

Hyperledger Avalon

0.5.0.dev1

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Hyperledger Avalon Documentation</b>	<b>1</b>
<b>2</b>	<b>BUILD</b>	<b>3</b>
<b>3</b>	<b>README</b>	<b>5</b>
<b>4</b>	<b>README</b>	<b>7</b>
<b>5</b>	<b>README</b>	<b>9</b>
<b>6</b>	<b>Solidity Connector Test Process</b>	<b>11</b>
<b>7</b>	<b>Namespace Index</b>	<b>13</b>
7.1	Namespace List . . . . .	13
<b>8</b>	<b>Hierarchical Index</b>	<b>15</b>
8.1	Class Hierarchy . . . . .	15
<b>9</b>	<b>Class Index</b>	<b>17</b>
9.1	Class List . . . . .	17
<b>10</b>	<b>File Index</b>	<b>19</b>
10.1	File List . . . . .	19

<b>11 Namespace Documentation</b>	<b>21</b>
11.1 <a href="#">tcf::crypto::constants Namespace Reference</a>	21
11.1.1 Detailed Description	21
11.1.2 Variable Documentation	21
11.1.2.1 IV_LEN	21
11.1.2.2 RSA_PLAINTEXT_LEN	22
11.1.2.3 SYM_KEY_LEN	22
11.1.2.4 TAG_LEN	22
11.2 <a href="#">tcf::crypto::skenc Namespace Reference</a>	22
11.2.1 Detailed Description	22
11.2.2 Function Documentation	22
11.2.2.1 DecryptMessage() [1/2]	23
11.2.2.2 DecryptMessage() [2/2]	23
11.2.2.3 EncryptMessage() [1/2]	23
11.2.2.4 EncryptMessage() [2/2]	23
11.2.2.5 GenerateIV()	23
11.2.2.6 GenerateKey()	23
<b>12 Class Documentation</b>	<b>25</b>
12.1 <a href="#">_sgx_errlist_t Struct Reference</a>	25
12.2 <a href="#">avalon_sdk.connector.direct.avalon_direct_client.AvalonDirectClient Class Reference</a>	25
12.2.1 Detailed Description	25
12.2.2 Constructor & Destructor Documentation	26
12.2.2.1 __init__()	26
12.2.3 Member Function Documentation	26
12.2.3.1 get_work_order_instance()	26
12.2.3.2 get_work_order_receipt_instance()	26
12.2.3.3 get_worker_registry_instance()	27
12.2.3.4 get_worker_registry_list_instance()	27
12.3 <a href="#">avalon_sdk.connector.blockchains.ethereum.ethereum_listener.BlockchainInterface Class Reference</a>	27
12.3.1 Detailed Description	27

12.3.2	Member Function Documentation	27
12.3.2.1	newListener()	28
12.4	avalon_sdk.connector.blockchains.fabric.base.ClientBase Class Reference	28
12.4.1	Detailed Description	28
12.5	avalon_crypto_utils.signature.ClientSignature Class Reference	29
12.5.1	Detailed Description	30
12.5.2	Member Function Documentation	30
12.5.2.1	calculate_datahash()	30
12.5.2.2	calculate_request_hash()	30
12.5.2.3	generate_client_signature()	31
12.5.2.4	generate_signature()	31
12.5.2.5	verify_create_receipt_signature()	31
12.5.2.6	verify_signature()	32
12.5.2.7	verify_update_receipt_signature()	32
12.6	avalon_sdk.connector.blockchains.common.contract_response.ContractResponse Class Reference	32
12.6.1	Detailed Description	33
12.7	tcf::error::CryptoError Class Reference	33
12.8	tcf::error::DivisionByZero Class Reference	34
12.9	avalon_crypto_utils.keys.EnclaveKeys Class Reference	35
12.9.1	Detailed Description	36
12.9.2	Constructor & Destructor Documentation	36
12.9.2.1	__init__()	36
12.10	tcf::error::Error Class Reference	37
12.11	avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.EthereumWorkerRegistry↔ Impl Class Reference	38
12.11.1	Detailed Description	39
12.11.2	Constructor & Destructor Documentation	39
12.11.2.1	__init__()	39
12.11.3	Member Function Documentation	39
12.11.3.1	worker_lookup()	39
12.11.3.2	worker_lookup_next()	40

12.11.3.3 worker_register()	41
12.11.3.4 worker_retrieve()	41
12.11.3.5 worker_set_status()	42
12.11.3.6 worker_update()	42
12.12 <a href="#">avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry_list.EthereumWorkerRegistryListImpl Class Reference</a>	43
12.12.1 Detailed Description	43
12.12.2 Member Function Documentation	44
12.12.2.1 registry_add()	44
12.12.2.2 registry_lookup()	44
12.12.2.3 registry_lookup_next()	45
12.12.2.4 registry_retrieve()	45
12.12.2.5 registry_set_status()	46
12.12.2.6 registry_update()	46
12.13 <a href="#">avalon_sdk.connector.blockchains.ethereum.ethereum_work_order.EthereumWorkOrderProxyImpl Class Reference</a>	47
12.13.1 Detailed Description	48
12.13.2 Member Function Documentation	48
12.13.2.1 encryption_key_get()	48
12.13.2.2 encryption_key_retrieve()	48
12.13.2.3 encryption_key_set()	49
12.13.2.4 encryption_key_start()	49
12.13.2.5 work_order_complete()	49
12.13.2.6 work_order_get_result()	50
12.13.2.7 work_order_submit()	50
12.14 <a href="#">avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper Class Reference</a>	51
12.14.1 Detailed Description	51
12.14.2 Member Function Documentation	51
12.14.2.1 compile_source_file()	51
12.14.2.2 deploy_contract()	52
12.14.2.3 execute_transaction()	52

