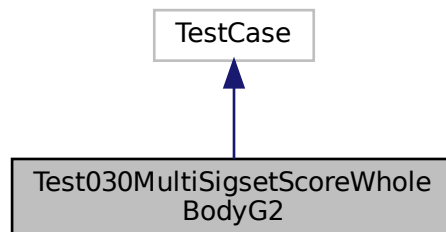
- [evaluation/full\\_evaluation.py](#)

## 7.53 Test030MultiSigsetScoreWholeBodyG2 Class Reference

Inheritance diagram for Test030MultiSigsetScoreWholeBodyG2:



Collaboration diagram for Test030MultiSigsetScoreWholeBodyG2:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_WB\\_multiprobe\\_gallery2](#) (self)
- def [test\\_02\\_sigset\\_search\\_WB\\_multiprobe\\_gallery2](#) (self)

### 7.53.1 Member Function Documentation

#### 7.53.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

#### 7.53.1.2 test\_01\_sigset\_verify\_WB\_multiprobe\_gallery2()

```
def test_01_sigset_verify_WB_multiprobe_gallery2 (  
    self )
```

#### 7.53.1.3 test\_02\_sigset\_search\_WB\_multiprobe\_gallery2()

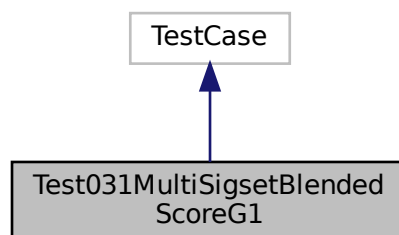
```
def test_02_sigset_search_WB_multiprobe_gallery2 (  
    self )
```

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

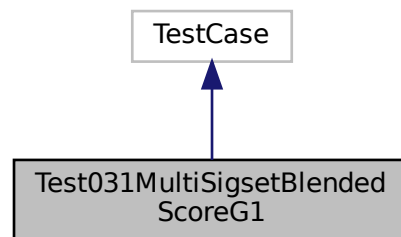
- [evaluation/full\\_evaluation.py](#)

## 7.54 Test031MultiSigsetBlendedScoreG1 Class Reference

Inheritance diagram for Test031MultiSigsetBlendedScoreG1:



Collaboration diagram for Test031MultiSigsetBlendedScoreG1:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_multiprobe\\_blended\\_gallery1](#) (self)
- def [test\\_02\\_sigset\\_search\\_multiprobe\\_blended\\_gallery1](#) (self)

### 7.54.1 Member Function Documentation

#### 7.54.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

#### 7.54.1.2 test\_01\_sigset\_verify\_multiprobe\_blended\_gallery1()

```
def test_01_sigset_verify_multiprobe_blended_gallery1 (  
    self )
```

#### 7.54.1.3 test\_02\_sigset\_search\_multiprobe\_blended\_gallery1()

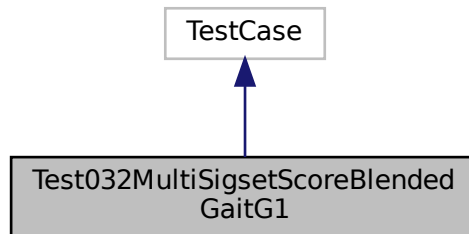
```
def test_02_sigset_search_multiprobe_blended_gallery1 (  
    self )
```

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

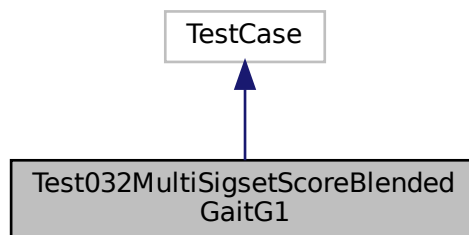
- [evaluation/full\\_evaluation.py](#)

## 7.55 Test032MultiSigsetScoreBlendedGaitG1 Class Reference

Inheritance diagram for Test032MultiSigsetScoreBlendedGaitG1:



Collaboration diagram for Test032MultiSigsetScoreBlendedGaitG1:



### Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_gait\\_multiprobe\\_blended\\_gallery1](#) (self)
- def [test\\_02\\_sigset\\_search\\_gait\\_multiprobe\\_blended\\_gallery1](#) (self)

### 7.55.1 Member Function Documentation

#### 7.55.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

### 7.55.1.2 test\_01\_sigset\_verify\_gait\_multiprobe\_blended\_gallery1()

```
def test_01_sigset_verify_gait_multiprobe_blended_gallery1 (
    self )
```

### 7.55.1.3 test\_02\_sigset\_search\_gait\_multiprobe\_blended\_gallery1()

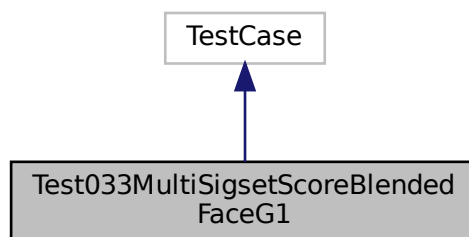
```
def test_02_sigset_search_gait_multiprobe_blended_gallery1 (
    self )
```

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

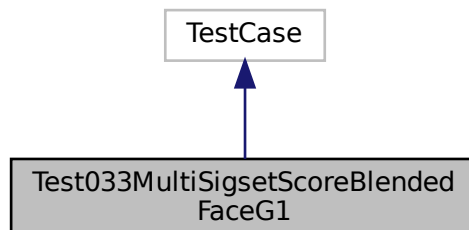
- [evaluation/full\\_evaluation.py](#)

## 7.56 Test033MultiSigsetScoreBlendedFaceG1 Class Reference

Inheritance diagram for Test033MultiSigsetScoreBlendedFaceG1:



Collaboration diagram for Test033MultiSigsetScoreBlendedFaceG1:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_face\\_multiprobe\\_blended\\_gallery1](#) (self)
- def [test\\_02\\_sigset\\_search\\_face\\_multiprobe\\_blended\\_gallery1](#) (self)

### 7.56.1 Member Function Documentation

#### 7.56.1.1 setUpClass()

```
None setUpClass (
    cls )
```

#### 7.56.1.2 test\_01\_sigset\_verify\_face\_multiprobe\_blended\_gallery1()

```
def test_01_sigset_verify_face_multiprobe_blended_gallery1 (
    self )
```

#### 7.56.1.3 test\_02\_sigset\_search\_face\_multiprobe\_blended\_gallery1()

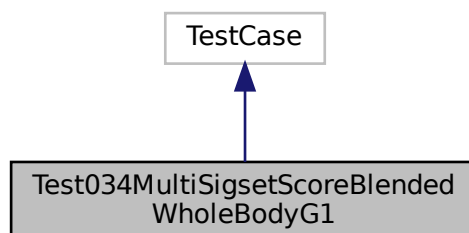
```
def test_02_sigset_search_face_multiprobe_blended_gallery1 (
    self )
```

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

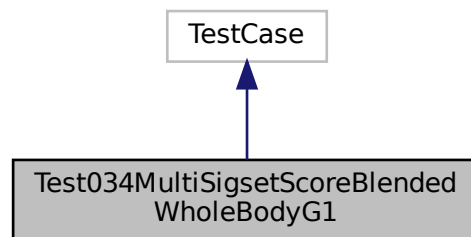
- [evaluation/full\\_evaluation.py](#)

## 7.57 Test034MultiSigsetScoreBlendedWholeBodyG1 Class Reference

Inheritance diagram for Test034MultiSigsetScoreBlendedWholeBodyG1:



Collaboration diagram for Test034MultiSigsetScoreBlendedWholeBodyG1:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_WB\\_multiprobe\\_blended\\_gallery1](#) (self)
- def [test\\_02\\_sigset\\_search\\_WB\\_multiprobe\\_blended\\_gallery1](#) (self)

## 7.57.1 Member Function Documentation

### 7.57.1.1 setUpClass()

```
None setUpClass (
    cls )
```

### 7.57.1.2 test\_01\_sigset\_verify\_WB\_multiprobe\_blended\_gallery1()

```
def test_01_sigset_verify_WB_multiprobe_blended_gallery1 (
    self )
```

### 7.57.1.3 test\_02\_sigset\_search\_WB\_multiprobe\_blended\_gallery1()

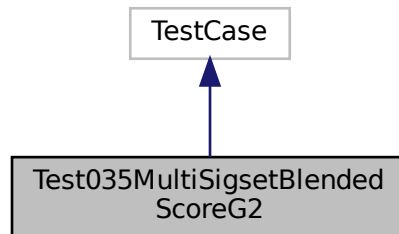
```
def test_02_sigset_search_WB_multiprobe_blended_gallery1 (
    self )
```

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

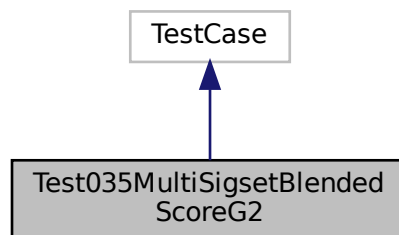
- [evaluation/full\\_evaluation.py](#)

## 7.58 Test035MultiSigsetBlendedScoreG2 Class Reference

Inheritance diagram for Test035MultiSigsetBlendedScoreG2:



Collaboration diagram for Test035MultiSigsetBlendedScoreG2:



### Public Member Functions

- None `setUpClass` (cls)
- def `test_01_sigset_verify_multiprobe_blended_gallery2` (self)
- def `test_02_sigset_search_multiprobe_blended_gallery2` (self)

### 7.58.1 Member Function Documentation

#### 7.58.1.1 setUpClass()

```

None setUpClass (
    cls )

```



### 7.58.1.2 test\_01\_sigset\_verify\_multiprobe\_blended\_gallery2()

```
def test_01_sigset_verify_multiprobe_blended_gallery2 (
    self )
```

### 7.58.1.3 test\_02\_sigset\_search\_multiprobe\_blended\_gallery2()

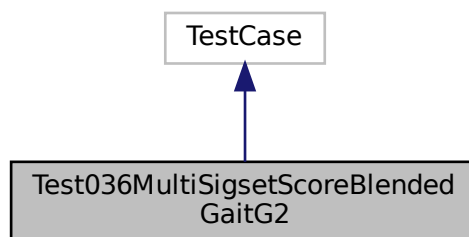
```
def test_02_sigset_search_multiprobe_blended_gallery2 (
    self )
```

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

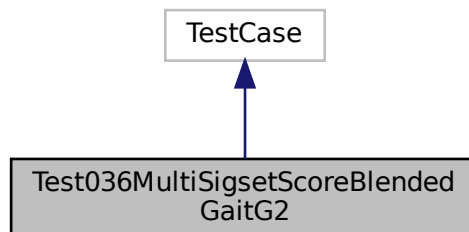
- [evaluation/full\\_evaluation.py](#)

## 7.59 Test036MultiSigsetScoreBlendedGaitG2 Class Reference

Inheritance diagram for Test036MultiSigsetScoreBlendedGaitG2:



Collaboration diagram for Test036MultiSigsetScoreBlendedGaitG2:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_gait\\_multiprobe\\_blended\\_gallery2](#) (self)
- def [test\\_02\\_sigset\\_search\\_gait\\_multiprobe\\_blended\\_gallery2](#) (self)

### 7.59.1 Member Function Documentation

#### 7.59.1.1 setUpClass()

```
None setUpClass (  
    cls )
```

#### 7.59.1.2 test\_01\_sigset\_verify\_gait\_multiprobe\_blended\_gallery2()

```
def test_01_sigset_verify_gait_multiprobe_blended_gallery2 (  
    self )
```

#### 7.59.1.3 test\_02\_sigset\_search\_gait\_multiprobe\_blended\_gallery2()

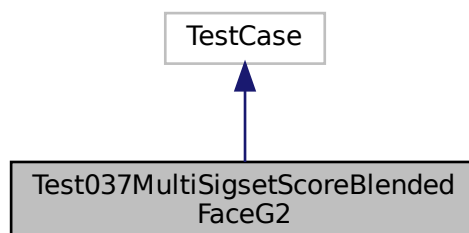
```
def test_02_sigset_search_gait_multiprobe_blended_gallery2 (  
    self )
```

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

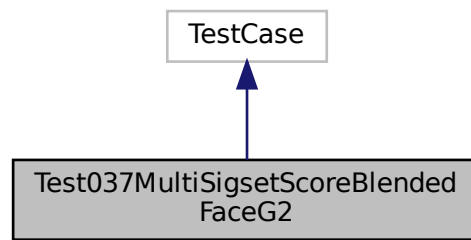
- [evaluation/full\\_evaluation.py](#)

## 7.60 Test037MultiSigsetScoreBlendedFaceG2 Class Reference

Inheritance diagram for Test037MultiSigsetScoreBlendedFaceG2:



Collaboration diagram for Test037MultiSigsetScoreBlendedFaceG2:



## Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_face\\_multiprobe\\_blended\\_gallery2](#) (self)
- def [test\\_02\\_sigset\\_search\\_face\\_multiprobe\\_blended\\_gallery2](#) (self)

### 7.60.1 Member Function Documentation

#### 7.60.1.1 setUpClass()

```
None setUpClass (
    cls )
```

#### 7.60.1.2 test\_01\_sigset\_verify\_face\_multiprobe\_blended\_gallery2()

```
def test_01_sigset_verify_face_multiprobe_blended_gallery2 (
    self )
```

#### 7.60.1.3 test\_02\_sigset\_search\_face\_multiprobe\_blended\_gallery2()

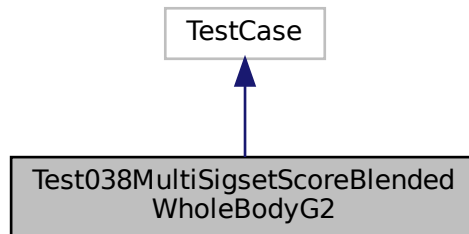
```
def test_02_sigset_search_face_multiprobe_blended_gallery2 (
    self )
```

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

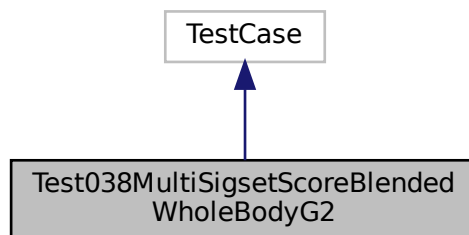
- [evaluation/full\\_evaluation.py](#)

## 7.61 Test038MultiSigsetScoreBlendedWholeBodyG2 Class Reference

Inheritance diagram for Test038MultiSigsetScoreBlendedWholeBodyG2:



Collaboration diagram for Test038MultiSigsetScoreBlendedWholeBodyG2:



### Public Member Functions

- None [setUpClass](#) (cls)
- def [test\\_01\\_sigset\\_verify\\_WB\\_multiprobe\\_blended\\_gallery2](#) (self)
- def [test\\_02\\_sigset\\_search\\_WB\\_multiprobe\\_blended\\_gallery2](#) (self)

### 7.61.1 Member Function Documentation

#### 7.61.1.1 setUpClass()

```

None setUpClass (
    cls )

```

### 7.61.1.2 test\_01\_sigset\_verify\_WB\_multiprobe\_blended\_gallery2()

```
def test_01_sigset_verify_WB_multiprobe_blended_gallery2 (
    self )
```

### 7.61.1.3 test\_02\_sigset\_search\_WB\_multiprobe\_blended\_gallery2()

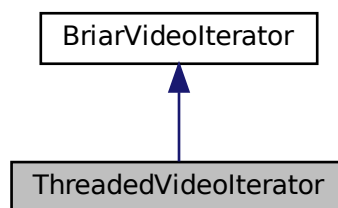
```
def test_02_sigset_search_WB_multiprobe_blended_gallery2 (
    self )
```

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

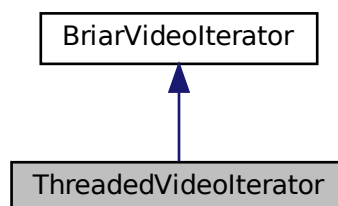
- [evaluation/full\\_evaluation.py](#)

## 7.62 ThreadedVideolterator Class Reference

Inheritance diagram for ThreadedVideolterator:



Collaboration diagram for ThreadedVideolterator:



## Public Member Functions

- `def __aiter__ (self)`
- `def __anext__ (self)`
- `def __init__ (self, filepath, start=None, stop=None, unit=None, debug\_empty=False, options=None)`
- `def __iter__ (self)`
- `def __len__ (self)`
- `def __next__ (self)`
- `def stop\_iteration (self, exception_type)`

## Public Attributes

- [cap](#)
- [debug\\_empty](#)
- [filepath](#)
- [fps](#)
- [frame\\_count](#)
- [frame\\_height](#)
- [frame\\_width](#)
- [i](#)
- [isOpened](#)
- [isStarted](#)
- [length](#)
- [msec](#)
- [options](#)
- [pos](#)
- [processed](#)
- [start\\_frame](#)
- [stop\\_frame](#)
- [stream](#)

## 7.62.1 Constructor & Destructor Documentation

### 7.62.1.1 `__init__()`

```
def __init__ (
    self,
    filepath,
    start = None,
    stop = None,
    unit = None,
    debug_empty = False,
    options = None )
```

The `__init__` function is called when the class is instantiated.

It sets up the instance of the class, and defines all its attributes.

The `__init__` function takes in arguments that are passed to it by whoever creates an instance of this class, and assigns these arguments to self variables so they can be used throughout this object.

```
:param self: Represent the instance of the class
:param filepath: Specify the path to the video file
:param start: Specify the start frame of the video
:param stop: Set the last frame to be read from the video
:param unit: Specify the unit of start and stop, choices: frame, time in seconds, NA (defaults to full video)
:param debug_empty: specified for creating a debug video iterator object that passes empty frames for testing
:return: Nothing
:doc-author: Joel Brogan, BRIAR team, Trelent
```

Reimplemented from [BriarVideoIterator](#).

## 7.62.2 Member Function Documentation

### 7.62.2.1 `__aiter__()`

```
def __aiter__ (
    self )
```

The `__aiter__` function is used to define an asynchronous iterator.

```
:param self: Refer to the current instance of a class
:return: The __iter__ function
:doc-author: Joel Brogan, BRIAR team, Trelent
```

Reimplemented from [BriarVideoIterator](#).