12.14.2.4	<a href="#">execute_unsigned_transaction()</a>	52
12.14.2.5	<a href="#">get_account_address()</a>	53
12.14.2.6	<a href="#">get_bytes_from_hex()</a>	53
12.14.2.7	<a href="#">get_chain_id()</a>	53
12.14.2.8	<a href="#">get_contract_instance()</a>	53
12.14.2.9	<a href="#">get_contract_instance_from_json()</a>	54
12.14.2.10	<a href="#">get_gas_limit()</a>	54
12.14.2.11	<a href="#">get_gas_price()</a>	54
12.14.2.12	<a href="#">get_transaction_params()</a>	54
12.14.2.13	<a href="#">get_txn_nonce()</a>	55
12.14.2.14	<a href="#">sign_execute_raw_transaction()</a>	55
12.15	<a href="#">avalon_sdk.connector.blockchains.fabric.event_listener.EventListener Class Reference</a>	55
12.15.1	Detailed Description	56
12.15.2	Member Function Documentation	56
12.15.2.1	<a href="#">config()</a>	56
12.15.2.2	<a href="#">get_single_event()</a>	57
12.15.2.3	<a href="#">start_event_handling()</a>	57
12.15.2.4	<a href="#">stop_event_handling()</a>	57
12.16	<a href="#">avalon_sdk.connector.blockchains.ethereum.ethereum_listener.EventProcessor Class Reference</a>	57
12.16.1	Detailed Description	58
12.16.2	Member Function Documentation	58
12.16.2.1	<a href="#">get_event_synchronously()</a>	58
12.16.2.2	<a href="#">handler()</a>	58
12.16.2.3	<a href="#">listener()</a>	59
12.16.2.4	<a href="#">start()</a>	59
12.16.2.5	<a href="#">stop()</a>	59
12.16.2.6	<a href="#">sync_handler()</a>	59
12.17	<a href="#">avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorkerRegistryImpl Class Reference</a>	60
12.17.1	Detailed Description	61
12.17.2	Constructor & Destructor Documentation	61

12.17.2.1 <code>__init__()</code> . . . . .	61
12.17.3 Member Function Documentation . . . . .	61
12.17.3.1 <code>worker_lookup()</code> . . . . .	61
12.17.3.2 <code>worker_lookup_next()</code> . . . . .	62
12.17.3.3 <code>worker_register()</code> . . . . .	63
12.17.3.4 <code>worker_retrieve()</code> . . . . .	63
12.17.3.5 <code>worker_set_status()</code> . . . . .	64
12.17.3.6 <code>worker_update()</code> . . . . .	64
12.18 <code>avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistryListImpl</code> Class Reference . . . . .	65
12.18.1 Detailed Description . . . . .	66
12.18.2 Constructor & Destructor Documentation . . . . .	66
12.18.2.1 <code>__init__()</code> . . . . .	66
12.18.3 Member Function Documentation . . . . .	66
12.18.3.1 <code>registry_add()</code> . . . . .	66
12.18.3.2 <code>registry_lookup()</code> . . . . .	67
12.18.3.3 <code>registry_lookup_next()</code> . . . . .	67
12.18.3.4 <code>registry_retrieve()</code> . . . . .	68
12.18.3.5 <code>registry_set_status()</code> . . . . .	68
12.18.3.6 <code>registry_update()</code> . . . . .	69
12.19 <code>avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl</code> Class Reference . . . . .	69
12.19.1 Detailed Description . . . . .	70
12.19.2 Constructor & Destructor Documentation . . . . .	70
12.19.2.1 <code>__init__()</code> . . . . .	71
12.19.3 Member Function Documentation . . . . .	71
12.19.3.1 <code>encryption_key_get()</code> . . . . .	71
12.19.3.2 <code>encryption_key_set()</code> . . . . .	71
12.19.3.3 <code>encryption_key_start()</code> . . . . .	72
12.19.3.4 <code>get_work_order_completed_event_handler()</code> . . . . .	72
12.19.3.5 <code>get_work_order_submitted_event_handler()</code> . . . . .	72
12.19.3.6 <code>work_order_complete()</code> . . . . .	73



12.19.3.7 work_order_get_result()	73
12.19.3.8 work_order_submit()	73
12.20avalon_sdk.connector.blockchains.fabric.fabric_work_order_receipt.FabricWorkOrderReceiptImpl Class Reference	74
12.20.1 Detailed Description	75
12.20.2 Constructor & Destructor Documentation	75
12.20.2.1 __init__()	75
12.20.3 Member Function Documentation	75
12.20.3.1 work_order_receipt_create()	76
12.20.3.2 work_order_receipt_lookup()	76
12.20.3.3 work_order_receipt_lookup_next()	77
12.20.3.4 work_order_receipt_retrieve()	77
12.20.3.5 work_order_receipt_update()	78
12.20.3.6 work_order_receipt_update_retrieve()	78
12.21avalon_sdk.connector.blockchains.fabric.fabric_wrapper.FabricWrapper Class Reference	79
12.21.1 Detailed Description	79
12.21.2 Constructor & Destructor Documentation	79
12.21.2.1 __init__()	79
12.21.3 Member Function Documentation	79
12.21.3.1 get_event_handler()	79
12.21.3.2 invoke_chaincode()	80
12.22FileIoExecutor Class Reference	80
12.22.1 Member Function Documentation	80
12.22.1.1 FileClose()	80
12.22.1.2 FileDelete()	81
12.22.1.3 FileOpen()	81
12.22.1.4 FileRead()	82
12.22.1.5 FileSeek()	82
12.22.1.6 FileTell()	82
12.22.1.7 FileWrite()	83
12.22.1.8 GetIoHandlerId()	83

12.22.1.9 GetMaxFileSize()	84
12.22.1.10 GetMaxIoResultSize()	84
12.23 avalon_sdk.http_client.http_jrpc_client.HttpJrpcClient Class Reference	85
12.23.1 Detailed Description	85
12.24 tcf::error::IndexError Class Reference	86
12.25 tcf::error::IOError Class Reference	87
12.26 avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl Class Reference	88
12.26.1 Detailed Description	88
12.26.2 Member Function Documentation	89
12.26.2.1 worker_lookup()	89
12.26.2.2 worker_lookup_next()	89
12.26.2.3 worker_register()	90
12.26.2.4 worker_retrieve()	90
12.26.2.5 worker_set_status()	91
12.26.2.6 worker_update()	91
12.27 avalon_sdk.connector.direct.jrpc.jrpc_work_order.JRPCWorkOrderImpl Class Reference	92
12.27.1 Detailed Description	93
12.27.2 Member Function Documentation	93
12.27.2.1 encryption_key_get()	93
12.27.2.2 encryption_key_set()	94
12.27.2.3 work_order_get_result()	94
12.27.2.4 work_order_get_result_nonblocking()	94
12.27.2.5 work_order_submit()	95
12.28 avalon_sdk.connector.direct.jrpc.jrpc_work_order_receipt.JRPCWorkOrderReceiptImpl Class Reference	95
12.28.1 Detailed Description	96
12.28.2 Member Function Documentation	96
12.28.2.1 work_order_receipt_create()	96
12.28.2.2 work_order_receipt_lookup()	97
12.28.2.3 work_order_receipt_lookup_next()	98
12.28.2.4 work_order_receipt_retrieve()	98

12.28.2.5 work_order_receipt_update()	98
12.28.2.6 work_order_receipt_update_retrieve()	99
12.29json_array_t Struct Reference	100
12.30json_object_t Struct Reference	101
12.31json_value_t Struct Reference	102
12.32json_value_value Union Reference	103
12.33avalon_sdk.connector.direct.jrpc.jrpc_util.JsonRpcErrorCode Class Reference	103
12.33.1 Detailed Description	104
12.34JsonValue Class Reference	105
12.35tcf::error::MemoryError Class Reference	106
12.36avalon_sdk.http_client.http_jrpc_client.MessageException Class Reference	107
12.36.1 Detailed Description	107
12.37tcf::error::OverflowError Class Reference	108
12.38tcf::crypto::pkenc::PrivateKey Class Reference	109
12.38.1 Constructor & Destructor Documentation	109
12.38.1.1 PrivateKey()	109
12.38.1.2 ~PrivateKey()	109
12.38.2 Member Function Documentation	109
12.38.2.1 DecryptMessage()	109
12.38.2.2 Deserialize()	110
12.38.2.3 Generate()	110
12.38.2.4 GetPublicKey()	110
12.38.2.5 operator=()	110
12.38.2.6 Serialize()	110
12.39tcf::crypto::sig::PrivateKey Class Reference	111
12.39.1 Constructor & Destructor Documentation	111
12.39.1.1 PrivateKey()	111
12.39.1.2 ~PrivateKey()	111
12.39.2 Member Function Documentation	111
12.39.2.1 Deserialize()	111

12.39.2.2 Generate()	111
12.39.2.3 GetPublicKey()	112
12.39.2.4 operator=()	112
12.39.2.5 Serialize()	112
12.39.2.6 SignMessage()	112
12.40tcf::crypto::sig::PublicKey Class Reference	112
12.40.1 Constructor & Destructor Documentation	113
12.40.1.1 PublicKey() [1/2]	113
12.40.1.2 PublicKey() [2/2]	113
12.40.1.3 ~PublicKey()	113
12.40.2 Member Function Documentation	113
12.40.2.1 Deserialize()	113
12.40.2.2 DeserializeXYFromHex()	113
12.40.2.3 operator=()	113
12.40.2.4 Serialize()	114
12.40.2.5 SerializeXYToHex()	114
12.40.2.6 VerifySignature()	114
12.41tcf::crypto::pkenc::PublicKey Class Reference	114
12.41.1 Constructor & Destructor Documentation	115
12.41.1.1 PublicKey() [1/2]	115
12.41.1.2 PublicKey() [2/2]	115
12.41.1.3 ~PublicKey()	115
12.41.2 Member Function Documentation	115
12.41.2.1 Deserialize()	115
12.41.2.2 EncryptMessage()	115
12.41.2.3 operator=()	116
12.41.2.4 Serialize()	116
12.42avalon_sdk.work_order_receipt.work_order_receipt.ReceiptCreateStatus Class Reference	116
12.42.1 Detailed Description	117
12.43avalon_sdk.registry.registry_status.RegistryStatus Class Reference	118

12.43.1 Detailed Description . . . . .	118
12.44tcf::error::RuntimeError Class Reference . . . . .	119
12.45avalon_sdk.worker.worker_details.SGXWorkerDetails Class Reference . . . . .	120
12.45.1 Detailed Description . . . . .	121
12.45.2 Member Function Documentation . . . . .	121
12.45.2.1 load_worker() . . . . .	121
12.46StringArray Class Reference . . . . .	121
12.46.1 Detailed Description . . . . .	122
12.47tcf::error::SystemBusyError Class Reference . . . . .	122
12.48tcf::error::SystemError Class Reference . . . . .	123
12.49test_ethereum_worker_registry_impl.TestEthereumWorkerRegistryImpl Class Reference . . . . .	124
12.50test_ethereum_worker_registry_list_impl.TestEthereumWorkerRegistryListImpl Class Reference . . . . .	125
12.51test_ethereum_work_order_impl.TestEthereumWorkOrderProxyImpl Class Reference . . . . .	126
12.51.1 Member Function Documentation . . . . .	127
12.51.1.1 test_is_wo_id_in_event_error_result() . . . . .	127
12.51.1.2 test_is_wo_id_in_event_positive() . . . . .	128
12.51.1.3 test_is_wo_id_in_event_wo_id_not_matched() . . . . .	128
12.51.1.4 test_work_order_complete() . . . . .	128
12.51.1.5 test_work_order_complete_error() . . . . .	128
12.52test_fabric_worker_registry_impl.TestFabricWorkerRegistryImpl Class Reference . . . . .	129
12.53test_fabric_worker_registry_list_impl.TestFabricWorkerRegistryListImpl Class Reference . . . . .	130
12.54test_worker_registry_jrpc_impl.TestWorkerRegistryJRPCImpl Class Reference . . . . .	131
12.55test_work_order_encryption_key_jrpc_impl.TestWorkOrderEncryptionKeyJRPCImpl Class Reference . . . . .	132
12.56test_work_order_jrpc_impl.TestWorkOrderJRPCImpl Class Reference . . . . .	133
12.57tcf::utility::Timer Class Reference . . . . .	133
12.58avalon_sdk.connector.blockchains.fabric.tx_committer.TxCommitter Class Reference . . . . .	134
12.58.1 Detailed Description . . . . .	134
12.58.2 Member Function Documentation . . . . .	135
12.58.2.1 cc_invoke() . . . . .	135
12.58.2.2 cc_query() . . . . .	135

12.59	<a href="#">tcf::error::UnknownError Class Reference</a>	136
12.60	<a href="#">tcf::error::ValueError Class Reference</a>	137
12.61	<a href="#">avalon_sdk.worker.worker_details.WorkerDetails Class Reference</a>	138
12.61.1	Detailed Description	138
12.61.2	Constructor & Destructor Documentation	138
12.61.2.1	<code>__init__()</code>	139
12.61.3	Member Function Documentation	139
12.61.3.1	<code>validate_worker_details()</code>	139
12.62	<a href="#">avalon_sdk.connector.interfaces.worker_registry.WorkerRegistry Class Reference</a>	139
12.62.1	Detailed Description	140
12.62.2	Member Function Documentation	140
12.62.2.1	<code>worker_lookup()</code>	140
12.62.2.2	<code>worker_lookup_next()</code>	141
12.62.2.3	<code>worker_register()</code>	142
12.62.2.4	<code>worker_retrieve()</code>	142
12.62.2.5	<code>worker_set_status()</code>	143
12.62.2.6	<code>worker_update()</code>	143
12.63	<a href="#">avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList Class Reference</a>	144
12.63.1	Detailed Description	144
12.63.2	Member Function Documentation	145
12.63.2.1	<code>registry_add()</code>	145
12.63.2.2	<code>registry_lookup()</code>	145
12.63.2.3	<code>registry_lookup_next()</code>	146
12.63.2.4	<code>registry_retrieve()</code>	146
12.63.2.5	<code>registry_set_status()</code>	147
12.63.2.6	<code>registry_update()</code>	147
12.64	<a href="#">avalon_sdk.worker.worker_details.WorkerStatus Class Reference</a>	148
12.64.1	Detailed Description	148
12.65	<a href="#">avalon_sdk.worker.worker_details.WorkerType Class Reference</a>	149
12.65.1	Detailed Description	149