### 7.62.2.2 `__anext__()`

```
def __anext__ (
    self )
```

The `__anext__` function is the asynchronous iterator protocol.  
It allows you to use `async` for loops, which are a lot more efficient than regular for loops.  
The `__anext__` function should return an awaitable object that resolves to the next item in your sequence.

```
:param self: Represent the instance of the class
:return: A frame, which is a numpy array
:doc-author: Joel Brogan, BRIAR team, Trelent
```

Reimplemented from [BriarVideoIterator](#).

### 7.62.2.3 `__iter__()`

```
def __iter__ (
    self )
```

The `__iter__` function is called when an iterator is required for a container.  
This function should return a new iterator object that can iterate over all the objects in the container.  
For mappings, it should iterate over the keys of the container, and should also be made available as the method

```
:param self: Represent the instance of the class
:return: Self
:doc-author: Joel Brogan, BRIAR team, Trelent
```

Reimplemented from [BriarVideoIterator](#).

#### 7.62.2.4 `__len__()`

```
def __len__ (
    self )
```

The `__len__` function is used to determine the length of an object.  
For example, if you have a list with 5 items in it, calling `len(my_list)` will return 5.  
The `__len__` function is called when using the built-in `len()` function.

```
:param self: Allow an object to refer to itself
:return: The length of the iterator
:doc-author: Joel Brogan, BRIAR team, Trelent
```

Reimplemented from [BriarVideoIterator](#).

#### 7.62.2.5 `__next__()`

```
def __next__ (
    self )
```

The `__next__` function is called by the for loop to get each item from the iterator.  
The `__next__` function should raise a `StopIteration` exception when there are no more items in the container.

```
:param self: Represent the instance of the class
:return: A frame from the video
:doc-author: Joel Brogan, BRIAR team, Trelent
```

Reimplemented from [BriarVideoIterator](#).

#### 7.62.2.6 `stop_iteration()`

```
def stop_iteration (
    self,
    exception_type )
```

### 7.62.3 Member Data Documentation

#### 7.62.3.1 `cap`

`cap`



### 7.62.3.2 debug\_empty

debug\_empty

### 7.62.3.3 filepath

filepath

### 7.62.3.4 fps

fps

### 7.62.3.5 frame\_count

frame\_count

### 7.62.3.6 frame\_height

frame\_height

### 7.62.3.7 frame\_width

frame\_width

### 7.62.3.8 i

i

### 7.62.3.9 isOpened

isOpened

**7.62.3.10 isStarted**

isStarted

**7.62.3.11 length**

length

**7.62.3.12 msec**

msec

**7.62.3.13 options**

options

**7.62.3.14 pos**

pos

**7.62.3.15 processed**

processed

**7.62.3.16 start\_frame**

start\_frame

**7.62.3.17 stop\_frame**

stop\_frame

### 7.62.3.18 stream

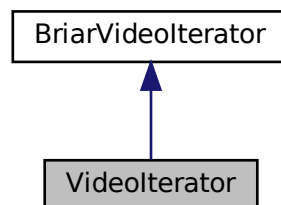
stream

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

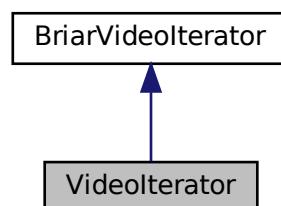
- [media/\\_\\_\\_init\\_\\_.py](#)

## 7.63 Videolterator Class Reference

Inheritance diagram for Videolterator:



Collaboration diagram for Videolterator:



### Public Member Functions

- `def \_\_aiter\_\_ (self)`
- `def \_\_anext\_\_ (self)`
- `def \_\_init\_\_ (self, filepath, start=None, stop=None, unit=None, debug\_empty=False, options=None)`
- `def \_\_iter\_\_ (self)`
- `def \_\_len\_\_ (self)`
- `def \_\_next\_\_ (self)`

## Public Attributes

- [cap](#)
- [debug\\_empty](#)
- [filepath](#)
- [fps](#)
- [frame\\_count](#)
- [frame\\_height](#)
- [frame\\_width](#)
- [i](#)
- [isOpened](#)
- [length](#)
- [msec](#)
- [pos](#)
- [processed](#)
- [start\\_frame](#)
- [stop\\_frame](#)

## 7.63.1 Constructor & Destructor Documentation

### 7.63.1.1 `__init__()`

```
def __init__ (
    self,
    filepath,
    start = None,
    stop = None,
    unit = None,
    debug_empty = False,
    options = None )
```

The `__init__` function is called when the class is instantiated.

It sets up the instance of the class, and defines all its attributes.

The `__init__` function takes in arguments that are passed to it by whoever creates an instance of this class, and assigns these arguments to self variables so they can be used throughout this object.

```
:param self: Represent the instance of the class
:param filepath: Specify the path to the video file
:param start: Specify the start frame of the video
:param stop: Set the last frame to be read from the video
:param unit: Specify the unit of start and stop, choices: frame, time in seconds, NA (defaults to full video)
:param debug_empty: specified for creating a debug video iterator object that passes empty frames for testing
:return: Nothing
:doc-author: Joel Brogan, BRIAR team, Trelent
```

Reimplemented from [BriarVideoIterator](#).

## 7.63.2 Member Function Documentation

### 7.63.2.1 `__aiter__()`

```
def __aiter__ (
    self )
```

The `__aiter__` function is used to define an asynchronous iterator.

```
:param self: Refer to the current instance of a class
:return: The __iter__ function
:doc-author: Joel Brogan, BRIAR team, Trelent
```

Reimplemented from [BriarVideolterator](#).

### 7.63.2.2 `__anext__()`

```
def __anext__ (
    self )
```

The `__anext__` function is the asynchronous iterator protocol.  
It allows you to use `async` for loops, which are a lot more efficient than regular for loops.  
The `__anext__` function should return an awaitable object that resolves to the next item in your sequence.

```
:param self: Represent the instance of the class
:return: A frame, which is a numpy array
:doc-author: Joel Brogan, BRIAR team, Trelent
```

Reimplemented from [BriarVideolterator](#).

### 7.63.2.3 `__iter__()`

```
def __iter__ (
    self )
```

The `__iter__` function is called when an iterator is required for a container.  
This function should return a new iterator object that can iterate over all the objects in the container.  
For mappings, it should iterate over the keys of the container, and should also be made available as the method

```
:param self: Represent the instance of the class
:return: Self
:doc-author: Joel Brogan, BRIAR team, Trelent
```

Reimplemented from [BriarVideolterator](#).

#### 7.63.2.4 `__len__()`

```
def __len__ (
    self )
```

Calculates the length of the iterator.

Returns:  
The length of the iterator.

Raises:  
Exception: If there is an error in calculating the length.

Reimplemented from [BriarVideoIterator](#).

#### 7.63.2.5 `__next__()`

```
def __next__ (
    self )
```

The `__next__` function is called by the for loop to get each item from the iterator.  
The `__next__` function should raise a `StopIteration` exception when there are no more items in the container.

:param self: Represent the instance of the class  
:return: A frame from the video  
:doc-author: Joel Brogan, BRIAR team, Trelent

Reimplemented from [BriarVideoIterator](#).

### 7.63.3 Member Data Documentation

#### 7.63.3.1 `cap`

`cap`

#### 7.63.3.2 `debug_empty`

`debug_empty`

### 7.63.3.3 filepath

filepath

### 7.63.3.4 fps

fps

### 7.63.3.5 frame\_count

frame\_count

### 7.63.3.6 frame\_height

frame\_height

### 7.63.3.7 frame\_width

frame\_width

### 7.63.3.8 i

i

### 7.63.3.9 isOpened

isOpened

### 7.63.3.10 length

length

**7.63.3.11 msec**

msec

**7.63.3.12 pos**

pos

**7.63.3.13 processed**

processed

**7.63.3.14 start\_frame**

start\_frame

**7.63.3.15 stop\_frame**

stop\_frame

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

- [media/\\_\\_\\_init\\_\\_\\_.py](#)



## Chapter 8

# File Documentation

### 8.1 `__init__.py` File Reference

#### Classes

- class [Rect](#)

#### Namespaces

- [briar](#)

#### Functions

- def [\\_wait\\_forever](#) (server)
- def [CLIServe](#) (serviceClass, add\_custom\_options=None)
- def [dyn\\_import](#) (name)
- def [get\\_process\\_number](#) ()
- def [get\\_thread\\_number](#) ()
- def [multiproc\\_serve](#) (serviceClass, options=None, serve\_port=None)
- def [parse\\_ports](#) (options)
- def [serve](#) (serviceClass, options=None, serve\_port=None)

#### Variables

- string [\\_\\_version\\_\\_](#) = '2.3.6'
- [\\_ONE\\_DAY](#) = datetime.timedelta(days=1)
- int [DEFAULT\\_MAX\\_MESSAGE\\_SIZE](#) = 64 \* 1024 \* 1024 \* 8
- string [DEFAULT\\_PORT](#) = "0.0.0.0:50051"
- string [DEFAULT\\_SERVE\\_PORT](#) = '[:]:50051'
- string [PLATFORM](#) = "UNKNOWN"

## 8.2 cli/\_\_\_init\_\_\_.py File Reference

### Namespaces

- [briar.cli](#)

*The modules contained in the CLI package each contain a function (detect, extract, enroll, etc.) which is a command within the broader CLI toolkit along with the assorted helper functions.*

## 8.3 cli/database/\_\_\_init\_\_\_.py File Reference

### Namespaces

- [briar.cli.database](#)

## 8.4 evaluation/\_\_\_init\_\_\_.py File Reference

### Namespaces

- [briar.evaluation](#)

### Functions

- def [runStages](#) (stages, report=False)

## 8.5 media/\_\_\_init\_\_\_.py File Reference

### Classes

- class [BriarProgress](#)
- class [BriarVideoIterator](#)
- class [ImageIterator](#)
- class [MediaSetIterator](#)
- class [ThreadedVideoIterator](#)
- class [VideoIterator](#)

### Namespaces

- [briar.media](#)

## Functions

- def [aenumerate](#) (asequence, start=0)
- def [decodeMedia](#) (media\_pb, newsource=None)  
*Convert protobuf media into a numpy array.*
- def [enroll\\_frames\\_iter](#) (database\_name, video, detect\_options=None, extract\_options=None, enroll\_options=None, det\_list\_list=None, whole\_image=False, request\_start=-1)  
*Iterates the paths in the media file list, loading them one by one and yielding grpc enroll requests.*
- def [enroll\\_frames\\_iter\\_async](#) (database\_name, video, detect\_options=None, extract\_options=None, enroll\_options=None, det\_list\_list=None, whole\_image=False, request\_start=-1)  
*Asynchronously iterates the paths in the media file list, loading them one by one and yielding grpc enroll requests.*
- def [file\\_iter](#) (list[str] media\_files, briar\_pb2.DetectionOptions clientoptions=None, dict options\_dict=None, str database\_name=None, bool verbose=False, int request\_start=-1, callable requestConstructor=None)
- def [frame\\_iter](#) (media\_iterator, clientoptions=None, dict options\_dict={}, database\_name=None, det\_list\_list=None, whole\_image=False, request\_start=-1, requestConstructor=None)
- def [ImageGenerator](#) (filepath, start=None, stop=None, unit=None, options=None)
- def [isFinalFrame](#) (request)
- def [single\\_frame\\_generate](#) (frame, frame\_num, itype, filepath, start\_frame, stop\_frame, fps, video\_length, clientoptions=None, dict options\_dict={}, database\_name=None, det\_list\_list=None, whole\_image=False, request\_start=-1, frame\_load\_time\_start=-1, file\_level\_client\_time\_end=-1, requestConstructor=None)

## 8.6 sigset/ \_\_init\_\_.py File Reference

### Namespaces

- [briar.sigset](#)

## 8.7 timing/ \_\_init\_\_.py File Reference

### Namespaces

- [briar.timing](#)

## Functions

- def [end\\_duration](#) (reply)
- def [generate\\_progress](#) (frame\_id, media)
- def [loadDurationsFolder](#) (durations\_directory)
- def [parseDurations](#) (durationsperfile\_dictionary)
- def [print\\_duration](#) (name, duration)
- def [print\\_durations](#) (durations)
- def [save\\_durations](#) (media\_file, durations\_list, options, operation, modality=None)
- def [start\\_duration](#) (request, reply)
- def [timeElapsed](#) (duration)
- def [timestamp](#) ()

## Variables

- string [DURATION\\_FILE\\_EXT](#) = ".durations"

## 8.8 `__main__.py` File Reference

### Namespaces

- [briar.\\_\\_main\\_\\_](#)

## 8.9 `briar_cli.py` File Reference

### Namespaces

- [briar.briar\\_cli](#)

*Created on 2021 at Oak Ridge National Laboratory.*

### Functions

- def [briar\\_command\\_line](#) ()  
*Entry point for the CLI - switches on the first command line argument (such as 'status', 'detect', etc) and builds the parser and help messages based upon the callback defined in COMMANDS.*
- def [briar\\_database\\_command\\_line](#) ()  
*Entry point for the Database CLI - switches on the second command line argument (such as 'delete', 'merge', etc) and builds the parser and help messages based upon the callback defined in COMMANDS.*
- def [briar\\_test\\_command\\_line](#) ()  
*Entry point for the Test CLI - switches on the second command line argument (such as 'delete', 'merge', etc) and builds the parser and help messages based upon the callback defined in COMMANDS.*
- def [incomplete](#) ()

### Variables

- dictionary [COMMANDS](#)
- dictionary [DATABASE\\_COMMANDS](#)
- string [DETECTION\\_FILE\\_EXT](#) = ".detection"
- int [FACE\\_COUNT](#) = 0
- string [MATCHES\\_FILE\\_EXT](#) = '.matches'
- string [TEMPLATE\\_FILE\\_EXT](#) = '.template'

## 8.10 `briar_client.py` File Reference

### Classes

- class [BriarClient](#)

*Provide a client to a BRIAR service.*

### Namespaces

- [briar.briar\\_client](#)

*Copyright 2021 Oak Ridge National Laboratory.*

## Functions

- def [\\_initialize\\_worker](#) (server\_address, proc\_number, thread\_number, count\_q)
- def [\\_shutdown\\_worker](#) ()

## Variables

- [\\_client\\_identifier\\_singleton](#) = None
- [\\_worker\\_channel\\_singleton](#) = None
- [\\_worker\\_port\\_singleton](#) = None
- [\\_worker\\_proccess\\_position\\_singleton](#) = None
- [\\_worker\\_stub\\_singleton](#) = None
- [\\_worker\\_thread\\_position\\_singleton](#) = None

## 8.11 briar\_media.py File Reference

### Classes

- class [BriarMedia](#)
- class [MediaStream](#)

### Namespaces

- [briar.briar\\_media](#)

*Defines a media class which acts as a wrapper for image and video files.*

### Functions

- def [briar\\_media\\_from\\_pb2](#) (pb2\_object)
- def [briar\\_media\\_to\\_pb2](#) (media)
- def [load\\_media\\_from\\_folder](#) (folder\_path, recursive=False)
- def [load\\_media\\_from\\_image](#) (image\_path)
- def [load\\_media\\_from\\_numpy](#) (numpy\_array)

## 8.12 cli/connection.py File Reference

### Namespaces

- [briar.cli.connection](#)

### Functions

- def [addConnectionOptions](#) (parser)  
*Accumulatively add options for connecting to the Briar API service.*

## Variables

- int `DEFAULT_MAX_ASYNC` = 8
- int `DEFAULT_MAX_MESSAGE_SIZE` = 64 \* 1024 \* 1024 \* 8

## 8.13 cli/database/checkpoint.py File Reference

### Namespaces

- `briar.cli.database.checkpoint`

### Functions

- def `database_checkpoint` (options=None, args=None, input\_command=None, ret=False)

## 8.14 cli/database/checkpoint\_subject.py File Reference

### Namespaces

- `briar.cli.database.checkpoint_subject`

### Functions

- def `database_checkpoint_subject` (options=None, args=None, input\_command=None, ret=False)
- def `parseDatabaseCheckpointSubjectOptions` (inputCommand=None)  
*Generate options for getting information about a pre-existing database and parse command line arguments into an API call.*

## 8.15 cli/database/common.py File Reference

### Namespaces

- `briar.cli.database.common`

### Functions

- def `db_no_exist` (name)

## 8.16 cli/database/compute\_scores.py File Reference

### Namespaces

- `briar.cli.database.compute_scores`

## Functions

- def [addDatabaseComputeScore\\_options2proto](#) (options)
- def [addDatabaseComputeScoreOptions](#) (parser)  
*Add options for search of a database using a database.*
- def [database\\_compute\\_verify](#) (options=None, args=None, input\_command=None, ret=False)  
*Using the options specified in the command line, runs a batch verification between the specified databases using specified probe database.*
- def [parseDatabaseComputeScoreOptions](#) (inputCommand=None)  
*Generate options for matching databases against other databases, and parse command line arguments into the API call.*

## 8.17 cli/database/compute\_search.py File Reference

### Namespaces

- [briar.cli.database.compute\\_search](#)

### Functions

- def [database\\_compute\\_search](#) (options=None, args=None, input\_command=None, ret=False)  
*Using the options specified in the command line, runs a search within the specified database using specified probe database.*

## 8.18 cli/database/create.py File Reference

### Namespaces

- [briar.cli.database.create](#)

### Functions

- def [database\\_create](#) (options=None, args=None, input\_command=None, ret=False)

## 8.19 cli/database/delete.py File Reference

### Namespaces

- [briar.cli.database.delete](#)

### Functions

- def [database\\_delete](#) (options=None, args=None, input\_command=None, ret=False)
- def [parseDatabaseDeleteOptions](#) (inputCommand=None)  
*Generate options for Deleting a pre-existing database and parse command line arguments into API call.*

## 8.20 cli/database/finalize.py File Reference

### Namespaces

- [briar.cli.database.finalize](#)

### Functions

- def [database\\_finalize](#) (options=None, args=None, input\_command=None, ret=False)  
*Parses the command line options and saves the database to disk.*
- def [finalizeParseOptions](#) (inputCommand=None)  
*Generate options for running 'finalize' (saving the loaded databases) and parse command line arguments into them.*

## 8.21 cli/finalize.py File Reference

### Namespaces

- [briar.cli.finalize](#)

### Functions

- def [database\\_finalize](#) (options=None, args=None, input\_command=None, ret=False)  
*Parses the command line options and saves the database to disk.*
- def [finalizeParseOptions](#) (inputCommand=None)  
*Generate options for running 'finalize' (saving the loaded databases) and parse command line arguments into them.*