12.66	tcf::error::WorkloadError Class Reference . . . . .	150
12.67	WorkloadProcessor Class Reference . . . . .	151
12.67.1	Detailed Description . . . . .	151
12.67.2	Member Function Documentation . . . . .	151
12.67.2.1	Clone() . . . . .	151
12.67.2.2	CreateWorkloadProcessor() . . . . .	151
12.67.2.3	ProcessWorkOrder() . . . . .	152
12.67.2.4	RegisterWorkloadProcessor() . . . . .	152
12.67.3	Member Data Documentation . . . . .	153
12.67.3.1	workload_processor_table . . . . .	153
12.68	avalon_sdk.connector.interfaces.work_order.WorkOrder Class Reference . . . . .	153
12.68.1	Detailed Description . . . . .	154
12.68.2	Member Function Documentation . . . . .	154
12.68.2.1	encryption_key_get() . . . . .	154
12.68.2.2	encryption_key_set() . . . . .	155
12.68.2.3	work_order_get_result() . . . . .	155
12.68.2.4	work_order_submit() . . . . .	156
12.69	tcf::WorkOrderData Class Reference . . . . .	158
12.69.1	Detailed Description . . . . .	158
12.70	avalon_sdk.work_order.work_order_params.WorkOrderParams Class Reference . . . . .	158
12.70.1	Member Function Documentation . . . . .	159
12.70.1.1	add_encrypted_request_hash() . . . . .	159
12.70.1.2	add_in_data() . . . . .	159
12.70.1.3	add_out_data() . . . . .	160
12.70.1.4	add_requester_signature() . . . . .	160
12.70.1.5	get_in_data() . . . . .	160
12.70.1.6	get_out_data() . . . . .	160
12.70.1.7	get_params() . . . . .	161
12.70.1.8	get_requester_id() . . . . .	161
12.70.1.9	get_requester_nonce() . . . . .	161

12.70.1.10	<a href="#">get_session_key_iv()</a>	161
12.70.1.11	<a href="#">get_work_order_id()</a>	161
12.70.1.12	<a href="#">get_worker_id()</a>	162
12.70.1.13	<a href="#">get_workload_id()</a>	162
12.70.1.14	<a href="#">set_data_encryption_algorithm()</a>	162
12.70.1.15	<a href="#">set_encrypted_session_key()</a>	162
12.70.1.16	<a href="#">set_notify_uri()</a>	162
12.70.1.17	<a href="#">set_payload_format()</a>	163
12.70.1.18	<a href="#">set_requester_id()</a>	163
12.70.1.19	<a href="#">set_requester_nonce()</a>	163
12.70.1.20	<a href="#">set_response_timeout_msecs()</a>	163
12.70.1.21	<a href="#">set_result_uri()</a>	163
12.70.1.22	<a href="#">set_session_key_iv()</a>	164
12.70.1.23	<a href="#">set_verifying_key()</a>	164
12.70.1.24	<a href="#">set_work_order_id()</a>	164
12.70.1.25	<a href="#">set_worker_encryption_key()</a>	164
12.70.1.26	<a href="#">set_worker_id()</a>	164
12.70.1.27	<a href="#">set_workload_id()</a>	165
12.70.1.28	<a href="#">to_jrpc_string()</a>	165
12.70.1.29	<a href="#">to_string()</a>	165
12.71	<a href="#">avalon_sdk.connector.interfaces.work_order_proxy.WorkOrderProxy Class Reference</a>	166
12.71.1	<a href="#">Detailed Description</a>	167
12.71.2	<a href="#">Member Function Documentation</a>	167
12.71.2.1	<a href="#">encryption_key_start()</a>	167
12.71.2.2	<a href="#">work_order_complete()</a>	167
12.72	<a href="#">avalon_sdk.connector.interfaces.work_order_receipt.WorkOrderReceipt Class Reference</a>	168
12.72.1	<a href="#">Detailed Description</a>	169
12.72.2	<a href="#">Member Function Documentation</a>	169
12.72.2.1	<a href="#">work_order_receipt_create()</a>	169
12.72.2.2	<a href="#">work_order_receipt_lookup()</a>	170



12.72.2.3 work_order_receipt_lookup_next()	170
12.72.2.4 work_order_receipt_retrieve()	171
12.72.2.5 work_order_receipt_update() [1/2]	171
12.72.2.6 work_order_receipt_update() [2/2]	172
12.72.2.7 work_order_receipt_update_retrieve()	173
12.73avalon_sdk.work_order_receipt.work_order_receipt.WorkOrderReceiptRequest Class Reference	174
12.73.1 Detailed Description	174
12.73.2 Member Function Documentation	174
12.73.2.1 create_receipt()	174
12.73.2.2 update_receipt()	175
12.74avalon_sdk.work_order.work_order_request_validator.WorkOrderRequestValidator Class Reference	175
12.74.1 Detailed Description	175
12.74.2 Constructor & Destructor Documentation	175
12.74.2.1 __init__()	175
12.74.3 Member Function Documentation	176
12.74.3.1 validate_data_format()	176
12.74.3.2 validate_parameters()	176
<b>13 File Documentation</b>	<b>177</b>
13.1 /home/dano/git/avalon/common/cpp/crypto/crypto_utils.h File Reference	177
13.1.1 Detailed Description	178
13.1.2 Function Documentation	178
13.1.2.1 ComputeMessageHash()	178
13.1.2.2 CreateHexEncodedEncryptionKey()	178
13.1.2.3 DecryptData()	179
13.1.2.4 EncryptData()	179
13.1.2.5 RandomBitString()	179
13.2 /home/dano/git/avalon/common/cpp/crypto/pkenc_private_key.cpp File Reference	179
13.2.1 Detailed Description	180
13.2.2 Function Documentation	180
13.2.2.1 deserializeRSAPrivateKey()	180

13.3	<a href="#">/home/dano/git/avalon/common/cpp/crypto/pkenc_private_key.h File Reference</a>	180
13.3.1	Detailed Description	181
13.4	<a href="#">/home/dano/git/avalon/common/cpp/crypto/pkenc_public_key.cpp File Reference</a>	181
13.4.1	Detailed Description	182
13.4.2	Function Documentation	182
13.4.2.1	<code>deserializeRSAPublicKey()</code>	182
13.5	<a href="#">/home/dano/git/avalon/common/cpp/crypto/pkenc_public_key.h File Reference</a>	182
13.5.1	Detailed Description	183
13.6	<a href="#">/home/dano/git/avalon/common/cpp/crypto/SAVE/verify_certificate.h File Reference</a>	183
13.6.1	Detailed Description	183
13.6.2	Function Documentation	183
13.6.2.1	<code>verify_certificate_chain()</code>	183
13.7	<a href="#">/home/dano/git/avalon/common/cpp/crypto/verify_certificate.h File Reference</a>	184
13.7.1	Detailed Description	184
13.7.2	Function Documentation	184
13.7.2.1	<code>verify_certificate_chain()</code>	184
13.8	<a href="#">/home/dano/git/avalon/common/cpp/crypto/SAVE/verify_signature.h File Reference</a>	184
13.8.1	Detailed Description	185
13.8.2	Function Documentation	185
13.8.2.1	<code>verify_signature()</code>	185
13.9	<a href="#">/home/dano/git/avalon/common/cpp/crypto/verify_signature.h File Reference</a>	185
13.9.1	Detailed Description	186
13.9.2	Function Documentation	186
13.9.2.1	<code>verify_signature()</code>	186
13.10	<a href="#">/home/dano/git/avalon/common/cpp/crypto/sig_private_key.cpp File Reference</a>	186
13.10.1	Detailed Description	187
13.10.2	Function Documentation	187
13.10.2.1	<code>deserializeECDSAPrivateKey()</code>	187
13.11	<a href="#">/home/dano/git/avalon/common/cpp/crypto/sig_private_key.h File Reference</a>	188
13.11.1	Detailed Description	188

13.12/home/dano/git/avalon/common/cpp/crypto/sig_public_key.cpp File Reference . . . . .	189
13.12.1 Detailed Description . . . . .	189
13.12.2 Function Documentation . . . . .	189
13.12.2.1 deserializeECDSAPublicKey() . . . . .	190
13.13/home/dano/git/avalon/common/cpp/crypto/sig_public_key.h File Reference . . . . .	190
13.13.1 Detailed Description . . . . .	191
13.14/home/dano/git/avalon/common/cpp/crypto/skenc.cpp File Reference . . . . .	191
13.14.1 Detailed Description . . . . .	191
13.15/home/dano/git/avalon/common/cpp/crypto/skenc.h File Reference . . . . .	192
13.15.1 Detailed Description . . . . .	193
13.16/home/dano/git/avalon/common/cpp/error.h File Reference . . . . .	193
13.16.1 Detailed Description . . . . .	194
13.17/home/dano/git/avalon/common/cpp/hex_string.h File Reference . . . . .	194
13.17.1 Detailed Description . . . . .	195
13.17.2 Macro Definition Documentation . . . . .	195
13.17.2.1 HEX_STRING_SIZE . . . . .	195
13.17.3 Function Documentation . . . . .	195
13.17.3.1 BinaryToHexString() . . . . .	195
13.17.3.2 HexStringToBinary() . . . . .	196
13.18/home/dano/git/avalon/common/cpp/json_utils.h File Reference . . . . .	196
13.18.1 Detailed Description . . . . .	196
13.19/home/dano/git/avalon/common/cpp/jsonvalue.h File Reference . . . . .	196
13.19.1 Detailed Description . . . . .	197
13.20/home/dano/git/avalon/common/cpp/packages/base64/base64.cpp File Reference . . . . .	197
13.20.1 Detailed Description . . . . .	197
13.20.2 Function Documentation . . . . .	197
13.20.2.1 base64_decode() . . . . .	197
13.20.2.2 base64_encode() . . . . .	198
13.21/home/dano/git/avalon/common/cpp/packages/base64/base64.h File Reference . . . . .	198
13.21.1 Detailed Description . . . . .	199

13.21.2 Function Documentation . . . . .	199
13.21.2.1 <code>base64_decode()</code> . . . . .	199
13.21.2.2 <code>base64_encode()</code> . . . . .	199
13.22/ <code>home/dano/git/avalon/common/cpp/tcf_error.h</code> File Reference . . . . .	200
13.22.1 Detailed Description . . . . .	201
13.22.2 Enumeration Type Documentation . . . . .	201
13.22.2.1 <code>tcf_err_t</code> . . . . .	201
13.23/ <code>home/dano/git/avalon/common/cpp/timer.h</code> File Reference . . . . .	201
13.23.1 Detailed Description . . . . .	202
13.24/ <code>home/dano/git/avalon/common/cpp/types.cpp</code> File Reference . . . . .	202
13.24.1 Detailed Description . . . . .	203
13.24.2 Function Documentation . . . . .	203
13.24.2.1 <code>Base64EncodedStringToByteArray()</code> . . . . .	203
13.24.2.2 <code>ByteArrayToBase64EncodedString()</code> . . . . .	203
13.24.2.3 <code>ByteArrayToHexEncodedString()</code> . . . . .	204
13.24.2.4 <code>ByteArrayToString()</code> . . . . .	204
13.24.2.5 <code>ByteArrayToStringArray()</code> . . . . .	204
13.24.2.6 <code>HexEncodedStringToByteArray()</code> . . . . .	204
13.25/ <code>home/dano/git/avalon/common/cpp/types.h</code> File Reference . . . . .	204
13.25.1 Detailed Description . . . . .	205
13.25.2 Typedef Documentation . . . . .	205
13.25.2.1 <code>Base64EncodedString</code> . . . . .	205
13.25.2.2 <code>ByteArray</code> . . . . .	206
13.25.2.3 <code>HexEncodedString</code> . . . . .	206
13.25.3 Enumeration Type Documentation . . . . .	206
13.25.3.1 <code>EnclaveType</code> . . . . .	206
13.25.4 Function Documentation . . . . .	206
13.25.4.1 <code>Base64EncodedStringToByteArray()</code> . . . . .	206
13.25.4.2 <code>ByteArrayToBase64EncodedString()</code> . . . . .	206
13.25.4.3 <code>ByteArrayToHexEncodedString()</code> . . . . .	206