## 8.22 cli/database/info.py File Reference

### Namespaces

- [briar.cli.database.info](#)

### Functions

- def [database\\_info](#) (options=None, args=None, input\_command=None, ret=False)
- def [parseDatabaseInfoOptions](#) (inputCommand=None)  
*Generate options for getting information about a pre-existing database and parse command line arguments into an API call.*

## 8.23 cli/database/list.py File Reference

### Namespaces

- [briar.cli.database.list](#)



## Functions

- def [database\\_list](#) (options=None, args=None, input\_command=None, ret=False)
- def [parseDatabaseListOptions](#) (inputCommand=None)  
*Generate options for listing all pre-existing databases and parse command line arguments into them.*

## 8.24 cli/database/list\_entries.py File Reference

### Namespaces

- [briar.cli.database.list\\_entries](#)

## Functions

- def [database\\_list\\_entries](#) (options=None, args=None, input\_command=None, ret=False)
- def [parseDatabaseListEntriesOptions](#) (inputCommand=None)  
*Generate options for Listing entries within a pre-existing database and parse command line arguments into an API call.*

## 8.25 cli/database/load.py File Reference

### Namespaces

- [briar.cli.database.load](#)

## Functions

- def [database\\_load](#) ()

## 8.26 cli/database/merge.py File Reference

### Namespaces

- [briar.cli.database.merge](#)

## Functions

- def [database\\_merge](#) (options=None, args=None, input\_command=None, ret=False)
- def [parseDatabaseMergeOptions](#) (inputCommand=None)  
*Generate options for merging databases and parse command line arguments into the API call.*

## 8.27 cli/database/move\_entry.py File Reference

### Namespaces

- [briar.cli.database.move\\_entry](#)

### Functions

- def [database\\_move\\_entry](#) (options=None, args=None, input\_command=None, ret=False, client=None)
- def [parseDatabaseMoveEntryOptions](#) (inputCommand=None)

*Generate options for moving database entries and parse command line arguments into the API call.*

## 8.28 cli/database/refresh.py File Reference

### Namespaces

- [briar.cli.database.refresh](#)

### Functions

- def [database\\_checkpoint](#) (options=None, args=None, input\_command=None, ret=False)
- def [database\\_refresh](#) (options=None, args=None, input\_command=None, ret=False)
- def [parseDatabaseRefreshOptions](#) (inputCommand=None)

*Generate options for getting information about a pre-existing database and parse command line arguments into an API call.*

## 8.29 cli/database/rename.py File Reference

### Namespaces

- [briar.cli.database.rename](#)

### Functions

- def [database\\_load](#) ()
- def [database\\_rename](#) (options=None, args=None, input\_command=None, ret=False)
- def [parseDatabaseRenameOptions](#) (inputCommand=None)

*Generate options for Renaming a pre-existing database to a new name and parse command line arguments into API call.*

## 8.30 cli/database/retrieve.py File Reference

### Namespaces

- [briar.cli.database.retrieve](#)

## Functions

- def [database\\_retrieve](#) (options=None, args=None, input\_command=None, ret=False)
- def [parseDatabaseRetrieveOptions](#) (inputCommand=None)  
*Generate options for retrieving a pre-existing database and parse command line arguments API call.*

## 8.31 cli/detect.py File Reference

### Namespaces

- [briar.cli.detect](#)

## Functions

- def [addDetectorOptions](#) (parser)  
*Add options for running detections to the parser.*
- def [addTrackingOptions](#) (parser)  
*Add options for running detections to the parser.*
- def [detect](#) (options=None, args=None, input\_command=None, ret=False)  
*Using the options specified in the command line, runs a detection on the specified files.*
- def [detect\\_options2proto](#) (options)
- def [detectParseOptions](#) (inputCommand=None)  
*Generate options for running detections and parse command line arguments into them.*
- def [detectRequestConstructor](#) (briar\_pb2.BriarMedia media, briar\_pb2.BriarDurations durations, options\_↵ dict={}, det\_list\_list=None, str database\_name=None)
- def [get\\_detection\\_path](#) (media\_file, options, i, modality=None, media\_id=None)
- def [isFinalReply](#) (srv\_pb2.DetectReply reply)
- def [save\\_detections](#) (media\_file, reply, options, i, modality=None, media\_id=None)
- def [tracking\\_options2proto](#) (options)

### Variables

- string [DETECTION\\_FILE\\_EXT](#) = ".detection"

## 8.32 cli/enhance.py File Reference

### Namespaces

- [briar.cli.enhance](#)

## Functions

- def [addEnhanceOptions](#) (parser)  
*Add options for running detections to the parser.*
- def [enhance](#) (options=None, args=None, input\_command=None, ret=False)  
*Using the options specified in the command line, runs a detection on the specified files.*
- def [enhance\\_options2proto](#) (options)
- def [enhanceParseOptions](#) (inputCommand=None)  
*Generate options for running enhancement and parse command line arguments into them.*
- def [enhanceRequestConstructor](#) (briar\_pb2.BriarMedia media, briar\_pb2.BriarDurations durations, options\_↵ \_dict={}, det\_list\_list=None, str database\_name=None)
- def [save\\_Enhancement](#) (media\_file, reply, options, i, modality=None)

## Variables

- string `ENHANCE_FILE_EXT` = '.enh'

## 8.33 cli/enroll.py File Reference

### Namespaces

- `briar.cli.enroll`

### Functions

- def `addEnrollOptions` (parser)  
*Add options for enrollment into a database.*
- def `enroll` (options=None, args=None, input\_command=None, ret=False)  
*Using the options specified in the command line, runs an enroll on the specified files.*
- def `enroll_options2proto` (options)
- def `enrollParseOptions` (inputCommand=None)  
*Generate options for running enrollments and parse command line arguments into them.*
- def `enrollRequestConstructor` (briar\_pb2.BriarMedia media, briar\_pb2.BriarDurations durations, options\_↵ dict={}, det\_list\_list=None, str database\_name=None)

## 8.34 cli/extract.py File Reference

### Namespaces

- `briar.cli.extract`

### Functions

- def `addExtractOptions` (parser)  
*Add options for extractions to the parser.*
- def `extract` (options=None, args=None, inputCommand=None, ret=False)  
*Using the options specified in the command line, runs an extract on the specified files.*
- def `extract_options2proto` (options)
- def `extractParseOptions` (inputCommand=None)  
*Generate options for running extracts and parse command line arguments into them.*
- def `extractRequestConstructor` (briar\_pb2.BriarMedia media, briar\_pb2.BriarDurations durations, options\_↵ dict={}, det\_list\_list=None, str database\_name=None)
- def `save_extractions` (media\_file, templates, options, i, modality=None, media\_id=None)

### Variables

- string `TEMPLATE_FILE_EXT` = '.template'

## 8.35 cli/media.py File Reference

### Namespaces

- [briar.cli.media](#)

### Functions

- def [addMediaOptions](#) (parser)  
*Add options for running detections to the parser.*
- def [collect\\_files](#) (args, options, extension=None)  
*Take the paths specified by 'args' and find all the media files that they define: folders will be searched for all media files contained inside.*
- def [hasExtension](#) (f, extension)

### Variables

- int [DEFAULT\\_MAX\\_SIZE](#) = 1920

## 8.36 cli/search.py File Reference

### Namespaces

- [briar.cli.search](#)

### Functions

- def [addSearchOptions](#) (parser)  
*Add options for search of a database.*
- def [search](#) (options=None, args=None, input\_command=None, ret=False)  
*Using the options specified in the command line, runs a search within the specified database using specified probe template(s).*
- def [search\\_options2proto](#) (options)
- def [searchParseOptions](#) (inputCommand=None)  
*Generate options for running searches and parse command line arguments into them.*
- def [searchRequestConstructor](#) (briar\_pb2.BriarMedia media, briar\_pb2.BriarDurations durations, options\_↵ dict={}, det\_list\_list=None, str database\_name=None)

### Variables

- string [MATCHES\\_FILE\\_EXT](#) = '.matches'

## 8.37 cli/sigset.py File Reference

### Namespaces

- [briar.cli.sigset](#)

## Functions

- def [checkpoint\\_subject\\_threaded](#) (obj)
- def [df\\_batch\\_consumer](#) (batch\_queue, identifier, port, server\_configuration, id\_queue, progress\_queue)
- def [df\\_batch\\_generator](#) (list\_of\_dfs, start, dataset\_dir, detect\_options, extract\_options, enroll\_options, options)
- def [df\\_batch\\_producer](#) (batch\_queue, identifier, progress\_position, list\_of\_dataframes, start, dataset\_dir, detect\_options, extract\_options, enroll\_options, options)
- def [df\\_row\\_generator](#) (rowiter, start, dataset\_dir, detect\_options, extract\_options, enroll\_options, options)
- def [enroll\\_call\\_threaded](#) (input)
- def [inner\\_pool\\_mapper](#) (batch\_obj, local\_pool, progress\_queue, port)
- def [parseSigsetEnrollOptions](#) (inputCommand=None)
- def [parseSigsetStatsOptions](#) (inputCommand=None)
- def [progress\\_consumer](#) (progress\_queue, progress\_position, total\_len, options)
- def [sigset\\_enroll](#) (input\_command=None)
- def [sigset\\_stats](#) (options=None, args=None)

## Variables

- [my\\_pool](#) = None
- [proc\\_number](#) = None
- [service\\_address\\_number](#) = None

## 8.38 cli/status.py File Reference

### Namespaces

- [briar.cli.status](#)

### Functions

- def [get\\_service\\_configuration](#) (options=None, args=None, input\_command=None, ret=False)
- def [print\\_service\\_configuration](#) (options=None, args=None)
- def [status](#) (options=None, args=None, input\_command=None, ret=False)  
*Connects to the server and gets status information.*
- def [statusParseOptions](#) (inputCommand=None)  
*Generate options for getting status and parse command line arguments into them.*

## 8.39 cli/test.py File Reference

### Classes

- class [BriarTest](#)
- class [BriarTestResult](#)
- class [DatabaseTest](#)
- class [DetectTest](#)
- class [EnrollTest](#)
- class [ExtractTest](#)
- class [Fore](#)

## Namespaces

- [briar.cli.test](#)

## Functions

- def [detection\\_output\\_tests](#) (detection\_obj\_loaded, testimage, return\_media)
- def [extraction\\_output\\_tests](#) (template\_obj\_loaded, testimage, return\_media)

## Variables

- bool [use\\_colorama](#) = False

## 8.40 cli/track.py File Reference

### Namespaces

- [briar.cli.track](#)

### Functions

- def [get\\_tracklet\\_path](#) (media\_file, options, i, modality=None, media\_id=None)
- def [save\\_tracklets](#) (media\_file, tracklets, options, i, verbose=False, modality=None, media\_id=None)
- def [track](#) (options=None, args=None, input\_command=None, ret=False)  
*Using the options specified in the command line, runs a detection on the specified files.*
- def [trackRequestConstructor](#) (briar\_pb2.BriarMedia media, briar\_pb2.BriarDurations durations, options\_↵ dict={}, det\_list\_list=None, str database\_name=None)

### Variables

- string [TRACKLET\\_FILE\\_EXT](#) = ".tracklet"

## 8.41 cli/verify.py File Reference

### Namespaces

- [briar.cli.verify](#)

### Functions

- def [addVerifyOptions](#) (parser)  
*Add options for verification to the parser.*
- def [save\\_verifications](#) (media\_file, reply, options, i, modality=None, media\_id=None)
- def [verify](#) (options=None, args=None, input\_command=None, ret=False)  
*Using the options specified in the command line, runs an extract on the specified files.*
- def [verify\\_options2proto](#) (options)
- def [verifyParseOptions](#) (inputCommand=None)  
*Generate options for running verifications and parse command line arguments into them.*

## Variables

- string [VERIFICATION\\_FILE\\_EXT](#) = ".verification"

## 8.42 cli/viz.py File Reference

### Namespaces

- [briar.cli.viz](#)

### Functions

- def [viz](#) ()  
*Using the options specified in the command line, runs visualization on the specified files.*
- def [vizParseOptions](#) ()  
*Generate options for running detections and parse command line arguments into them.*

## 8.43 evaluation/full\_evaluation.py File Reference

### Classes

- class [Test000InitialConfig](#)
- class [Test001SigsetEnrollProbe](#)
- class [Test002ProbeDatabaseMerge](#)
- class [Test003SigsetEnrollMultiProbe](#)
- class [Test004MultiProbeDatabaseMerge](#)
- class [Test005SigsetEnrollGalleries](#)
- class [Test006GalleryDatabaseMerge](#)
- class [Test007SigsetScoreG1](#)
- class [Test008SigsetScoreGaitG1](#)
- class [Test009SigsetScoreFaceG1](#)
- class [Test010SigsetScoreWholeBodyG1](#)
- class [Test011SigsetScoreG2](#)
- class [Test012SigsetScoreGaitG2](#)
- class [Test013SigsetScoreFaceG2](#)
- class [Test014SigsetScoreWholeBodyG2](#)
- class [Test015SigsetBlendedScoreG1](#)
- class [Test016SigsetScoreBlendedGaitG1](#)
- class [Test017SigsetScoreBlendedFaceG1](#)
- class [Test018SigsetScoreBlendedWholeBodyG1](#)
- class [Test019SigsetBlendedScoreG2](#)
- class [Test020SigsetScoreBlendedGaitG2](#)
- class [Test021SigsetScoreBlendedFaceG2](#)
- class [Test022SigsetScoreBlendedWholeBodyG2](#)
- class [Test023MultiSigsetScoreG1](#)
- class [Test023SigsetSearchOutputFormatting](#)
- class [Test024MultiSigsetScoreGaitG1](#)
- class [Test024SigsetVerifyOutputFormatting](#)
- class [Test025MultiSigsetScoreFaceG1](#)



- class [Test026MultiSigsetScoreWholeBodyG1](#)
- class [Test027MultiSigsetScoreG2](#)
- class [Test028MultiSigsetScoreGaitG2](#)
- class [Test029MultiSigsetScoreFaceG2](#)
- class [Test030MultiSigsetScoreWholeBodyG2](#)
- class [Test031MultiSigsetBlendedScoreG1](#)
- class [Test032MultiSigsetScoreBlendedGaitG1](#)
- class [Test033MultiSigsetScoreBlendedFaceG1](#)
- class [Test034MultiSigsetScoreBlendedWholeBodyG1](#)
- class [Test035MultiSigsetBlendedScoreG2](#)
- class [Test036MultiSigsetScoreBlendedGaitG2](#)
- class [Test037MultiSigsetScoreBlendedFaceG2](#)
- class [Test038MultiSigsetScoreBlendedWholeBodyG2](#)

## Namespaces

- [briar.evaluation.full\\_evaluation](#)

## Functions

- def [compute\\_search](#) (probe\_db\_name, gal\_db\_name, probe\_sigset\_path, output\_path, modality=None, blended=False)
- def [compute\\_verify](#) (probe\_db\_name, gal\_db\_name, probe\_sigset\_path, gal\_sigset\_path, output\_path, csv↵\_path, modality=None, blended=False)
- def [get\\_info](#) (self, db\_name)
- def [get\\_multi\\_info](#) (self, base\_db\_name)
- def [merge\\_dbs](#) (self, db\_name)
- def [run\\_on\\_multi](#) (self, base\_db\_name, mapped\_function)
- def [runall](#) ()
- None [setUpClass\\_main](#) (cls)
- def [setUpModule](#) ()

## Variables

- string [args\\_string](#) = "--progress "
- string [blended\\_gallery\\_1\\_filename](#) = "sigsets\_gallery/Blended\_Gallery\_1.xml"
- string [blended\\_gallery\\_2\\_filename](#) = "sigsets\_gallery/Blended\_Gallery\_2.xml"
- [category](#)
- string [database\\_blended\\_gallery\\_1\\_name](#) = 'db\_eval\_phase2\_blended\_gallery\_1'
- string [database\\_blended\\_gallery\\_2\\_name](#) = 'db\_eval\_phase2\_blended\_gallery\_2'
- string [database\\_gallery\\_1\\_name](#) = 'db\_eval\_phase2\_gallery\_1'
- string [database\\_gallery\\_2\\_name](#) = 'db\_eval\_phase2\_gallery\_2'
- string [database\\_multi\\_probe\\_name](#) = 'db\_eval\_phase2\_multisubject\_probe'
- string [database\\_probe\\_name](#) = 'db\_eval\_phase2\_probe'
- [DATABASE\\_SUFFIX\\_FLAG](#) = os.getenv('BRIAR\_DATABASE\_SUFFIX\_FLAG')
- [DATASET\\_DIR](#) = os.getenv('BRIAR\_DATASET\_DIR')
- string [enroll\\_args](#) = "--auto-create-database "
- [EVAL\\_PHASE](#) = os.getenv('BRIAR\_EVAL\_PHASE')
- [EVALUATION\\_DIR](#) = os.getenv('BRIAR\_EVALUATION\_DIR')
- [EVALUATION\\_MULTISUBJECT\\_DIR](#) = os.getenv('BRIAR\_MULTISUBJECT\_EVALUATION\_DIR')
- [gallery1\\_blended\\_sigset](#)
- [gallery1\\_sigset](#)

- [gallery2\\_blended\\_sigset](#)
- [gallery2\\_sigset](#)
- [gallery\\_1\\_blended\\_sigset\\_path](#)
- string [gallery\\_1\\_filename](#) = "sigsets\_gallery/Gallery\_1.xml"
- [gallery\\_1\\_sigset\\_path](#)
- [gallery\\_2\\_blended\\_sigset\\_path](#)
- string [gallery\\_2\\_filename](#) = "sigsets\_gallery/Gallery\_2.xml"
- [gallery\\_2\\_sigset\\_path](#)
- [generate\\_report](#) = os.environ.get('REPORT', False)
- string [media\\_args](#) = "--no-save "
- [module](#)
- string [multisubject\\_probe\\_filename](#) = "sigsets\_multiperson/Probe\_BTS\_briar-rd\_multi.xml"
- int [number\\_of\\_partitions](#) = 1
- [OUTPUT\\_DIR](#) = os.getenv('BRIAR\_EVALUATION\_OUTPUT\_DIR')
- list [port\\_list](#) = []
- string [probe\\_filename](#) = "sigsets\_main/Probe\_BTS\_briar-rd\_ALL.xml"
- [probe\\_multisubject\\_sigset](#)
- [probe\\_multisubject\\_sigset\\_path](#)
- [probe\\_sigset](#)
- [probe\\_sigset\\_path](#)
- bool [requires\\_database\\_merge](#) = False
- [run\\_multisubject\\_evaluation](#)
- [RUN\\_STAGES](#) = os.getenv('RUN\_STAGES', None)
- [sortTestMethodsUsing](#)
- list [stages\\_temp](#) = []
- [USE\\_SINGLE\\_SUBJECT](#) = os.getenv('BRIAR\_USE\_SINGLE\_SUBJECT')
- [USES\\_FRONTEND\\_MERGING](#) = os.getenv('BRIAR\_USE\_FRONTEND\_MERGING')

## 8.44 evaluation/stage1\_probe\_enroll.py File Reference

### Namespaces

- [briar.evaluation.stage1\\_probe\\_enroll](#)

### Variables

- [generate\\_report](#) = os.environ.get('REPORT', False)
- [main](#) = unittest.TestProgram

## 8.45 evaluation/stage2.1\_gallery1\_simple\_enroll.py File Reference

### Namespaces

- [briar.evaluation.stage2](#)

### Variables

- [generate\\_report](#) = os.environ.get('REPORT', False)
- [main](#) = unittest.TestProgram

## 8.46 evaluation/stage2.2\_gallery2\_simple\_enroll.py File Reference

### Namespaces

- [briar.evaluation.stage2](#)

## 8.47 evaluation/stage2.3\_gallery1\_blended\_enroll.py File Reference

### Namespaces

- [briar.evaluation.stage2](#)

## 8.48 evaluation/stage2.4\_gallery2\_blended\_enroll.py File Reference

### Namespaces

- [briar.evaluation.stage2](#)

## 8.49 evaluation/stage3\_result\_scoring.py File Reference

### Namespaces

- [briar.evaluation.stage3\\_result\\_scoring](#)

### Variables

- [generate\\_report](#) = os.environ.get('REPORT', False)
- [main](#) = unittest.TestProgram

## 8.50 grpc\_json.py File Reference

### Classes

- class [GrpcDecoder](#)  
*Object which extends the JSONDecoded to allow it to read saved gRPC files.*
- class [GrpcEncoder](#)  
*Encoder class which extends the normal JSON encoder to allow the encoding of gRPC objects.*

### Namespaces

- [briar.grpc\\_json](#)  
*I couldn't find a good library to write gRPC/protobuf objects to disk so this is my own implementation of a generalized json converter for said objects.*

## Functions

- def [dict\\_to\\_proto\\_obj](#) (obj\_dict, options=None)  
*Take the object dictionary, read the dict which is saved in in the 'class' key, and initialize it with values stored in the dictionary's key/value pairs.*
- def [load](#) (load\_path, options=None)  
*Load the json file at the given directory, reloading dictionaries with "\_\_class\_\_" fields into the specified objects and initializing them with values defined by key/value pairs within the dictionary.*
- def [proto\\_obj\\_to\\_dict](#) (obj, options=None)  
*Takes a general gRPC/protobuf object, eliminates the unnecessary fields, and stores the data in a dict.*
- def [save](#) (json\_obj, save\_path, options=None)  
*Save a list or dictionary containing protobuf classes to a json file.*

## Variables

- list [ATTRIB\\_IGNORE](#)

## 8.51 media/VideoStream.py File Reference

### Classes

- class [FileVideoStream\\_cv2](#)
- class [FileVideoStream\\_imageio](#)

### Namespaces

- [briar.media.VideoStream](#)

## 8.52 media/visualize.py File Reference

### Classes

- class [match\\_matrix\\_visualizer](#)

### Namespaces

- [briar.media.visualize](#)

## Functions

- def [decode\\_track](#) (tracklet, framenum=None, newsource=None)
- def [get\\_frame](#) (vidfile)
- def [playVideo](#) (vidfiles, titles=None, attributes=None, isvideo=True)
- def [update\\_annot](#) (ind, visualizer, pltloc, playvid=False)
- def [update\\_annot\\_filename\\_only](#) (ind, visualizer, pltloc)
- def [visualize\\_detection](#) (detection\_path)
- def [visualize\\_matches](#) (matches\_path)
- def [visualize\\_track](#) (track\_path, options)
- def [windowclick](#) (event, visualizer)
- def [windowhover](#) (event, visualizer, playvid=False)
- def [windowhover\\_filename\\_only](#) (event, visualizer)

## Variables

- string `fdir` = `"/Users/2r6/Projects/briar/briar-api/media/test_probe/clinton3.matches"`
- list `files` = `[os.path.join(fdir, f) for f in os.listdir(fdir)]`

## 8.53 media\_converters.py File Reference

### Namespaces

- `briar.media_converters`

*Contained in this are functions for converting numpy arrays into various protobuf objects and back again since numpy arrays cannot be sent directly over gRPC.*

### Functions

- def `attribute_find` (key, List[briar\_pb2.Attribute] attributes)
- def `attribute_proto2val` (briar\_pb2.Attribute attribute)
- def `attribute_retrieve` (briar\_pb2.Attribute attribute, return\_type=False)
- def `attribute_val2proto` (key, val, briar\_pb2.BriarDataType override\_type=None)
- def `check_if_delete_request` (srv\_pb2.DatabaseInsertRequest request)
- def `check_if_delete_request_due_to_error` (srv\_pb2.DatabaseInsertRequest request)
- def `get_entry_id_list` (srv\_pb2.DatabaseInsertRequest request)
- def `image_cv2proto` (im, compression='uint8', quality=99, flip\_channels=True)  
*Convert a cv2 numpy array to a protobuf format.*
- def `image_file2proto` (imfile, path\_map={})
- def `image_np2proto` (im, compression='uint8', quality=99, flip\_channels=True)  
*Convert a numpy array to a protobuf format.*
- def `image_proto2cv` (pb\_data, flip\_channels=False)  
*Convert a protobuf BriarMedia image to a cv2 numpy array.*
- def `image_proto2np` (pb\_data, flip\_channels=True)  
*Convert a protobuf image to a numpy array.*
- def `matrix_np2proto` (mat)  
*Convert a numpy matrix into a BriarMatrix.*
- def `matrix_proto2np` (protomat)  
*Convert a protobuf matrix into a numpy matrix.*
- def `modality_proto2string` (modality)
- def `modality_string2proto` (modality)
- def `pathmap_path2remotepath` (path, path\_map, exclude\_cases\_containing\_folder=['mugshots'])
- def `pathmap_str2dict` (path\_map)
- def `subjectID_int2str` (subjectid)
- def `subjectID_str2int` (subjectid)
- def `subjectList_list2string` (subject\_list\_str, chomp=True)
- def `subjectList_string2list` (subject\_list)
- def `tracklet_list2proto` (track\_list)
- def `vector_np2proto` (vec)  
*Convert a 1 dimensional np array into a BriarVector.*
- def `vector_proto2np` (protovec)  
*Convert a protobuf vector into a numpy array.*
- def `video_file2proto` (vidfile, start, end, path\_map={})

## Variables

- dictionary `attribute_type_name_map` = {'int': 'ivalue', 'float': 'fvalue', 'string': 'text'}
- dictionary `modalityDict`
- dictionary `reverseModalityDict` = {modalityDict[k]: k for k in modalityDict}

## 8.54 readme-cli.md File Reference

## 8.55 sigset/parse.py File Reference

## Namespaces

- `briar.sigset.parse`

## Functions

- def `create_test_sigset` (str sigset\_probe\_file, str sigset\_gallery\_file, str base\_dir, output\_dir)
- def `expandTree` (root, level=0, spaces=3)
- def `parseBriarSigset` (filename)

## Variables

- `args` = sys.argv

# Index

- `__ONE_DAY`
  - `briar`, [14](#)
- `__aiter__`
  - `BriarVideoIterator`, [125](#)
  - `ThreadedVideoIterator`, [227](#)
  - `VideoIterator`, [232](#)
- `__anext__`
  - `BriarVideoIterator`, [125](#)
  - `ThreadedVideoIterator`, [227](#)
  - `VideoIterator`, [233](#)
- `__init__`
  - `BriarClient`, [103](#)
  - `BriarMedia`, [116](#)
  - `BriarProgress`, [119](#)
  - `BriarTest`, [122](#)
  - `BriarTestResult`, [123](#)
  - `BriarVideoIterator`, [124](#)
  - `FileVideoStream_cv2`, [134](#)
  - `FileVideoStream_imageio`, [137](#)
  - `GrpcDecoder`, [143](#)
  - `GrpcEncoder`, [145](#)
  - `ImageIterator`, [147](#)
  - `match_matrix_visualizer`, [151](#)
  - `MediaSetIterator`, [155](#)
  - `MediaStream`, [158](#)
  - `Rect`, [159](#)
  - `ThreadedVideoIterator`, [226](#)
  - `VideoIterator`, [232](#)
- `__init__.py`, [237](#)
- `__iter__`
  - `BriarVideoIterator`, [125](#)
  - `ImageIterator`, [148](#)
  - `MediaSetIterator`, [155](#)
  - `MediaStream`, [158](#)
  - `ThreadedVideoIterator`, [227](#)
  - `VideoIterator`, [233](#)
- `__len__`
  - `BriarVideoIterator`, [126](#)
  - `ImageIterator`, [148](#)
  - `MediaSetIterator`, [155](#)
  - `ThreadedVideoIterator`, [227](#)
  - `VideoIterator`, [233](#)
- `__main__.py`, [240](#)
- `__next__`
  - `BriarVideoIterator`, [126](#)
  - `ImageIterator`, [148](#)
  - `MediaSetIterator`, [156](#)
  - `ThreadedVideoIterator`, [228](#)
  - `VideoIterator`, [234](#)

- `__version__`
  - `briar`, [14](#)
- `_client_identifier_singleton`
  - `briar.briar_client`, [20](#)
- `_initialize_worker`
  - `briar.briar_client`, [19](#)
- `_media_list`
  - `MediaStream`, [158](#)
- `_shutdown_worker`
  - `briar.briar_client`, [19](#)
- `_wait_forever`
  - `briar`, [12](#)
- `_worker_channel_singleton`
  - `briar.briar_client`, [20](#)
- `_worker_port_singleton`
  - `briar.briar_client`, [20](#)
- `_worker_process_position_singleton`
  - `briar.briar_client`, [20](#)
- `_worker_stub_singleton`
  - `briar.briar_client`, [20](#)
- `_worker_thread_position_singleton`
  - `briar.briar_client`, [20](#)
- `addConnectionOptions`
  - `briar.cli.connection`, [23](#)
- `addDatabaseComputeScore_options2proto`
  - `briar.cli.database.compute_scores`, [26](#)
- `addDatabaseComputeScoreOptions`
  - `briar.cli.database.compute_scores`, [26](#)
- `addDetectorOptions`
  - `briar.cli.detect`, [37](#)
- `addEnhanceOptions`
  - `briar.cli.enhance`, [40](#)
- `addEnrollOptions`
  - `briar.cli.enroll`, [42](#)
- `addExtractOptions`
  - `briar.cli.extract`, [44](#)
- `addMediaOptions`
  - `briar.cli.media`, [47](#)
- `addSearchOptions`
  - `briar.cli.search`, [49](#)
- `addTrackingOptions`
  - `briar.cli.detect`, [37](#)
- `addVerifyOptions`
  - `briar.cli.verify`, [60](#)
- `aenumerate`
  - `briar.media`, [77](#)
- `annotations`
  - `match_matrix_visualizer`, [152](#)
- `args`

- briar.sigset.parse, 98
- args\_string
  - briar.evaluation.full\_evaluation, 66
- ATTRIB\_IGNORE
  - briar.grpc\_json, 76
- attribute\_find
  - briar.media\_converters, 86
- attribute\_proto2val
  - briar.media\_converters, 87
- attribute\_retrieve
  - briar.media\_converters, 87
- attribute\_type\_name\_map
  - briar.media\_converters, 96
- attribute\_val2proto
  - briar.media\_converters, 87
- ax
  - match\_matrix\_visualizer, 152
- backend
  - FileVideoStream\_imageio, 140
- blended\_gallery\_1\_filename
  - briar.evaluation.full\_evaluation, 67
- blended\_gallery\_2\_filename
  - briar.evaluation.full\_evaluation, 67
- BLUE
  - Fore, 141
- briar, 11
  - \_ONE\_DAY, 14
  - \_\_version\_\_, 14
  - \_wait\_forever, 12
  - CLIServe, 12
  - DEFAULT\_MAX\_MESSAGE\_SIZE, 14
  - DEFAULT\_PORT, 14
  - DEFAULT\_SERVE\_PORT, 15
  - dyn\_import, 12
  - get\_process\_number, 13
  - get\_thread\_number, 13
  - multiproc\_serve, 13
  - parse\_ports, 13
  - PLATFORM, 15
  - serve, 14
- briar.\_\_main\_\_, 15
- briar.briar\_cli, 15
  - briar\_command\_line, 16
  - briar\_database\_command\_line, 16
  - briar\_test\_command\_line, 16
  - COMMANDS, 17
  - DATABASE\_COMMANDS, 17
  - DETECTION\_FILE\_EXT, 18
  - FACE\_COUNT, 18
  - incomplete, 17
  - MATCHES\_FILE\_EXT, 18
  - TEMPLATE\_FILE\_EXT, 18
- briar.briar\_client, 18
  - \_client\_identifier\_singleton, 20
  - \_initialize\_worker, 19
  - \_shutdown\_worker, 19
  - \_worker\_channel\_singleton, 20
  - \_worker\_port\_singleton, 20
  - \_worker\_proccess\_position\_singleton, 20
  - \_worker\_stub\_singleton, 20
  - \_worker\_thread\_position\_singleton, 20
- briar.briar\_media, 21
  - briar\_media\_from\_pb2, 21
  - briar\_media\_to\_pb2, 21
  - load\_media\_from\_folder, 21
  - load\_media\_from\_image, 21
  - load\_media\_from\_numpy, 22
- briar.cli, 22
  - briar.cli.connection, 23
    - addConnectionOptions, 23
    - DEFAULT\_MAX\_ASYNC, 23
    - DEFAULT\_MAX\_MESSAGE\_SIZE, 23
  - briar.cli.database, 24
    - briar.cli.database.checkpoint, 24
      - database\_checkpoint, 24
    - briar.cli.database.checkpoint\_subject, 24
      - database\_checkpoint\_subject, 25
      - parseDatabaseCheckpointSubjectOptions, 25
  - briar.cli.database.common, 25
    - db\_no\_exist, 25
  - briar.cli.database.compute\_scores, 26
    - addDatabaseComputeScore\_options2proto, 26
    - addDatabaseComputeScoreOptions, 26
    - database\_compute\_verify, 26
    - parseDatabaseComputeScoreOptions, 27
  - briar.cli.database.compute\_search, 27
    - database\_compute\_search, 27
  - briar.cli.database.create, 28
    - database\_create, 28
  - briar.cli.database.delete, 28
    - database\_delete, 28
    - parseDatabaseDeleteOptions, 28
  - briar.cli.database.finalize, 29
    - database\_finalize, 29
    - finalizeParseOptions, 29
  - briar.cli.database.info, 30
    - database\_info, 30
    - parseDatabaseInfoOptions, 30
  - briar.cli.database.list, 30
    - database\_list, 31
    - parseDatabaseListOptions, 31
  - briar.cli.database.list\_entries, 31
    - database\_list\_entries, 31
    - parseDatabaseListEntriesOptions, 31
  - briar.cli.database.load, 32
    - database\_load, 32
  - briar.cli.database.merge, 32
    - database\_merge, 32
    - parseDatabaseMergeOptions, 33
  - briar.cli.database.move\_entry, 33
    - database\_move\_entry, 33
    - parseDatabaseMoveEntryOptions, 33
  - briar.cli.database.refresh, 34
    - database\_checkpoint, 34
    - database\_refresh, 34
    - parseDatabaseRefreshOptions, 34



- briar.cli.database.rename, 35
  - database\_load, 35
  - database\_rename, 35
  - parseDatabaseRenameOptions, 35
- briar.cli.database.retrieve, 36
  - database\_retrieve, 36
  - parseDatabaseRetrieveOptions, 36
- briar.cli.detect, 37
  - addDetectorOptions, 37
  - addTrackingOptions, 37
  - detect, 38
  - detect\_options2proto, 38
  - DETECTION\_FILE\_EXT, 40
  - detectParseOptions, 38
  - detectRequestConstructor, 38
  - get\_detection\_path, 39
  - isFinalReply, 39
  - save\_detections, 39
  - tracking\_options2proto, 39
- briar.cli.enhance, 40
  - addEnhanceOptions, 40
  - enhance, 40
  - ENHANCE\_FILE\_EXT, 42
  - enhance\_options2proto, 41
  - enhanceParseOptions, 41
  - enhanceRequestConstructor, 41
  - save\_Enhancement, 41
- briar.cli.enroll, 42
  - addEnrollOptions, 42
  - enroll, 43
  - enroll\_options2proto, 43
  - enrollParseOptions, 43
  - enrollRequestConstructor, 43
- briar.cli.extract, 44
  - addExtractOptions, 44
  - extract, 45
  - extract\_options2proto, 45
  - extractParseOptions, 45
  - extractRequestConstructor, 45
  - save\_extractions, 46
  - TEMPLATE\_FILE\_EXT, 46
- briar.cli.finalize, 46
  - database\_finalize, 46
  - finalizeParseOptions, 47
- briar.cli.media, 47
  - addMediaOptions, 47
  - collect\_files, 48
  - DEFAULT\_MAX\_SIZE, 48
  - hasExtension, 48
- briar.cli.search, 49
  - addSearchOptions, 49
  - MATCHES\_FILE\_EXT, 50
  - search, 49
  - search\_options2proto, 50
  - searchParseOptions, 50
  - searchRequestConstructor, 50
- briar.cli.sigset, 51
  - checkpoint\_subject\_threaded, 51
  - df\_batch\_consumer, 51
  - df\_batch\_generator, 51
  - df\_batch\_producer, 52
  - df\_row\_generator, 52
  - enroll\_call\_threaded, 53
  - inner\_pool\_mapper, 53
  - my\_pool, 55
  - parseSigsetEnrollOptions, 54
  - parseSigsetStatsOptions, 54
  - proc\_number, 55
  - progress\_consumer, 54
  - service\_address\_number, 55
  - sigset\_enroll, 54
  - sigset\_stats, 55
- briar.cli.status, 56
  - get\_service\_configuration, 56
  - print\_service\_configuration, 56
  - status, 56
  - statusParseOptions, 56
- briar.cli.test, 57
  - detection\_output\_tests, 57
  - extraction\_output\_tests, 57
  - use\_colorama, 58
- briar.cli.track, 58
  - get\_tracklet\_path, 58
  - save\_tracklets, 58
  - track, 59
  - TRACKLET\_FILE\_EXT, 59
  - trackRequestConstructor, 59
- briar.cli.verify, 60
  - addVerifyOptions, 60
  - save\_verifications, 60
  - VERIFICATION\_FILE\_EXT, 61
  - verify, 60
  - verify\_options2proto, 61
  - verifyParseOptions, 61
- briar.cli.viz, 62
  - viz, 62
  - vizParseOptions, 62
- briar.evaluation, 62
  - runStages, 63
- briar.evaluation.full\_evaluation, 63
  - args\_string, 66
  - blended\_gallery\_1\_filename, 67
  - blended\_gallery\_2\_filename, 67
  - category, 67
  - compute\_search, 65
  - compute\_verify, 65
  - database\_blended\_gallery\_1\_name, 67
  - database\_blended\_gallery\_2\_name, 67
  - database\_gallery\_1\_name, 67
  - database\_gallery\_2\_name, 67
  - database\_multi\_probe\_name, 67
  - database\_probe\_name, 68
  - DATABASE\_SUFFIX\_FLAG, 68
  - DATASET\_DIR, 68
  - enroll\_args, 68
  - EVAL\_PHASE, 68