13.25.4.4 ByteArrayToString()	207
13.25.4.5 ByteArrayToStringArray()	207
13.25.4.6 HexEncodedStringToByteArray()	207
13.26/home/dano/git/avalon/common/cpp/utils.cpp File Reference	207
13.26.1 Detailed Description	208
13.26.2 Function Documentation	208
13.26.2.1 ByteArrayToStr()	208
13.26.2.2 StrToByteArray()	208
13.27/home/dano/git/avalon/common/cpp/utils.h File Reference	208
13.27.1 Detailed Description	209
13.27.2 Function Documentation	209
13.27.2.1 ByteArrayToStr()	209
13.27.2.2 StrToByteArray()	209
13.28/home/dano/git/avalon/common/cpp/zero.h File Reference	210
13.28.1 Detailed Description	210
13.29/home/dano/git/avalon/common/sgx_workload/iohandler/file_io.cpp File Reference	210
13.29.1 Detailed Description	211
13.30/home/dano/git/avalon/common/sgx_workload/iohandler/file_io.h File Reference	211
13.30.1 Detailed Description	212
13.31/home/dano/git/avalon/common/sgx_workload/iohandler/file_io_wrapper.cpp File Reference	212
13.31.1 Detailed Description	213
13.31.2 Function Documentation	213
13.31.2.1 Delete()	213
13.31.2.2 Read()	214
13.31.2.3 Write()	214
13.32/home/dano/git/avalon/common/sgx_workload/iohandler/file_io_wrapper.h File Reference	214
13.32.1 Detailed Description	215
13.32.2 Function Documentation	215
13.32.2.1 Delete()	216
13.32.2.2 Read()	216
13.32.2.3 Write()	216
13.33/home/dano/git/avalon/common/sgx_workload/workload/work_order_data.h File Reference	217
13.33.1 Detailed Description	218
13.34/home/dano/git/avalon/common/sgx_workload/workload/workload_processor.cpp File Reference	218
13.34.1 Detailed Description	219
13.35/home/dano/git/avalon/common/sgx_workload/workload/workload_processor.h File Reference	219
13.35.1 Detailed Description	221
13.35.2 Macro Definition Documentation	221
13.35.2.1 IMPL_WORKLOAD_PROCESSOR_CLONE	221
13.35.2.2 REGISTER_WORKLOAD_PROCESSOR	221



# Chapter 1

## Hyperledger Avalon Documentation

### Introduction

- [README](#). Overview of Avalon and its source code
- [FAQ](#). Frequently-asked questions with answers about Avalon
  - [Glossary](#)
  - [Videos](#)
- [Avalon Proposal](#). Avalon Proposal, initial members, motivation, and proposed solutions (2019)
- ["Ecosystem Support for EEA Trusted Compute Specification v1.0 Improves Blockchain Privacy and Scalability"](#). Introductory blog by Michael Reed (2019)
- [Introduction to Hyperledger Avalon video \(20:24\)](#)

### Community

- [Project Wiki](#)
- [RocketChat](#)
- [Email list](#)
- [JIRA feature & bug tracking](#)

### Tutorial

- [Workload Application Tutorial](#)
- [Example Applications](#)

### Source Code

- [Avalon source code repository, `https://github.com/hyperledger/avalon`](#)
- [Building source code](#)
- [Example Avalon applications](#)
- [Contributing source code](#)

## SDK Reference Manual

The [Hyperledger Avalon SDK Reference Manual](#) (also available as a [PDF](#) file) documents the SDK used to create worker order requestors (clients) and processors.

The Avalon SDK Reference Manual is generated with Doxygen. To generate the Reference Manual, type the following:

```
cd $TCF_HOME/docs # this directory
sudo apt-get update
sudo apt-get install -y make doxygen texlive-full graphviz
make
```

Documentation generated will be here:

- `$TCF_HOME/docs/refman/html/HTML` documentation
- `$TCF_HOME/docs/refman/man/man3man page` documentation
- `$TCF_HOME/docs/refman/latexLaTeX` documentation
- `$TCF_HOME/docs/refman/refman.pdfPDF` documentation (generated from LaTeX)

## Reference

- [Hyperledger Avalon Architecture Overview](#). Overview of Avalon architecture by Eugene Yarmosh (2020)
- [Off-Chain Trusted Compute Specification](#) defined by Enterprise Ethereum Alliance (EEA) Task Force
- [Cryptography](#). Cryptographic primitives used, libraries used, and implementation



## Chapter 2

# BUILD

### Building the common libraries

Make sure you have environment variables `SGX_SDK` and `SGX_SSL` defined (see `../../BUILD.md` "BUILD.md") and then run

```
mkdir build
cd build
cmake .. -G "Unix Makefiles" && make
```



## Chapter 3

# README

### Common Crypto library documentation

This `common/cpp/crypto/` directory contains cryptographic code used by untrusted and trusted (Intel SGX Enclave) code.

This code is written in C++, but a Python wrapper is also available (see `../README.md` "`../README.md`")

Avalon applications are free to use third-party cryptographic implementations (such as what a programming language binding may provide) or the cryptographic interfaces provided here.

### Software Components Required

OpenSSL 1.1 library and Intel SGX OpenSSL library built from OpenSSL 1.1:

- <https://www.openssl.org/>
- <https://github.com/intel/intel-sgx-ssl>

### Cryptographic Primitives Used

Primitive	Algorithm	Keysize	Comments
Digital signature	ECDSA-SECP256K1	256	(1) (2)
Asymmetric encryption	RSA-OAEP	3072	(1)
Authenticated encryption	AES-GCM	256	96b IV, 128b tag
Digest	SHA-256	256	(2)
Digest	KECCACK	256	(2) Differs from SHA-3

(1) Not PQ resistant

(2) Blockchain legacy algorithm

## Cryptographic Primitive Usage

- **SHA-256** Computing digests of the work order request and response
- **KECCAK-256** Computing digests of the work order request and response or Ethereum raw transactions Packet bytes
- **AES-GCM-256** Encrypts data items within work order request and response. It also used to encrypt a request digest and custom data encryption keys
- **RSA-OAEP-3072** Encrypt symmetric data encryption keys
- **ECDSA-SECP256K1** Signs work order response digest and worker's encryption RSA-OAEP public key

## Implementation of Cryptographic Elements

Cryptographic elements include cryptographic keys, signature, ciphertexts, plaintexts, hashes, and random bit-strings.

Element	Implementation	Representation	Serialize/Deserialize function?
ECDSA public key	C++ class	Custom object	Yes, PEM encoding and 65-byte Bitcoin Hex format
ECDSA private key	C++ class	Custom object	Yes, PEM encoding
ECDSA signature	C++ string	DER binary	No, user defined
RSA public key	C++ class	Custom object	Yes, PEM encoding
RSA private key	C++ class	Custom object	Yes, PEM encoding
RSA ciphertext	C++ string	raw binary	No, user defined
RSA plaintext	C++ string	raw binary	No, user defined
AES-GCM key	C++ string	raw binary	No, user defined
AES-GCM iv	C++ string	raw binary	No, user defined
AES-GCM ciphertext+tag	C++ string	raw binary	No, user defined
AES-GCM plaintext	C++ string	raw binary	No, user defined
SHA-256 digest	C++ string	raw binary	No, user defined
Random bitstring	C++ string	raw binary	No, user defined

## Security notes

- **AES-GCM** When using of AES-GCM inside Intel SGX enclaves to preserve confidentiality and integrity of data to be stored outside of the Intel SGX enclaves a different unique or random 12-byte IV must be used for each encrypted message. At most  $2^{32}$  distinct IVs can be used until the key needs to be regenerated for security. This limitation can possibly be mitigated in the future by using alternatives to AES-GCM like AES-GCM SIV.

## Chapter 4

# README

### Purpose of Common

The common directory contains source code shared by untrusted and trusted (Intel SGX Enclave) code.

### Dependencies:

1. OpenSSL 1.1 <https://www.openssl.org/>
2. Intel SGX OpenSSL library built from OpenSSL 1.1 <https://github.com/intel/intel-sgx-ssl>
3. Intel SGX SDK <https://software.intel.com/en-us/sgx-sdk/download>

### Source Directories

#### Dir Content

`crypto/` \*.cpp,\*.h for OpenSSL based crypto functions. For more information, see `crypto/README.md`

`.` \*.cpp,\*.h error handling and common types

`packages/base64/` \*.cpp,\*.h of Renee Nyffinger base64 encoding/decoding

`packages/parson/` \*.cpp,\*.h of Parson JSON encoding/decoding

### Python Wrapper

The Python SWIG wrapper exports the functions and classes defined in `crypto.h`, `tcf_error.h` and `types.h`. Several classes and functions are renamed. Check `common/python/crypto_utils/crypto/crypto.i` for details.



## Chapter 5

# README

### Purpose of Common

The common directory contains source code shared by trusted (Intel SGX Enclave) code and different workloads(↔ Example workloads).

### Dependencies:

1. SGX SDK <https://software.intel.com/en-us/sgx-sdk/download>
2. SGX OpenSSL library built from OpenSSL 1.1 <https://github.com/intel/intel-sgx-ssl>

### Source Directories

#### Dir Content

sgx/iohandler/ \*.cpp,\*.h files are custom iohandlers which help workloads to execute IO operations from the Intel SGX enclave

sgx/workload/ work\_order\_data.cpp,[work\\_order\\_data.h](#) files are wrapper files for work order data [workload](#)↔  
[\\_processor.cpp](#), [workload\\_processor.h](#) are workload processor which overrides function exposed by work order interface and also facilitates auto registration of workloads





## Chapter 6

# Solidity Connector Test Process

1. You have two choices for building Avalon: Docker-based build (recommended) or standalone build.
  - For standalone builds, follow instructions in the "Standalone based Build" section of the the [build document](#). Then continue with the next step, step 2.
  - For Docker-based builds, follow instructions in the "Docker-based Build and Execution" section of the the [build document](#) through step 4 (Docker container shell). Then continue with step 8, below.
2. (Standalone builds only) If needed, update the Ethereum account and direct registry contract information in `sdk/avalon_sdk/tcf_connector.toml`
3. (Standalone builds only) Install Python 3.6.8 if not currently installed. Determine your Python version with `python3 --version`. If it is not installed, install it as follows:  

```
“bash wget https://www.python.org/ftp/python/3.6.8/Python-3.6.8.tgz tar -xvf Python-3.6.8.tgz cd Python-3.6.8 ./configure make sudo make install make sure “
```
4. (Standalone builds only) Install the Solidity compiler to compile Solidity contracts from Python: 

```
“bash pip3 install --upgrade py-solc-x python3 -m solcx.install v0.5.15 “
```
5. (Standalone builds only) To run smart contracts using a Ropsten network account, first install the MetaMask Chrome plugin to your Chrome web browser and create an account in the Ropsten network
6. (Standalone builds only) After creating an account, make sure to add fake ether to the account using:
  - <https://faucet.metamask.io/>
  - <https://blog.bankex.org/how-to-buy-ethereum-using-metamask-ccea0703daec>
7. Install web3.py, which is an Ethereum Python client that interacts with the Ethereum network. For more information about web3.py, see <https://web3py.readthedocs.io/en/stable/quickstart.html>  

```
“bash pip install web3 “
```
8. Run `cd $TCF_HOME/examples/common/python/connectors/ethereum`
9. Fill in your Ropsten testnet address in `eth_account` in `sdk/avalon_sdk/tcf_connector.toml`
10. Deploy solidity contracts to Ropsten network using `eth_cli.py`  

```
“bash ./eth_cli.py “
```

The above command will display the contract instance address for `direct_registry_contract_↵` address and `worker_registry_contract_address`
11. Fill in your your contract addresses `direct_registry_contract_address` and `worker_↵` registry\_contract\_address in `sdk/avalon_sdk/tcf_connector.toml`

12. Test the DirectRegistry and WorkerRegistry contracts with: `“bash cd $TCF_HOME/examples/common/python/connectors/ether_/_tests python3 test_ethereum_worker_registry_impl.py python3 test_ethereum_worker_registry_list_impl.py “`
13. Test echo client with direct mode using Ropsten test network. `“bash cd $TCF_HOME/examples/apps/echo/client ./echo_client.py -m "Hello world" “`