- EVALUATION\_DIR, 68
- EVALUATION\_MULTISUBJECT\_DIR, 68
- gallery1\_blended\_sigset, 68
- gallery1\_sigset, 69
- gallery2\_blended\_sigset, 69
- gallery2\_sigset, 69
- gallery\_1\_blended\_sigset\_path, 69
- gallery\_1\_filename, 69
- gallery\_1\_sigset\_path, 69
- gallery\_2\_blended\_sigset\_path, 69
- gallery\_2\_filename, 69
- gallery\_2\_sigset\_path, 70
- generate\_report, 70
- get\_info, 65
- get\_multi\_info, 65
- media\_args, 70
- merge\_dbs, 66
- module, 70
- multisubject\_probe\_filename, 70
- number\_of\_partitions, 70
- OUTPUT\_DIR, 70
- port\_list, 70
- probe\_filename, 71
- probe\_multisubject\_sigset, 71
- probe\_multisubject\_sigset\_path, 71
- probe\_sigset, 71
- probe\_sigset\_path, 71
- requires\_database\_merge, 71
- run\_multisubject\_evaluation, 71
- run\_on\_multi, 66
- RUN\_STAGES, 71
- runall, 66
- setUpClass\_main, 66
- setUpModule, 66
- sortTestMethodsUsing, 72
- stages\_temp, 72
- USE\_SINGLE\_SUBJECT, 72
- USES\_FRONTEND\_MERGING, 72
- briar.evaluation.stage1\_probe\_enroll, 72
  - generate\_report, 72
  - main, 72
- briar.evaluation.stage2, 73
  - generate\_report, 73
  - main, 73
- briar.evaluation.stage3\_result\_scoring, 73
  - generate\_report, 73
  - main, 73
- briar.grpc\_json, 74
  - ATTRIB\_IGNORE, 76
  - dict\_to\_proto\_obj, 74
  - load, 75
  - proto\_obj\_to\_dict, 75
  - save, 75
- briar.media, 76
  - aenumerate, 77
  - decodeMedia, 77
  - enroll\_frames\_iter, 77
  - enroll\_frames\_iter\_async, 79
  - file\_iter, 80
  - frame\_iter, 81
  - ImageGenerator, 81
  - isFinalFrame, 81
  - single\_frame\_generate, 82
- briar.media.VideoStream, 82
- briar.media.visualize, 83
  - decode\_track, 83
  - fdir, 85
  - files, 85
  - get\_frame, 83
  - playVideo, 83
  - update\_annot, 84
  - update\_annot\_filename\_only, 84
  - visualize\_detection, 84
  - visualize\_matches, 84
  - visualize\_track, 84
  - windowclick, 84
  - windowhover, 85
  - windowhover\_filename\_only, 85
- briar.media\_converters, 85
  - attribute\_find, 86
  - attribute\_proto2val, 87
  - attribute\_retrieve, 87
  - attribute\_type\_name\_map, 96
  - attribute\_val2proto, 87
  - check\_if\_delete\_request, 88
  - check\_if\_delete\_request\_due\_to\_error, 88
  - get\_entry\_id\_list, 88
  - image\_cv2proto, 89
  - image\_file2proto, 89
  - image\_np2proto, 89
  - image\_proto2cv, 91
  - image\_proto2np, 91
  - matrix\_np2proto, 91
  - matrix\_proto2np, 92
  - modality\_proto2string, 92
  - modality\_string2proto, 92
  - modalityDict, 96
  - pathmap\_path2remotepath, 93
  - pathmap\_str2dict, 93
  - reverseModalityDict, 97
  - subjectID\_int2str, 93
  - subjectID\_str2int, 94
  - subjectList\_list2string, 94
  - subjectList\_string2list, 94
  - tracklet\_list2proto, 95
  - vector\_np2proto, 95
  - vector\_proto2np, 95
  - video\_file2proto, 96
- briar.sigset, 97
- briar.sigset.parse, 97
  - args, 98
  - create\_test\_sigset, 97
  - expandTree, 97
  - parseBriarSigset, 98
- briar.timing, 98
  - DURATION\_FILE\_EXT, 100

- end\_duration, 99
- generate\_progress, 99
- loadDurationsFolder, 99
- parseDurations, 99
- print\_duration, 99
- print\_durations, 99
- save\_durations, 100
- start\_duration, 100
- timeElapsed, 100
- timestamp, 100
- briar\_cli.py, 240
- briar\_client.py, 240
- briar\_command\_line
  - briar.briar\_cli, 16
- briar\_database\_command\_line
  - briar.briar\_cli, 16
- briar\_media.py, 241
- briar\_media\_from\_pb2
  - briar.briar\_media, 21
- briar\_media\_to\_pb2
  - briar.briar\_media, 21
- briar\_test\_command\_line
  - briar.briar\_cli, 16
- BriarClient, 101
  - \_\_init\_\_, 103
  - channel, 115
  - database\_create, 104
  - database\_insert, 104
  - database\_list\_templates, 105
  - database\_refresh, 105
  - database\_remove\_templates, 105
  - database\_retrieve, 105
  - DEFAULT\_PORT, 115
  - detect, 106
  - enhance, 106
  - enroll, 107
  - enroll\_frames\_iter, 107
  - enroll\_frames\_iter\_async, 107
  - extract, 108
  - finalize, 108
  - get\_database\_names, 108
  - get\_service\_configuration, 109
  - get\_status, 109
  - iter\_over\_async, 109
  - load\_database, 110
  - options, 115
  - port, 116
  - print\_verbose, 110
  - retrieve\_req\_iter, 110
  - search, 111
  - stub, 116
  - sync\_enroll\_frames\_iter, 111
  - track, 112
  - verify, 112
  - verify\_file\_iter, 113
  - verify\_files, 114
- BriarMedia, 116
  - \_\_init\_\_, 116
- channels, 117
- DATA\_TYPES, 117
- datetime, 117
- description, 117
- fps, 117
- height, 117
- IMAGE\_FORMATS, 118
- len, 118
- metadata, 118
- source, 118
- VIDEO\_FORMATS, 118
- width, 118
- BriarProgress, 119
  - \_\_init\_\_, 119
  - close, 119
  - desc, 120
  - enabled, 120
  - leave, 120
  - name, 120
  - pbar, 121
  - position, 121
  - prevstep, 121
  - refresh, 120
  - tqdm, 121
  - update, 120
- BriarTest, 121
  - \_\_init\_\_, 122
  - description, 122
  - run, 122
  - test, 122
- BriarTestResult, 123
  - \_\_init\_\_, 123
  - level, 123
  - name, 123
  - passed, 123
  - reason, 123
- BriarVideoIterator, 124
  - \_\_aiter\_\_, 125
  - \_\_anext\_\_, 125
  - \_\_init\_\_, 124
  - \_\_iter\_\_, 125
  - \_\_len\_\_, 126
  - \_\_next\_\_, 126
- cap
  - ThreadedVideoIterator, 228
  - VideoIterator, 234
- category
  - briar.evaluation.full\_evaluation, 67
- channel
  - BriarClient, 115
- channels
  - BriarMedia, 117
- check\_if\_delete\_request
  - briar.media\_converters, 88
- check\_if\_delete\_request\_due\_to\_error
  - briar.media\_converters, 88
- checkpoint\_subject\_threaded
  - briar.cli.sigset, 51

- cli/\_\_\_init\_\_\_py, 238
- cli/connection.py, 241
- cli/database/\_\_\_init\_\_\_py, 238
- cli/database/checkpoint.py, 242
- cli/database/checkpoint\_subject.py, 242
- cli/database/common.py, 242
- cli/database/compute\_scores.py, 242
- cli/database/compute\_search.py, 243
- cli/database/create.py, 243
- cli/database/delete.py, 243
- cli/database/finalize.py, 244
- cli/database/info.py, 244
- cli/database/list.py, 244
- cli/database/list\_entries.py, 245
- cli/database/load.py, 245
- cli/database/merge.py, 245
- cli/database/move\_entry.py, 246
- cli/database/refresh.py, 246
- cli/database/rename.py, 246
- cli/database/retrieve.py, 246
- cli/detect.py, 247
- cli/enhance.py, 247
- cli/enroll.py, 248
- cli/extract.py, 248
- cli/finalize.py, 244
- cli/media.py, 249
- cli/search.py, 249
- cli/sigset.py, 249
- cli/status.py, 250
- cli/test.py, 250
- cli/track.py, 251
- cli/verify.py, 251
- cli/viz.py, 252
- CLIServe
  - briar, 12
- close
  - BriarProgress, 119
- collect\_files
  - briar.cli.media, 48
- COMMANDS
  - briar.briar\_cli, 17
- compute\_search
  - briar.evaluation.full\_evaluation, 65
- compute\_verify
  - briar.evaluation.full\_evaluation, 65
- config\_reply
  - Test000InitialConfig, 163
- create\_test\_sigset
  - briar.sigset.parse, 97
- DATA\_TYPES
  - BriarMedia, 117
- database\_blended\_gallery\_1\_name
  - briar.evaluation.full\_evaluation, 67
- database\_blended\_gallery\_2\_name
  - briar.evaluation.full\_evaluation, 67
- database\_checkpoint
  - briar.cli.database.checkpoint, 24
  - briar.cli.database.refresh, 34
- database\_checkpoint\_subject
  - briar.cli.database.checkpoint\_subject, 25
- DATABASE\_COMMANDS
  - briar.briar\_cli, 17
- database\_compute\_search
  - briar.cli.database.compute\_search, 27
- database\_compute\_verify
  - briar.cli.database.compute\_scores, 26
- database\_create
  - briar.cli.database.create, 28
  - BriarClient, 104
- database\_delete
  - briar.cli.database.delete, 28
- database\_finalize
  - briar.cli.database.finalize, 29
  - briar.cli.finalize, 46
- database\_gallery\_1\_name
  - briar.evaluation.full\_evaluation, 67
- database\_gallery\_2\_name
  - briar.evaluation.full\_evaluation, 67
- database\_info
  - briar.cli.database.info, 30
- database\_insert
  - BriarClient, 104
- database\_list
  - briar.cli.database.list, 31
- database\_list\_entries
  - briar.cli.database.list\_entries, 31
- database\_list\_templates
  - BriarClient, 105
- database\_load
  - briar.cli.database.load, 32
  - briar.cli.database.rename, 35
- database\_merge
  - briar.cli.database.merge, 32
- database\_move\_entry
  - briar.cli.database.move\_entry, 33
- database\_multi\_probe\_name
  - briar.evaluation.full\_evaluation, 67
- database\_probe\_name
  - briar.evaluation.full\_evaluation, 68
- database\_refresh
  - briar.cli.database.refresh, 34
  - BriarClient, 105
- database\_remove\_templates
  - BriarClient, 105
- database\_rename
  - briar.cli.database.rename, 35
- database\_retrieve
  - briar.cli.database.retrieve, 36
  - BriarClient, 105
- DATABASE\_SUFFIX\_FLAG
  - briar.evaluation.full\_evaluation, 68
- DatabaseTest, 127
  - test, 127
- DATASET\_DIR
  - briar.evaluation.full\_evaluation, 68
- datetime

- BriarMedia, 117
- db\_no\_exist
  - briar.cli.database.common, 25
- debug\_empty
  - ImageIterator, 148
  - ThreadedVideoIterator, 228
  - VideoIterator, 234
- decode\_track
  - briar.media.visualize, 83
- decodeMedia
  - briar.media, 77
- default
  - GrpcDecoder, 143
  - GrpcEncoder, 145
- DEFAULT\_MAX\_ASYNC
  - briar.cli.connection, 23
- DEFAULT\_MAX\_MESSAGE\_SIZE
  - briar, 14
  - briar.cli.connection, 23
- DEFAULT\_MAX\_SIZE
  - briar.cli.media, 48
- DEFAULT\_PORT
  - briar, 14
  - BriarClient, 115
- DEFAULT\_SERVE\_PORT
  - briar, 15
- desc
  - BriarProgress, 120
- description
  - BriarMedia, 117
  - BriarTest, 122
  - DetectTest, 129
  - ExtractTest, 132
- detect
  - briar.cli.detect, 38
  - BriarClient, 106
- detect\_options2proto
  - briar.cli.detect, 38
- DETECTION\_FILE\_EXT
  - briar.briar\_cli, 18
  - briar.cli.detect, 40
- detection\_file\_path
  - DetectTest, 129
  - ExtractTest, 133
- detection\_output\_tests
  - briar.cli.test, 57
- detectParseOptions
  - briar.cli.detect, 38
- detectRequestConstructor
  - briar.cli.detect, 38
- DetectTest, 128
  - description, 129
  - detection\_file\_path, 129
  - output\_path, 130
  - test\_1\_detection\_image, 129
  - test\_2\_detection\_image\_output, 129
  - test\_3\_detection\_image\_withreturn, 129
  - test\_4\_detection\_image\_output\_withreturn, 129
  - testim\_path, 130
- df\_batch\_consumer
  - briar.cli.sigset, 51
- df\_batch\_generator
  - briar.cli.sigset, 51
- df\_batch\_producer
  - briar.cli.sigset, 52
- df\_row\_generator
  - briar.cli.sigset, 52
- dict\_to\_proto\_obj
  - briar.grpc\_json, 74
- DURATION\_FILE\_EXT
  - briar.timing, 100
- dyn\_import
  - briar, 12
- enabled
  - BriarProgress, 120
- end\_duration
  - briar.timing, 99
- enhance
  - briar.cli.enhance, 40
  - BriarClient, 106
- ENHANCE\_FILE\_EXT
  - briar.cli.enhance, 42
- enhance\_options2proto
  - briar.cli.enhance, 41
- enhanceParseOptions
  - briar.cli.enhance, 41
- enhanceRequestConstructor
  - briar.cli.enhance, 41
- enroll
  - briar.cli.enroll, 43
  - BriarClient, 107
- enroll\_args
  - briar.evaluation.full\_evaluation, 68
- enroll\_call\_threaded
  - briar.cli.sigset, 53
- enroll\_frames\_iter
  - briar.media, 77
  - BriarClient, 107
- enroll\_frames\_iter\_async
  - briar.media, 79
  - BriarClient, 107
- enroll\_options2proto
  - briar.cli.enroll, 43
- enrollParseOptions
  - briar.cli.enroll, 43
- enrollRequestConstructor
  - briar.cli.enroll, 43
- EnrollTest, 130
  - test, 131
- EVAL\_PHASE
  - briar.evaluation.full\_evaluation, 68
- evaluation/\_\_init\_\_.py, 238
- evaluation/full\_evaluation.py, 252
- evaluation/stage1\_probe\_enroll.py, 254
- evaluation/stage2.1\_gallery1\_simple\_enroll.py, 254
- evaluation/stage2.2\_gallery2\_simple\_enroll.py, 255

- evaluation/stage2.3\_gallery1\_blended\_enroll.py, 255
- evaluation/stage2.4\_gallery2\_blended\_enroll.py, 255
- evaluation/stage3\_result\_scoring.py, 255
- EVALUATION\_DIR
  - briar.evaluation.full\_evaluation, 68
- EVALUATION\_MULTISUBJECT\_DIR
  - briar.evaluation.full\_evaluation, 68
- expandTree
  - briar.sigset.parse, 97
- extract
  - briar.cli.extract, 45
  - BriarClient, 108
- extract\_options2proto
  - briar.cli.extract, 45
- extraction\_output\_tests
  - briar.cli.test, 57
- extractParseOptions
  - briar.cli.extract, 45
- extractRequestConstructor
  - briar.cli.extract, 45
- ExtractTest, 131
  - description, 132
  - detection\_file\_path, 133
  - output\_path, 133
  - template\_file\_path, 133
  - test\_1\_extraction\_image, 132
  - test\_2\_extraction\_image\_output, 132
  - testim\_path, 133
- FACE\_COUNT
  - briar.briar\_cli, 18
- fdir
  - briar.media.visualize, 85
- fig
  - match\_matrix\_visualizer, 152
- figures
  - match\_matrix\_visualizer, 152
- file\_iter
  - briar.media, 80
- filepath
  - ImageIterator, 149
  - ThreadedVideoIterator, 229
  - VideoIterator, 234
- filepaths
  - MediaSetIterator, 156
- files
  - briar.media.visualize, 85
- FileVideoStream\_cv2, 133
  - \_\_init\_\_, 134
  - get\_fps, 134
  - get\_height, 134
  - get\_length, 134
  - get\_position, 134
  - get\_width, 135
  - is\_open, 135
  - more, 135
  - options, 136
  - Q, 136
  - read, 135
  - running, 135
  - scrub\_to, 135
  - start, 135
  - stop, 136
  - stopped, 136
  - stream, 136
  - thread, 136
  - transform, 137
  - update, 136
- FileVideoStream\_imageio, 137
  - \_\_init\_\_, 137
  - backend, 140
  - fps, 140
  - get\_fps, 138
  - get\_height, 138
  - get\_length, 138
  - get\_position, 138
  - get\_width, 138
  - is\_open, 138
  - more, 139
  - options, 140
  - Q, 140
  - read, 139
  - running, 139
  - scrub\_to, 139
  - start, 139
  - stop, 139
  - stopped, 140
  - stream, 140
  - thread, 140
  - transform, 141
  - update, 139
- finalize
  - BriarClient, 108
- finalizeParseOptions
  - briar.cli.database.finalize, 29
  - briar.cli.finalize, 47
- Fore, 141
  - BLUE, 141
  - GREEN, 141
  - RED, 141
  - RESET, 141
  - YELLOW, 142
- fps
  - BriarMedia, 117
  - FileVideoStream\_imageio, 140
  - ImageIterator, 149
  - ThreadedVideoIterator, 229
  - VideoIterator, 235
- frame
  - ImageIterator, 149
- frame\_count
  - ImageIterator, 149
  - ThreadedVideoIterator, 229
  - VideoIterator, 235
- frame\_height
  - ImageIterator, 149
  - ThreadedVideoIterator, 229

- Videolterator, [235](#)
- frame\_iter
  - briar.media, [81](#)
- frame\_width
  - Imageliterator, [149](#)
  - ThreadedVideolterator, [229](#)
  - Videolterator, [235](#)
- gallery1\_blended\_sigset
  - briar.evaluation.full\_evaluation, [68](#)
- gallery1\_sigset
  - briar.evaluation.full\_evaluation, [69](#)
- gallery2\_blended\_sigset
  - briar.evaluation.full\_evaluation, [69](#)
- gallery2\_sigset
  - briar.evaluation.full\_evaluation, [69](#)
- gallery\_1\_blended\_sigset\_path
  - briar.evaluation.full\_evaluation, [69](#)
- gallery\_1\_filename
  - briar.evaluation.full\_evaluation, [69](#)
- gallery\_1\_sigset\_path
  - briar.evaluation.full\_evaluation, [69](#)
- gallery\_2\_blended\_sigset\_path
  - briar.evaluation.full\_evaluation, [69](#)
- gallery\_2\_filename
  - briar.evaluation.full\_evaluation, [69](#)
- gallery\_2\_sigset\_path
  - briar.evaluation.full\_evaluation, [70](#)
- gallerydb\_name
  - match\_matrix\_visualizer, [152](#)
- generate\_progress
  - briar.timing, [99](#)
- generate\_report
  - briar.evaluation.full\_evaluation, [70](#)
  - briar.evaluation.stage1\_probe\_enroll, [72](#)
  - briar.evaluation.stage2, [73](#)
  - briar.evaluation.stage3\_result\_scoring, [73](#)
- get\_database\_names
  - BriarClient, [108](#)
- get\_detection\_path
  - briar.cli.detect, [39](#)
- get\_entry\_id\_list
  - briar.media\_converters, [88](#)
- get\_fps
  - FileVideoStream\_cv2, [134](#)
  - FileVideoStream\_imageio, [138](#)
- get\_frame
  - briar.media.visualize, [83](#)
- get\_height
  - FileVideoStream\_cv2, [134](#)
  - FileVideoStream\_imageio, [138](#)
- get\_info
  - briar.evaluation.full\_evaluation, [65](#)
- get\_length
  - FileVideoStream\_cv2, [134](#)
  - FileVideoStream\_imageio, [138](#)
- get\_multi\_info
  - briar.evaluation.full\_evaluation, [65](#)
- get\_position
  - FileVideoStream\_cv2, [134](#)
  - FileVideoStream\_imageio, [138](#)
- get\_process\_number
  - briar, [13](#)
- get\_service\_configuration
  - briar.cli.status, [56](#)
  - BriarClient, [109](#)
- get\_status
  - BriarClient, [109](#)
- get\_thread\_number
  - briar, [13](#)
- get\_tracklet\_path
  - briar.cli.track, [58](#)
- get\_width
  - FileVideoStream\_cv2, [135](#)
  - FileVideoStream\_imageio, [138](#)
- GREEN
  - Fore, [141](#)
- grpc\_json.py, [255](#)
- GrpcDecoder, [142](#)
  - \_\_init\_\_, [143](#)
  - default, [143](#)
  - options, [144](#)
- GrpcEncoder, [144](#)
  - \_\_init\_\_, [145](#)
  - default, [145](#)
  - options, [146](#)
- gt
  - match\_matrix\_visualizer, [152](#)
- hasExtension
  - briar.cli.media, [48](#)
- height
  - BriarMedia, [117](#)
  - Rect, [159](#)
- i
  - Imageliterator, [149](#)
  - MediaSetIterator, [156](#)
  - ThreadedVideolterator, [229](#)
  - Videolterator, [235](#)
- image\_cv2proto
  - briar.media\_converters, [89](#)
- image\_file2proto
  - briar.media\_converters, [89](#)
- IMAGE\_FORMATS
  - BriarMedia, [118](#)
- image\_np2proto
  - briar.media\_converters, [89](#)
- image\_proto2cv
  - briar.media\_converters, [91](#)
- image\_proto2np
  - briar.media\_converters, [91](#)
- ImageGenerator
  - briar.media, [81](#)
- Imageliterator, [146](#)
  - \_\_init\_\_, [147](#)
  - \_\_iter\_\_, [148](#)
  - \_\_len\_\_, [148](#)

- `__next__`, 148
- `debug_empty`, 148
- `filepath`, 149
- `fps`, 149
- `frame`, 149
- `frame_count`, 149
- `frame_height`, 149
- `frame_width`, 149
- `i`, 149
- `isOpened`, 149
- `length`, 150
- `msec`, 150
- `pos`, 150
- `processed`, 150
- `start_frame`, 150
- `stop_frame`, 150
- `incomplete`
  - `briar.briar_cli`, 17
- `inner_pool_mapper`
  - `briar.cli.sigset`, 53
- `is_open`
  - `FileVideoStream_cv2`, 135
  - `FileVideoStream_imageio`, 138
- `isFinalFrame`
  - `briar.media`, 81
- `isFinalReply`
  - `briar.cli.detect`, 39
- `isOpened`
  - `ImageIterator`, 149
  - `MediaSetIterator`, 156
  - `ThreadedVideoIterator`, 229
  - `VideoIterator`, 235
- `isStarted`
  - `ThreadedVideoIterator`, 229
- `iter_over_async`
  - `BriarClient`, 109
- `leave`
  - `BriarProgress`, 120
- `len`
  - `BriarMedia`, 118
- `length`
  - `ImageIterator`, 150
  - `ThreadedVideoIterator`, 230
  - `VideoIterator`, 235
- `level`
  - `BriarTestResult`, 123
- `load`
  - `briar.grpc_json`, 75
- `load_database`
  - `BriarClient`, 110
- `load_media_from_folder`
  - `briar.briar_media`, 21
- `load_media_from_image`
  - `briar.briar_media`, 21
- `load_media_from_numpy`
  - `briar.briar_media`, 22
- `loadDurationsFolder`
  - `briar.timing`, 99
- `main`
  - `briar.evaluation.stage1_probe_enroll`, 72
  - `briar.evaluation.stage2`, 73
  - `briar.evaluation.stage3_result_scoring`, 73
- `mat`
  - `match_matrix_visualizer`, 152
- `match_matrix_visualizer`, 151
  - `__init__`, 151
  - `annotations`, 152
  - `ax`, 152
  - `fig`, 152
  - `figures`, 152
  - `gallerydb_name`, 152
  - `gt`, 152
  - `mat`, 152
  - `prevx`, 152
  - `prevy`, 153
  - `probedb_name`, 153
  - `searchReply`, 153
  - `showmat_interactive`, 151
  - `xlabs`, 153
  - `xsources`, 153
  - `ylabs`, 153
  - `ysources`, 153
- `MATCHES_FILE_EXT`
  - `briar.briar_cli`, 18
  - `briar.cli.search`, 50
- `matrix_np2proto`
  - `briar.media_converters`, 91
- `matrix_proto2np`
  - `briar.media_converters`, 92
- `media/__init__.py`, 238
- `media/VideoStream.py`, 256
- `media/visualize.py`, 256
- `media_args`
  - `briar.evaluation.full_evaluation`, 70
- `media_converters.py`, 257
- `media_set`
  - `MediaSetIterator`, 156
- `MediaSetIterator`, 154
  - `__init__`, 155
  - `__iter__`, 155
  - `__len__`, 155
  - `__next__`, 156
  - `filepaths`, 156
  - `i`, 156
  - `isOpened`, 156
  - `media_set`, 156
  - `processed`, 157
  - `start_frame`, 157
  - `start_frames`, 157
  - `stop_frame`, 157
  - `stop_frames`, 157
- `MediaStream`, 157
  - `__init__`, 158
  - `__iter__`, 158
  - `_media_list`, 158
- `merge_db_func`



- Test006GalleryDatabaseMerge, 173
- merge\_dbs
  - briar.evaluation.full\_evaluation, 66
- merged\_dbinfo
  - Test002ProbeDatabaseMerge, 166
  - Test004MultiProbeDatabaseMerge, 169
- metadata
  - BriarMedia, 118
- modality\_proto2string
  - briar.media\_converters, 92
- modality\_string2proto
  - briar.media\_converters, 92
- modalityDict
  - briar.media\_converters, 96
- module
  - briar.evaluation.full\_evaluation, 70
- more
  - FileVideoStream\_cv2, 135
  - FileVideoStream\_imageio, 139
- msec
  - Imageliterator, 150
  - ThreadedVideoliterator, 230
  - Videoliterator, 235
- multiproc\_serve
  - briar, 13
- multisubject\_probe\_filename
  - briar.evaluation.full\_evaluation, 70
- my\_pool
  - briar.cli.sigset, 55
- name
  - BriarProgress, 120
  - BriarTestResult, 123
- number\_of\_partitions
  - briar.evaluation.full\_evaluation, 70
- options
  - BriarClient, 115
  - FileVideoStream\_cv2, 136
  - FileVideoStream\_imageio, 140
  - GrpcDecoder, 144
  - GrpcEncoder, 146
  - ThreadedVideoliterator, 230
- OUTPUT\_DIR
  - briar.evaluation.full\_evaluation, 70
- output\_path
  - DetectTest, 130
  - ExtractTest, 133
- parse\_ports
  - briar, 13
- parseBriarSigset
  - briar.sigset.parse, 98
- parseDatabaseCheckpointSubjectOptions
  - briar.cli.database.checkpoint\_subject, 25
- parseDatabaseComputeScoreOptions
  - briar.cli.database.compute\_scores, 27
- parseDatabaseDeleteOptions
  - briar.cli.database.delete, 28
- parseDatabaseInfoOptions
  - briar.cli.database.info, 30
- parseDatabaseListEntriesOptions
  - briar.cli.database.list\_entries, 31
- parseDatabaseListOptions
  - briar.cli.database.list, 31
- parseDatabaseMergeOptions
  - briar.cli.database.merge, 33
- parseDatabaseMoveEntryOptions
  - briar.cli.database.move\_entry, 33
- parseDatabaseRefreshOptions
  - briar.cli.database.refresh, 34
- parseDatabaseRenameOptions
  - briar.cli.database.rename, 35
- parseDatabaseRetrieveOptions
  - briar.cli.database.retrieve, 36
- parseDurations
  - briar.timing, 99
- parseSigsetEnrollOptions
  - briar.cli.sigset, 54
- parseSigsetStatsOptions
  - briar.cli.sigset, 54
- passed
  - BriarTestResult, 123
- pathmap\_path2remotepath
  - briar.media\_converters, 93
- pathmap\_str2dict
  - briar.media\_converters, 93
- pbar
  - BriarProgress, 121
- PLATFORM
  - briar, 15
- playVideo
  - briar.media.visualize, 83
- port
  - BriarClient, 116
- port\_list
  - briar.evaluation.full\_evaluation, 70
- pos
  - Imageliterator, 150
  - ThreadedVideoliterator, 230
  - Videoliterator, 236
- position
  - BriarProgress, 121
- prevstep
  - BriarProgress, 121
- prevx
  - match\_matrix\_visualizer, 152
- prevy
  - match\_matrix\_visualizer, 153
- print\_duration
  - briar.timing, 99
- print\_durations
  - briar.timing, 99
- print\_service\_configuration
  - briar.cli.status, 56
- print\_verbose
  - BriarClient, 110

- probe\_filename
  - briar.evaluation.full\_evaluation, 71
- probe\_multisubject\_sigset
  - briar.evaluation.full\_evaluation, 71
- probe\_multisubject\_sigset\_path
  - briar.evaluation.full\_evaluation, 71
- probe\_sigset
  - briar.evaluation.full\_evaluation, 71
- probe\_sigset\_path
  - briar.evaluation.full\_evaluation, 71
- probedb\_name
  - match\_matrix\_visualizer, 153
- proc\_number
  - briar.cli.sigset, 55
- processed
  - ImageIterator, 150
  - MediaSetIterator, 157
  - ThreadedVideoIterator, 230
  - VideoIterator, 236
- progress\_consumer
  - briar.cli.sigset, 54
- proto\_obj\_to\_dict
  - briar.grpc\_json, 75
- Q
  - FileVideoStream\_cv2, 136
  - FileVideoStream\_imageio, 140
- read
  - FileVideoStream\_cv2, 135
  - FileVideoStream\_imageio, 139
- readme-cli.md, 258
- reason
  - BriarTestResult, 123
- Rect, 158
  - \_\_init\_\_, 159
  - height, 159
  - width, 159
  - x, 159
  - y, 160
- RED
  - Fore, 141
- refresh
  - BriarProgress, 120
- requires\_database\_merge
  - briar.evaluation.full\_evaluation, 71
- RESET
  - Fore, 141
- retrieve\_req\_iter
  - BriarClient, 110
- reverseModalityDict
  - briar.media\_converters, 97
- run
  - BriarTest, 122
- run\_multisubject\_evaluation
  - briar.evaluation.full\_evaluation, 71
- run\_on\_multi
  - briar.evaluation.full\_evaluation, 66
- RUN\_STAGES
  - briar.evaluation.full\_evaluation, 71
- runall
  - briar.evaluation.full\_evaluation, 66
- runGallery
  - Test005SigsetEnrollGalleries, 171
- running
  - FileVideoStream\_cv2, 135
  - FileVideoStream\_imageio, 139
- runStages
  - briar.evaluation, 63
- save
  - briar.grpc\_json, 75
- save\_detections
  - briar.cli.detect, 39
- save\_durations
  - briar.timing, 100
- save\_Enhancement
  - briar.cli.enhance, 41
- save\_extractions
  - briar.cli.extract, 46
- save\_tracklets
  - briar.cli.track, 58
- save\_verifications
  - briar.cli.verify, 60
- score\_file\_check
  - Test024SigsetVerifyOutputFormatting, 204
- scrub\_to
  - FileVideoStream\_cv2, 135
  - FileVideoStream\_imageio, 139
- search
  - briar.cli.search, 49
  - BriarClient, 111
- search\_file\_check
  - Test023SigsetSearchOutputFormatting, 199
- search\_options2proto
  - briar.cli.search, 50
- searchParseOptions
  - briar.cli.search, 50
- searchReply
  - match\_matrix\_visualizer, 153
- searchRequestConstructor
  - briar.cli.search, 50
- serve
  - briar, 14
- service\_address\_number
  - briar.cli.sigset, 55
- setUpClass
  - Test000InitialConfig, 161
  - Test001SigsetEnrollProbe, 164
  - Test002ProbeDatabaseMerge, 165
  - Test003SigsetEnrollMultiProbe, 167
  - Test004MultiProbeDatabaseMerge, 169
  - Test005SigsetEnrollGalleries, 171
  - Test006GalleryDatabaseMerge, 174
  - Test007SigsetScoreG1, 176
  - Test008SigsetScoreGaitG1, 177
  - Test009SigsetScoreFaceG1, 178
  - Test010SigsetScoreWholeBodyG1, 180

- Test011SigsetScoreG2, [181](#)
- Test012SigsetScoreGaitG2, [182](#)
- Test013SigsetScoreFaceG2, [184](#)
- Test014SigsetScoreWholeBodyG2, [185](#)
- Test015SigsetBlendedScoreG1, [186](#)
- Test016SigsetScoreBlendedGaitG1, [188](#)
- Test017SigsetScoreBlendedFaceG1, [189](#)
- Test018SigsetScoreBlendedWholeBodyG1, [190](#)
- Test019SigsetBlendedScoreG2, [192](#)
- Test020SigsetScoreBlendedGaitG2, [193](#)
- Test021SigsetScoreBlendedFaceG2, [194](#)
- Test022SigsetScoreBlendedWholeBodyG2, [196](#)
- Test023MultiSigsetScoreG1, [197](#)
- Test024MultiSigsetScoreGaitG1, [202](#)
- Test025MultiSigsetScoreFaceG1, [207](#)
- Test026MultiSigsetScoreWholeBodyG1, [208](#)
- Test027MultiSigsetScoreG2, [210](#)
- Test028MultiSigsetScoreGaitG2, [211](#)
- Test029MultiSigsetScoreFaceG2, [212](#)
- Test030MultiSigsetScoreWholeBodyG2, [214](#)
- Test031MultiSigsetBlendedScoreG1, [215](#)
- Test032MultiSigsetScoreBlendedGaitG1, [216](#)
- Test033MultiSigsetScoreBlendedFaceG1, [218](#)
- Test034MultiSigsetScoreBlendedWholeBodyG1, [219](#)
- Test035MultiSigsetBlendedScoreG2, [220](#)
- Test036MultiSigsetScoreBlendedGaitG2, [222](#)
- Test037MultiSigsetScoreBlendedFaceG2, [223](#)
- Test038MultiSigsetScoreBlendedWholeBodyG2, [224](#)
- setUpClass\_main
  - briar.evaluation.full\_evaluation, [66](#)
- setUpModule
  - briar.evaluation.full\_evaluation, [66](#)
- showmat\_interactive
  - match\_matrix\_visualizer, [151](#)
- sigset/\_\_init\_\_.py, [239](#)
- sigset/parse.py, [258](#)
- sigset\_enroll
  - briar.cli.sigset, [54](#)
- sigset\_stats
  - briar.cli.sigset, [55](#)
- single\_frame\_generate
  - briar.media, [82](#)
- sortTestMethodsUsing
  - briar.evaluation.full\_evaluation, [72](#)
- source
  - BriarMedia, [118](#)
- stages\_temp
  - briar.evaluation.full\_evaluation, [72](#)
- start
  - FileVideoStream\_cv2, [135](#)
  - FileVideoStream\_imageio, [139](#)
- start\_duration
  - briar.timing, [100](#)
- start\_frame
  - ImageIterator, [150](#)
  - MediaSetIterator, [157](#)
- ThreadedVideoIterator, [230](#)
- VideoIterator, [236](#)
- start\_frames
  - MediaSetIterator, [157](#)
- status
  - briar.cli.status, [56](#)
- statusParseOptions
  - briar.cli.status, [56](#)
- stop
  - FileVideoStream\_cv2, [136](#)
  - FileVideoStream\_imageio, [139](#)
- stop\_frame
  - ImageIterator, [150](#)
  - MediaSetIterator, [157](#)
  - ThreadedVideoIterator, [230](#)
  - VideoIterator, [236](#)
- stop\_frames
  - MediaSetIterator, [157](#)
- stop\_iteration
  - ThreadedVideoIterator, [228](#)
- stopped
  - FileVideoStream\_cv2, [136](#)
  - FileVideoStream\_imageio, [140](#)
- stream
  - FileVideoStream\_cv2, [136](#)
  - FileVideoStream\_imageio, [140](#)
  - ThreadedVideoIterator, [230](#)
- stub
  - BriarClient, [116](#)
- subjectID\_int2str
  - briar.media\_converters, [93](#)
- subjectID\_str2int
  - briar.media\_converters, [94](#)
- subjectList\_list2string
  - briar.media\_converters, [94](#)
- subjectList\_string2list
  - briar.media\_converters, [94](#)
- sync\_enroll\_frames\_iter
  - BriarClient, [111](#)
- TEMPLATE\_FILE\_EXT
  - briar.briar\_cli, [18](#)
  - briar.cli.extract, [46](#)
- template\_file\_path
  - ExtractTest, [133](#)
- test
  - BriarTest, [122](#)
  - DatabaseTest, [127](#)
  - EnrollTest, [131](#)
- Test000InitialConfig, [160](#)
  - config\_reply, [163](#)
  - setUpClass, [161](#)
  - test\_01\_config\_portlist, [161](#)
  - test\_02\_port\_connections, [161](#)
  - test\_03\_num\_service\_ports, [161](#)
  - test\_04\_num\_procs\_per\_port, [162](#)
  - test\_05\_num\_threads\_per\_port, [162](#)
  - test\_06\_correct\_database\_creation, [162](#)
  - testDatasetDir, [162](#)

- testOutDir, 162
- testValidationDir, 162
- Test001SigsetEnrollProbe, 163
  - setUpClass, 164
  - test\_01\_sigset\_enroll\_probe, 164
  - test\_02\_probe\_checkpoint, 164
- Test002ProbeDatabaseMerge, 164
  - merged\_dbinfo, 166
  - setUpClass, 165
  - test\_02\_merge\_probe\_dbs, 165
  - test\_03\_checkpoint\_merged\_probe\_db, 165
  - total\_entries, 166
  - total\_failed, 166
  - total\_templates, 166
- Test003SigsetEnrollMultiProbe, 167
  - setUpClass, 167
  - test\_01\_sigset\_enroll\_probe, 167
  - test\_02\_probe\_checkpoint, 168
- Test004MultiProbeDatabaseMerge, 168
  - merged\_dbinfo, 169
  - setUpClass, 169
  - test\_02\_merge\_probe\_dbs, 169
  - test\_03\_checkpoint\_merged\_probe\_db, 169
  - total\_entries, 169
  - total\_failed, 170
  - total\_templates, 170
- Test005SigsetEnrollGalleries, 170
  - runGallery, 171
  - setUpClass, 171
  - test\_01\_sigset\_enroll\_gallery1, 171
  - test\_02\_gallery1\_partitioned\_checkpoint, 171
  - test\_03\_sigset\_enroll\_gallery2, 171
  - test\_04\_gallery2\_partitioned\_checkpoint, 172
  - test\_05\_sigset\_enroll\_gallery1, 172
  - test\_06\_gallery1\_partitioned\_checkpoint, 172
  - test\_07\_sigset\_enroll\_gallery2, 172
  - test\_08\_gallery2\_partitioned\_checkpoint, 172
- Test006GalleryDatabaseMerge, 173
  - merge\_db\_func, 173
  - setUpClass, 174
  - test\_01\_merge\_gallery1\_dbs, 174
  - test\_02\_merge\_gallery2\_dbs, 174
  - test\_03\_merge\_blended\_gallery2\_dbs, 174
  - test\_04\_merge\_blended\_gallery2\_dbs, 174
  - test\_05\_finalized\_merged\_gallery1\_db, 174
  - test\_06\_finalized\_merged\_gallery1\_db, 174
  - test\_07\_finalized\_merged\_gallery1\_db, 175
  - test\_08\_finalized\_merged\_gallery1\_db, 175
- Test007SigsetScoreG1, 175
  - setUpClass, 176
  - test\_01\_sigset\_verify\_probe\_gallery1, 176
  - test\_02\_sigset\_search\_probe\_gallery1, 176
- Test008SigsetScoreGaitG1, 176
  - setUpClass, 177
  - test\_01\_sigset\_verify\_gait\_probe\_gallery1, 177
  - test\_02\_sigset\_search\_gait\_probe\_gallery1, 177
- Test009SigsetScoreFaceG1, 178
  - setUpClass, 178
  - test\_01\_sigset\_verify\_face\_probe\_gallery1, 178
  - test\_02\_sigset\_search\_face\_probe\_gallery1, 179
- Test010SigsetScoreWholeBodyG1, 179
  - setUpClass, 180
  - test\_01\_sigset\_verify\_WB\_probe\_gallery1, 180
  - test\_02\_sigset\_search\_WB\_probe\_gallery1, 180
- Test011SigsetScoreG2, 180
  - setUpClass, 181
  - test\_01\_sigset\_verify\_probe\_gallery2, 181
  - test\_02\_sigset\_search\_probe\_gallery2, 181
- Test012SigsetScoreGaitG2, 182
  - setUpClass, 182
  - test\_01\_sigset\_verify\_gait\_probe\_gallery2, 182
  - test\_02\_sigset\_search\_gait\_probe\_gallery2, 183
- Test013SigsetScoreFaceG2, 183
  - setUpClass, 184
  - test\_01\_sigset\_verify\_face\_probe\_gallery2, 184
  - test\_02\_sigset\_search\_face\_probe\_gallery2, 184
- Test014SigsetScoreWholeBodyG2, 184
  - setUpClass, 185
  - test\_01\_sigset\_verify\_WB\_probe\_gallery2, 185
  - test\_02\_sigset\_search\_WB\_probe\_gallery2, 185
- Test015SigsetBlendedScoreG1, 186
  - setUpClass, 186
  - test\_01\_sigset\_verify\_probe\_blended\_gallery1, 186
  - test\_02\_sigset\_search\_probe\_blended\_gallery1, 187
- Test016SigsetScoreBlendedGaitG1, 187
  - setUpClass, 188
  - test\_01\_sigset\_verify\_gait\_probe\_blended\_gallery1, 188
  - test\_02\_sigset\_search\_gait\_probe\_blended\_gallery1, 188
- Test017SigsetScoreBlendedFaceG1, 188
  - setUpClass, 189
  - test\_01\_sigset\_verify\_face\_probe\_blended\_gallery1, 189
  - test\_02\_sigset\_search\_face\_probe\_blended\_gallery1, 189
- Test018SigsetScoreBlendedWholeBodyG1, 190
  - setUpClass, 190
  - test\_01\_sigset\_verify\_WB\_probe\_blended\_gallery1, 190
  - test\_02\_sigset\_search\_WB\_probe\_blended\_gallery1, 191
- Test019SigsetBlendedScoreG2, 191
  - setUpClass, 192
  - test\_01\_sigset\_verify\_probe\_blended\_gallery2, 192
  - test\_02\_sigset\_search\_probe\_blended\_gallery2, 192
- Test020SigsetScoreBlendedGaitG2, 192
  - setUpClass, 193
  - test\_01\_sigset\_verify\_gait\_probe\_blended\_gallery2, 193
  - test\_02\_sigset\_search\_gait\_probe\_blended\_gallery2, 193

- Test021SigsetScoreBlendedFaceG2, 194
  - setUpClass, 194
  - test\_01\_sigset\_verify\_face\_probe\_blended\_gallery2, 194
  - test\_02\_sigset\_search\_face\_probe\_blended\_gallery2, 195
- Test022SigsetScoreBlendedWholeBodyG2, 195
  - setUpClass, 196
  - test\_01\_sigset\_verify\_WB\_probe\_blended\_gallery2, 196
  - test\_02\_sigset\_search\_WB\_probe\_blended\_gallery2, 196
- Test023MultiSigsetScoreG1, 196
  - setUpClass, 197
  - test\_01\_sigset\_verify\_multiprobe\_gallery1, 197
  - test\_02\_sigset\_search\_multiprobe\_gallery1, 197
- Test023SigsetSearchOutputFormatting, 198
  - search\_file\_check, 199
  - test\_01\_sigset\_search\_pickle\_fileG1, 199
  - test\_02\_sigset\_search\_gait\_pickle\_fileG1, 199
  - test\_03\_sigset\_search\_face\_pickle\_fileG1, 199
  - test\_04\_sigset\_search\_wb\_pickle\_fileG1, 199
  - test\_05\_sigset\_search\_pickle\_fileG2, 199
  - test\_06\_sigset\_search\_gait\_pickle\_fileG2, 199
  - test\_07\_sigset\_search\_face\_pickle\_fileG2, 200
  - test\_08\_sigset\_search\_wb\_pickle\_fileG2, 200
  - test\_09\_sigset\_blended\_search\_pickle\_fileG1, 200
  - test\_10\_sigset\_blended\_search\_gait\_pickle\_fileG1, 200
  - test\_11\_sigset\_blended\_search\_face\_pickle\_fileG1, 200
  - test\_12\_sigset\_blended\_search\_wb\_pickle\_fileG1, 200
  - test\_13\_sigset\_blended\_search\_pickle\_fileG2, 200
  - test\_14\_sigset\_blended\_search\_gait\_pickle\_fileG2, 201
  - test\_15\_sigset\_blended\_search\_face\_pickle\_fileG2, 201
  - test\_16\_sigset\_blended\_search\_wb\_pickle\_fileG2, 201
- Test024MultiSigsetScoreGaitG1, 201
  - setUpClass, 202
  - test\_01\_sigset\_verify\_gait\_multiprobe\_gallery1, 202
  - test\_02\_sigset\_search\_gait\_multiprobe\_gallery1, 202
- Test024SigsetVerifyOutputFormatting, 203
  - score\_file\_check, 204
  - test\_01\_sigset\_verify\_pickle\_fileG1, 204
  - test\_02\_sigset\_verify\_gait\_pickle\_fileG1, 204
  - test\_03\_sigset\_verify\_face\_pickle\_fileG1, 204
  - test\_04\_sigset\_verify\_wb\_pickle\_fileG1, 204
  - test\_05\_sigset\_verify\_pickle\_fileG2, 204
  - test\_06\_sigset\_verify\_gait\_pickle\_fileG2, 204
  - test\_07\_sigset\_verify\_face\_pickle\_fileG2, 205
  - test\_08\_sigset\_verify\_wb\_pickle\_fileG2, 205
  - test\_09\_sigset\_blended\_verify\_pickle\_fileG1, 205
  - test\_10\_sigset\_blended\_verify\_gait\_pickle\_fileG1, 205
  - test\_11\_sigset\_blended\_verify\_face\_pickle\_fileG1, 205
  - test\_12\_sigset\_blended\_verify\_wb\_pickle\_fileG1, 205
  - test\_13\_sigset\_blended\_verify\_pickle\_fileG2, 205
  - test\_14\_sigset\_blended\_verify\_gait\_pickle\_fileG2, 206
  - test\_15\_sigset\_blended\_verify\_face\_pickle\_fileG2, 206
  - test\_16\_sigset\_blended\_verify\_wb\_pickle\_fileG2, 206
- Test025MultiSigsetScoreFaceG1, 206
  - setUpClass, 207
  - test\_01\_sigset\_verify\_face\_multiprobe\_gallery1, 207
  - test\_02\_sigset\_search\_face\_multiprobe\_gallery1, 207
- Test026MultiSigsetScoreWholeBodyG1, 208
  - setUpClass, 208
  - test\_01\_sigset\_verify\_WB\_multiprobe\_gallery1, 208
  - test\_02\_sigset\_search\_WB\_multiprobe\_gallery1, 209
- Test027MultiSigsetScoreG2, 209
  - setUpClass, 210
  - test\_01\_sigset\_verify\_multiprobe\_gallery2, 210
  - test\_02\_sigset\_search\_multiprobe\_gallery2, 210
- Test028MultiSigsetScoreGaitG2, 210
  - setUpClass, 211
  - test\_01\_sigset\_verify\_gait\_multiprobe\_gallery2, 211
  - test\_02\_sigset\_search\_gait\_multiprobe\_gallery2, 211
- Test029MultiSigsetScoreFaceG2, 212
  - setUpClass, 212
  - test\_01\_sigset\_verify\_face\_multiprobe\_gallery2, 212
  - test\_02\_sigset\_search\_face\_multiprobe\_gallery2, 213
- Test030MultiSigsetScoreWholeBodyG2, 213
  - setUpClass, 214
  - test\_01\_sigset\_verify\_WB\_multiprobe\_gallery2, 214
  - test\_02\_sigset\_search\_WB\_multiprobe\_gallery2, 214
- Test031MultiSigsetBlendedScoreG1, 214
  - setUpClass, 215
  - test\_01\_sigset\_verify\_multiprobe\_blended\_gallery1, 215
  - test\_02\_sigset\_search\_multiprobe\_blended\_gallery1, 215
- Test032MultiSigsetScoreBlendedGaitG1, 216
  - setUpClass, 216
  - test\_01\_sigset\_verify\_gait\_multiprobe\_blended\_gallery1, 216

- test\_02\_sigset\_search\_gait\_multiprobe\_blended\_gallery1, Test017SigsetScoreBlendedFaceG1, 189
- 217
- Test033MultiSigsetScoreBlendedFaceG1, 217
- setUpClass, 218
- test\_01\_sigset\_verify\_face\_multiprobe\_blended\_gallery1, Test009SigsetScoreFaceG1, 178
- 218
- test\_02\_sigset\_search\_face\_multiprobe\_blended\_gallery1, Test013SigsetScoreFaceG2, 184
- 218
- Test034MultiSigsetScoreBlendedWholeBodyG1, 218
- setUpClass, 219
- test\_01\_sigset\_verify\_WB\_multiprobe\_blended\_gallery1, Test036MultiSigsetScoreBlendedGaitG2, 222
- 219
- test\_02\_sigset\_search\_WB\_multiprobe\_blended\_gallery1, Test024MultiSigsetScoreGaitG1, 202
- 219
- Test035MultiSigsetBlendedScoreG2, 220
- setUpClass, 220
- test\_01\_sigset\_verify\_multiprobe\_blended\_gallery2, 220
- test\_02\_sigset\_search\_multiprobe\_blended\_gallery2, 221
- Test036MultiSigsetScoreBlendedGaitG2, 221
- setUpClass, 222
- test\_01\_sigset\_verify\_gait\_multiprobe\_blended\_gallery2, 222
- test\_02\_sigset\_search\_gait\_multiprobe\_blended\_gallery2, 222
- Test037MultiSigsetScoreBlendedFaceG2, 222
- setUpClass, 223
- test\_01\_sigset\_verify\_face\_multiprobe\_blended\_gallery2, 223
- test\_02\_sigset\_search\_face\_multiprobe\_blended\_gallery2, 223
- Test038MultiSigsetScoreBlendedWholeBodyG2, 224
- setUpClass, 224
- test\_01\_sigset\_verify\_WB\_multiprobe\_blended\_gallery2, 224
- test\_02\_sigset\_search\_WB\_multiprobe\_blended\_gallery2, 225
- test\_01\_config\_portlist
- Test000InitialConfig, 161
- test\_01\_merge\_gallery1\_dbs
- Test006GalleryDatabaseMerge, 174
- test\_01\_sigset\_enroll\_gallery1
- Test005SigsetEnrollGalleries, 171
- test\_01\_sigset\_enroll\_probe
- Test001SigsetEnrollProbe, 164
- Test003SigsetEnrollMultiProbe, 167
- test\_01\_sigset\_search\_pickle\_fileG1
- Test023SigsetSearchOutputFormatting, 199
- test\_01\_sigset\_verify\_face\_multiprobe\_blended\_gallery1
- Test033MultiSigsetScoreBlendedFaceG1, 218
- test\_01\_sigset\_verify\_face\_multiprobe\_blended\_gallery2
- Test037MultiSigsetScoreBlendedFaceG2, 223
- test\_01\_sigset\_verify\_face\_multiprobe\_gallery1
- Test025MultiSigsetScoreFaceG1, 207
- test\_01\_sigset\_verify\_face\_multiprobe\_gallery2
- Test029MultiSigsetScoreFaceG2, 212
- test\_01\_sigset\_verify\_face\_probe\_blended\_gallery1
- test\_01\_sigset\_verify\_face\_probe\_blended\_gallery2
- Test021SigsetScoreBlendedFaceG2, 194
- test\_01\_sigset\_verify\_face\_probe\_gallery1
- test\_01\_sigset\_verify\_face\_probe\_gallery2
- Test013SigsetScoreFaceG2, 184
- test\_01\_sigset\_verify\_gait\_multiprobe\_blended\_gallery1
- Test032MultiSigsetScoreBlendedGaitG1, 216
- test\_01\_sigset\_verify\_gait\_multiprobe\_blended\_gallery2
- Test036MultiSigsetScoreBlendedGaitG2, 222
- test\_01\_sigset\_verify\_gait\_multiprobe\_gallery1
- Test024MultiSigsetScoreGaitG1, 202
- test\_01\_sigset\_verify\_gait\_multiprobe\_gallery2
- Test028MultiSigsetScoreGaitG2, 211
- test\_01\_sigset\_verify\_gait\_probe\_blended\_gallery1
- Test016SigsetScoreBlendedGaitG1, 188
- test\_01\_sigset\_verify\_gait\_probe\_blended\_gallery2
- Test020SigsetScoreBlendedGaitG2, 193
- test\_01\_sigset\_verify\_gait\_probe\_gallery1
- Test008SigsetScoreGaitG1, 177
- test\_01\_sigset\_verify\_gait\_probe\_gallery2
- Test012SigsetScoreGaitG2, 182
- test\_01\_sigset\_verify\_multiprobe\_blended\_gallery1
- Test031MultiSigsetBlendedScoreG1, 215
- test\_01\_sigset\_verify\_multiprobe\_blended\_gallery2
- Test035MultiSigsetBlendedScoreG2, 220
- test\_01\_sigset\_verify\_multiprobe\_gallery1
- Test023MultiSigsetScoreG1, 197
- test\_01\_sigset\_verify\_multiprobe\_gallery2
- Test027MultiSigsetScoreG2, 210
- test\_01\_sigset\_verify\_pickle\_fileG1
- Test024SigsetVerifyOutputFormatting, 204
- test\_01\_sigset\_verify\_probe\_blended\_gallery1
- Test015SigsetBlendedScoreG1, 186
- test\_01\_sigset\_verify\_probe\_blended\_gallery2
- Test019SigsetBlendedScoreG2, 192
- test\_01\_sigset\_verify\_probe\_gallery1
- Test007SigsetScoreG1, 176
- test\_01\_sigset\_verify\_probe\_gallery2
- Test011SigsetScoreG2, 181
- test\_01\_sigset\_verify\_WB\_multiprobe\_blended\_gallery1
- Test034MultiSigsetScoreBlendedWholeBodyG1, 219
- test\_01\_sigset\_verify\_WB\_multiprobe\_blended\_gallery2
- Test038MultiSigsetScoreBlendedWholeBodyG2, 224
- test\_01\_sigset\_verify\_WB\_multiprobe\_gallery1
- Test026MultiSigsetScoreWholeBodyG1, 208
- test\_01\_sigset\_verify\_WB\_multiprobe\_gallery2
- Test030MultiSigsetScoreWholeBodyG2, 214
- test\_01\_sigset\_verify\_WB\_probe\_blended\_gallery1
- Test018SigsetScoreBlendedWholeBodyG1, 190
- test\_01\_sigset\_verify\_WB\_probe\_blended\_gallery2
- Test022SigsetScoreBlendedWholeBodyG2, 196
- test\_01\_sigset\_verify\_WB\_probe\_gallery1
- Test010SigsetScoreWholeBodyG1, 180
- test\_01\_sigset\_verify\_WB\_probe\_gallery2



- Test014SigsetScoreWholeBodyG2, [185](#)
- test\_02\_gallery1\_partitioned\_checkpoint
  - Test005SigsetEnrollGalleries, [171](#)
- test\_02\_merge\_gallery2\_dbs
  - Test006GalleryDatabaseMerge, [174](#)
- test\_02\_merge\_probe\_dbs
  - Test002ProbeDatabaseMerge, [165](#)
  - Test004MultiProbeDatabaseMerge, [169](#)
- test\_02\_port\_connections
  - Test000InitialConfig, [161](#)
- test\_02\_probe\_checkpoint
  - Test001SigsetEnrollProbe, [164](#)
  - Test003SigsetEnrollMultiProbe, [168](#)
- test\_02\_sigset\_search\_face\_multiprobe\_blended\_gallery1
  - Test033MultiSigsetScoreBlendedFaceG1, [218](#)
- test\_02\_sigset\_search\_face\_multiprobe\_blended\_gallery2
  - Test037MultiSigsetScoreBlendedFaceG2, [223](#)
- test\_02\_sigset\_search\_face\_multiprobe\_gallery1
  - Test025MultiSigsetScoreFaceG1, [207](#)
- test\_02\_sigset\_search\_face\_multiprobe\_gallery2
  - Test029MultiSigsetScoreFaceG2, [213](#)
- test\_02\_sigset\_search\_face\_probe\_blended\_gallery1
  - Test017SigsetScoreBlendedFaceG1, [189](#)
- test\_02\_sigset\_search\_face\_probe\_blended\_gallery2
  - Test021SigsetScoreBlendedFaceG2, [195](#)
- test\_02\_sigset\_search\_face\_probe\_gallery1
  - Test009SigsetScoreFaceG1, [179](#)
- test\_02\_sigset\_search\_face\_probe\_gallery2
  - Test013SigsetScoreFaceG2, [184](#)
- test\_02\_sigset\_search\_gait\_multiprobe\_blended\_gallery1
  - Test032MultiSigsetScoreBlendedGaitG1, [217](#)
- test\_02\_sigset\_search\_gait\_multiprobe\_blended\_gallery2
  - Test036MultiSigsetScoreBlendedGaitG2, [222](#)
- test\_02\_sigset\_search\_gait\_multiprobe\_gallery1
  - Test024MultiSigsetScoreGaitG1, [202](#)
- test\_02\_sigset\_search\_gait\_multiprobe\_gallery2
  - Test028MultiSigsetScoreGaitG2, [211](#)
- test\_02\_sigset\_search\_gait\_pickle\_fileG1
  - Test023SigsetSearchOutputFormatting, [199](#)
- test\_02\_sigset\_search\_gait\_probe\_blended\_gallery1
  - Test016SigsetScoreBlendedGaitG1, [188](#)
- test\_02\_sigset\_search\_gait\_probe\_blended\_gallery2
  - Test020SigsetScoreBlendedGaitG2, [193](#)
- test\_02\_sigset\_search\_gait\_probe\_gallery1
  - Test008SigsetScoreGaitG1, [177](#)
- test\_02\_sigset\_search\_gait\_probe\_gallery2
  - Test012SigsetScoreGaitG2, [183](#)
- test\_02\_sigset\_search\_multiprobe\_blended\_gallery1
  - Test031MultiSigsetBlendedScoreG1, [215](#)
- test\_02\_sigset\_search\_multiprobe\_blended\_gallery2
  - Test035MultiSigsetBlendedScoreG2, [221](#)
- test\_02\_sigset\_search\_multiprobe\_gallery1
  - Test023MultiSigsetScoreG1, [197](#)
- test\_02\_sigset\_search\_multiprobe\_gallery2
  - Test027MultiSigsetScoreG2, [210](#)
- test\_02\_sigset\_search\_probe\_blended\_gallery1
  - Test015SigsetBlendedScoreG1, [187](#)
- test\_02\_sigset\_search\_probe\_blended\_gallery2
  - Test019SigsetBlendedScoreG2, [192](#)
- test\_02\_sigset\_search\_probe\_gallery1
  - Test007SigsetScoreG1, [176](#)
- test\_02\_sigset\_search\_probe\_gallery2
  - Test011SigsetScoreG2, [181](#)
- test\_02\_sigset\_search\_WB\_multiprobe\_blended\_gallery1
  - Test034MultiSigsetScoreBlendedWholeBodyG1, [219](#)
- test\_02\_sigset\_search\_WB\_multiprobe\_blended\_gallery2
  - Test038MultiSigsetScoreBlendedWholeBodyG2, [225](#)
- test\_02\_sigset\_search\_WB\_multiprobe\_gallery1
  - Test026MultiSigsetScoreWholeBodyG1, [209](#)
- test\_02\_sigset\_search\_WB\_multiprobe\_gallery2
  - Test030MultiSigsetScoreWholeBodyG2, [214](#)
- test\_02\_sigset\_search\_WB\_probe\_blended\_gallery1
  - Test018SigsetScoreBlendedWholeBodyG1, [191](#)
- test\_02\_sigset\_search\_WB\_probe\_blended\_gallery2
  - Test022SigsetScoreBlendedWholeBodyG2, [196](#)
- test\_02\_sigset\_search\_WB\_probe\_gallery1
  - Test010SigsetScoreWholeBodyG1, [180](#)
- test\_02\_sigset\_search\_WB\_probe\_gallery2
  - Test014SigsetScoreWholeBodyG2, [185](#)
- test\_02\_sigset\_verify\_gait\_pickle\_fileG1
  - Test024SigsetVerifyOutputFormatting, [204](#)
- test\_03\_checkpoint\_merged\_probe\_db
  - Test002ProbeDatabaseMerge, [165](#)
  - Test004MultiProbeDatabaseMerge, [169](#)
- test\_03\_merge\_blended\_gallery2\_dbs
  - Test006GalleryDatabaseMerge, [174](#)
- test\_03\_num\_service\_ports
  - Test000InitialConfig, [161](#)
- test\_03\_sigset\_enroll\_gallery2
  - Test005SigsetEnrollGalleries, [171](#)
- test\_03\_sigset\_search\_face\_pickle\_fileG1
  - Test023SigsetSearchOutputFormatting, [199](#)
- test\_03\_sigset\_verify\_face\_pickle\_fileG1
  - Test024SigsetVerifyOutputFormatting, [204](#)
- test\_04\_gallery2\_partitioned\_checkpoint
  - Test005SigsetEnrollGalleries, [172](#)
- test\_04\_merge\_blended\_gallery2\_dbs
  - Test006GalleryDatabaseMerge, [174](#)
- test\_04\_num\_procs\_per\_port
  - Test000InitialConfig, [162](#)
- test\_04\_sigset\_search\_wb\_pickle\_fileG1
  - Test023SigsetSearchOutputFormatting, [199](#)
- test\_04\_sigset\_verify\_wb\_pickle\_fileG1
  - Test024SigsetVerifyOutputFormatting, [204](#)
- test\_05\_finalized\_merged\_gallery1\_db
  - Test006GalleryDatabaseMerge, [174](#)
- test\_05\_num\_threads\_per\_port
  - Test000InitialConfig, [162](#)
- test\_05\_sigset\_enroll\_gallery1
  - Test005SigsetEnrollGalleries, [172](#)
- test\_05\_sigset\_search\_pickle\_fileG2
  - Test023SigsetSearchOutputFormatting, [199](#)
- test\_05\_sigset\_verify\_pickle\_fileG2
  - Test024SigsetVerifyOutputFormatting, [204](#)

- test\_06\_correct\_database\_creation
  - Test000InitialConfig, 162
- test\_06\_finalized\_merged\_gallery1\_db
  - Test006GalleryDatabaseMerge, 174
- test\_06\_gallery1\_partitioned\_checkpoint
  - Test005SigsetEnrollGalleries, 172
- test\_06\_sigset\_search\_gait\_pickle\_fileG2
  - Test023SigsetSearchOutputFormatting, 199
- test\_06\_sigset\_verify\_gait\_pickle\_fileG2
  - Test024SigsetVerifyOutputFormatting, 204
- test\_07\_finalized\_merged\_gallery1\_db
  - Test006GalleryDatabaseMerge, 175
- test\_07\_sigset\_enroll\_gallery2
  - Test005SigsetEnrollGalleries, 172
- test\_07\_sigset\_search\_face\_pickle\_fileG2
  - Test023SigsetSearchOutputFormatting, 200
- test\_07\_sigset\_verify\_face\_pickle\_fileG2
  - Test024SigsetVerifyOutputFormatting, 205
- test\_08\_finalized\_merged\_gallery1\_db
  - Test006GalleryDatabaseMerge, 175
- test\_08\_gallery2\_partitioned\_checkpoint
  - Test005SigsetEnrollGalleries, 172
- test\_08\_sigset\_search\_wb\_pickle\_fileG2
  - Test023SigsetSearchOutputFormatting, 200
- test\_08\_sigset\_verify\_wb\_pickle\_fileG2
  - Test024SigsetVerifyOutputFormatting, 205
- test\_09\_sigset\_blended\_search\_pickle\_fileG1
  - Test023SigsetSearchOutputFormatting, 200
- test\_09\_sigset\_blended\_verify\_pickle\_fileG1
  - Test024SigsetVerifyOutputFormatting, 205
- test\_10\_sigset\_blended\_search\_gait\_pickle\_fileG1
  - Test023SigsetSearchOutputFormatting, 200
- test\_10\_sigset\_blended\_verify\_gait\_pickle\_fileG1
  - Test024SigsetVerifyOutputFormatting, 205
- test\_11\_sigset\_blended\_search\_face\_pickle\_fileG1
  - Test023SigsetSearchOutputFormatting, 200
- test\_11\_sigset\_blended\_verify\_face\_pickle\_fileG1
  - Test024SigsetVerifyOutputFormatting, 205
- test\_12\_sigset\_blended\_search\_wb\_pickle\_fileG1
  - Test023SigsetSearchOutputFormatting, 200
- test\_12\_sigset\_blended\_verify\_wb\_pickle\_fileG1
  - Test024SigsetVerifyOutputFormatting, 205
- test\_13\_sigset\_blended\_search\_pickle\_fileG2
  - Test023SigsetSearchOutputFormatting, 200
- test\_13\_sigset\_blended\_verify\_pickle\_fileG2
  - Test024SigsetVerifyOutputFormatting, 205
- test\_14\_sigset\_blended\_search\_gait\_pickle\_fileG2
  - Test023SigsetSearchOutputFormatting, 201
- test\_14\_sigset\_blended\_verify\_gait\_pickle\_fileG2
  - Test024SigsetVerifyOutputFormatting, 206
- test\_15\_sigset\_blended\_search\_face\_pickle\_fileG2
  - Test023SigsetSearchOutputFormatting, 201
- test\_15\_sigset\_blended\_verify\_face\_pickle\_fileG2
  - Test024SigsetVerifyOutputFormatting, 206
- test\_16\_sigset\_blended\_search\_wb\_pickle\_fileG2
  - Test023SigsetSearchOutputFormatting, 201
- test\_16\_sigset\_blended\_verify\_wb\_pickle\_fileG2
  - Test024SigsetVerifyOutputFormatting, 206
- test\_1\_detection\_image
  - DetectTest, 129
- test\_1\_extraction\_image
  - ExtractTest, 132
- test\_2\_detection\_image\_output
  - DetectTest, 129
- test\_2\_extraction\_image\_output
  - ExtractTest, 132
- test\_3\_detection\_image\_withreturn
  - DetectTest, 129
- test\_4\_detection\_image\_output\_withreturn
  - DetectTest, 129
- testDatasetDir
  - Test000InitialConfig, 162
- testim\_path
  - DetectTest, 130
  - ExtractTest, 133
- testOutDir
  - Test000InitialConfig, 162
- testValidationDir
  - Test000InitialConfig, 162
- thread
  - FileVideoStream\_cv2, 136
  - FileVideoStream\_imageio, 140
- ThreadedVideoIterator, 225
  - \_\_aiter\_\_, 227
  - \_\_anext\_\_, 227
  - \_\_init\_\_, 226
  - \_\_iter\_\_, 227
  - \_\_len\_\_, 227
  - \_\_next\_\_, 228
  - cap, 228
  - debug\_empty, 228
  - filepath, 229
  - fps, 229
  - frame\_count, 229
  - frame\_height, 229
  - frame\_width, 229
  - i, 229
  - isOpened, 229
  - isStarted, 229
  - length, 230
  - msec, 230
  - options, 230
  - pos, 230
  - processed, 230
  - start\_frame, 230
  - stop\_frame, 230
  - stop\_iteration, 228
  - stream, 230
- timeElapsed
  - briar.timing, 100
- timestamp
  - briar.timing, 100
- timing/\_init\_.py, 239
- total\_entries
  - Test002ProbeDatabaseMerge, 166
  - Test004MultiProbeDatabaseMerge, 169



- total\_failed
  - Test002ProbeDatabaseMerge, 166
  - Test004MultiProbeDatabaseMerge, 170
- total\_templates
  - Test002ProbeDatabaseMerge, 166
  - Test004MultiProbeDatabaseMerge, 170
- tqdm
  - BriarProgress, 121
- track
  - briar.cli.track, 59
  - BriarClient, 112
- tracking\_options2proto
  - briar.cli.detect, 39
- TRACKLET\_FILE\_EXT
  - briar.cli.track, 59
- tracklet\_list2proto
  - briar.media.converters, 95
- trackRequestConstructor
  - briar.cli.track, 59
- transform
  - FileVideoStream\_cv2, 137
  - FileVideoStream\_imageio, 141
- update
  - BriarProgress, 120
  - FileVideoStream\_cv2, 136
  - FileVideoStream\_imageio, 139
- update\_annot
  - briar.media.visualize, 84
- update\_annot\_filename\_only
  - briar.media.visualize, 84
- use\_colorama
  - briar.cli.test, 58
- USE\_SINGLE\_SUBJECT
  - briar.evaluation.full\_evaluation, 72
- USES\_FRONTEND\_MERGING
  - briar.evaluation.full\_evaluation, 72
- vector\_np2proto
  - briar.media.converters, 95
- vector\_proto2np
  - briar.media.converters, 95
- VERIFICATION\_FILE\_EXT
  - briar.cli.verify, 61
- verify
  - briar.cli.verify, 60
  - BriarClient, 112
- verify\_file\_iter
  - BriarClient, 113
- verify\_files
  - BriarClient, 114
- verify\_options2proto
  - briar.cli.verify, 61
- verifyParseOptions
  - briar.cli.verify, 61
- video\_file2proto
  - briar.media.converters, 96
- VIDEO\_FORMATS
  - BriarMedia, 118
- Videolterator, 231
  - \_\_aiter\_\_, 232
  - \_\_anext\_\_, 233
  - \_\_init\_\_, 232
  - \_\_iter\_\_, 233
  - \_\_len\_\_, 233
  - \_\_next\_\_, 234
  - cap, 234
  - debug\_empty, 234
  - filepath, 234
  - fps, 235
  - frame\_count, 235
  - frame\_height, 235
  - frame\_width, 235
  - i, 235
  - isOpened, 235
  - length, 235
  - msec, 235
  - pos, 236
  - processed, 236
  - start\_frame, 236
  - stop\_frame, 236
- visualize\_detection
  - briar.media.visualize, 84
- visualize\_matches
  - briar.media.visualize, 84
- visualize\_track
  - briar.media.visualize, 84
- viz
  - briar.cli.viz, 62
- vizParseOptions
  - briar.cli.viz, 62
- width
  - BriarMedia, 118
  - Rect, 159
- windowclick
  - briar.media.visualize, 84
- windowhover
  - briar.media.visualize, 85
- windowhover\_filename\_only
  - briar.media.visualize, 85
- x
  - Rect, 159
- xlabs
  - match\_matrix\_visualizer, 153
- xsources
  - match\_matrix\_visualizer, 153
- y
  - Rect, 160
- YELLOW
  - Fore, 142
- ylabs
  - match\_matrix\_visualizer, 153
- ysources
  - match\_matrix\_visualizer, 153