## Chapter 7

# Namespace Index

### 7.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

<a href="#">tcf::crypto::constants</a>	21
<a href="#">tcf::crypto::skenc</a>	22



## Chapter 8

# Hierarchical Index

### 8.1 Class Hierarchy

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

_sgx_errlist_t . . . . .	25
avalon_sdk.connector.direct.avalon_direct_client.AvalonDirectClient . . . . .	25
avalon_sdk.connector.blockchains.ethereum.ethereum_listener.BlockchainInterface . . . . .	27
avalon_sdk.connector.blockchains.fabric.base.ClientBase . . . . .	28
avalon_sdk.connector.blockchains.fabric.event_listener.EventListener . . . . .	55
avalon_sdk.connector.blockchains.fabric.tx_committer.TxCommitter . . . . .	134
avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper . . . . .	51
avalon_sdk.connector.blockchains.ethereum.ethereum_listener.EventProcessor . . . . .	57
Exception	
avalon_sdk.http_client.http_jrpc_client.MessageException . . . . .	107
avalon_sdk.connector.blockchains.fabric.fabric_wrapper.FabricWrapper . . . . .	79
FileIoExecutor . . . . .	80
json_array_t . . . . .	100
json_object_t . . . . .	101
json_value_t . . . . .	102
json_value_value . . . . .	103
JsonValue . . . . .	105
object	
avalon_crypto_utils.keys.EnclaveKeys . . . . .	35
avalon_crypto_utils.signature.ClientSignature . . . . .	29
avalon_sdk.http_client.http_jrpc_client.HttpJrpcClient . . . . .	85
tcf::crypto::pkenc::PrivateKey . . . . .	109
tcf::crypto::sig::PrivateKey . . . . .	111
tcf::crypto::sig::PublicKey . . . . .	112
tcf::crypto::pkenc::PublicKey . . . . .	114
runtime_error	
tcf::error::Error . . . . .	37
tcf::error::CryptoError . . . . .	33
tcf::error::DivisionByZero . . . . .	34
tcf::error::IndexError . . . . .	86
tcf::error::IOError . . . . .	87
tcf::error::MemoryError . . . . .	106
tcf::error::OverflowError . . . . .	108
tcf::error::RuntimeError . . . . .	119
tcf::error::SystemBusyError . . . . .	122

tcf::error::SystemError . . . . .	123
tcf::error::UnknownError . . . . .	136
tcf::error::ValueError . . . . .	137
tcf::error::WorkloadError . . . . .	150
TestCase	
test_ethereum_work_order_impl.TestEthereumWorkOrderProxyImpl . . . . .	126
test_ethereum_worker_registry_impl.TestEthereumWorkerRegistryImpl . . . . .	124
test_ethereum_worker_registry_list_impl.TestEthereumWorkerRegistryListImpl . . . . .	125
test_fabric_worker_registry_impl.TestFabricWorkerRegistryImpl . . . . .	129
test_fabric_worker_registry_list_impl.TestFabricWorkerRegistryListImpl . . . . .	130
test_work_order_encryption_key_jrpc_impl.TestWorkOrderEncryptionKeyJRPCImpl . . . . .	132
test_work_order_jrpc_impl.TestWorkOrderJRPCImpl . . . . .	133
test_worker_registry_jrpc_impl.TestWorkerRegistryJRPCImpl . . . . .	131
tcf::utility::Timer . . . . .	133
vector	
StringArray . . . . .	121
avalon_sdk.worker.worker_details.WorkerDetails . . . . .	138
avalon_sdk.worker.worker_details.SGXWorkerDetails . . . . .	120
WorkloadProcessor . . . . .	151
tcf::WorkOrderData . . . . .	158
avalon_sdk.work_order.work_order_params.WorkOrderParams . . . . .	158
avalon_sdk.work_order_receipt.work_order_receipt.WorkOrderReceiptRequest . . . . .	174
avalon_sdk.work_order.work_order_request_validator.WorkOrderRequestValidator . . . . .	175
ABC	
avalon_sdk.connector.interfaces.work_order.WorkOrder . . . . .	153
avalon_sdk.connector.direct.jrpc.jrpc_work_order.JRPCWorkOrderImpl . . . . .	92
avalon_sdk.connector.interfaces.work_order_proxy.WorkOrderProxy . . . . .	166
avalon_sdk.connector.interfaces.work_order_receipt.WorkOrderReceipt . . . . .	168
avalon_sdk.connector.interfaces.worker_registry.WorkerRegistry . . . . .	139
avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList . . . . .	144
Enum	
avalon_sdk.connector.blockchains.common.contract_response.ContractResponse . . . . .	32
avalon_sdk.registry.registry_status.RegistryStatus . . . . .	118
avalon_sdk.work_order_receipt.work_order_receipt.ReceiptCreateStatus . . . . .	116
avalon_sdk.worker.worker_details.WorkerStatus . . . . .	148
avalon_sdk.worker.worker_details.WorkerType . . . . .	149
IntEnum	
avalon_sdk.connector.direct.jrpc.jrpc_util.JsonRpcErrorCode . . . . .	103
WorkerRegistry	
avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.EthereumWorkerRegistryImpl . . . . .	38
avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorkerRegistryImpl . . . . .	60
avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl . . . . .	88
WorkerRegistryList	
avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry_list.EthereumWorkerRegistryListImpl . . . . .	43
avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistryListImpl . . . . .	65
WorkOrderProxy	
avalon_sdk.connector.blockchains.ethereum.ethereum_work_order.EthereumWorkOrderProxyImpl . . . . .	47
avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl . . . . .	69
WorkOrderReceipt	
avalon_sdk.connector.blockchains.fabric.fabric_work_order_receipt.FabricWorkOrderReceiptImpl . . . . .	74
avalon_sdk.connector.direct.jrpc.jrpc_work_order_receipt.JRPCWorkOrderReceiptImpl . . . . .	95

## Chapter 9

# Class Index

### 9.1 Class List

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

<a href="#">_sgx_errlist_t</a>	25
<a href="#">avalon_sdk.connector.direct.avalon_direct_client.AvalonDirectClient</a>	25
<a href="#">avalon_sdk.connector.blockchains.ethereum.ethereum_listener.BlockchainInterface</a>	27
<a href="#">avalon_sdk.connector.blockchains.fabric.base.ClientBase</a>	28
<a href="#">avalon_crypto_utils.signature.ClientSignature</a>	29
<a href="#">avalon_sdk.connector.blockchains.common.contract_response.ContractResponse</a>	32
<a href="#">tcf::error::CryptoError</a>	33
<a href="#">tcf::error::DivisionByZero</a>	34
<a href="#">avalon_crypto_utils.keys.EnclaveKeys</a>	35
<a href="#">tcf::error::Error</a>	37
<a href="#">avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.EthereumWorkerRegistryImpl</a>	38
<a href="#">avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry_list.EthereumWorkerRegistry↔ ListImpl</a>	43
<a href="#">avalon_sdk.connector.blockchains.ethereum.ethereum_work_order.EthereumWorkOrderProxyImpl</a>	47
<a href="#">avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper</a>	51
<a href="#">avalon_sdk.connector.blockchains.fabric.event_listener.EventListener</a>	55
<a href="#">avalon_sdk.connector.blockchains.ethereum.ethereum_listener.EventProcessor</a>	57
<a href="#">avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorkerRegistryImpl</a>	60
<a href="#">avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistryListImpl</a>	65
<a href="#">avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl</a>	69
<a href="#">avalon_sdk.connector.blockchains.fabric.fabric_work_order_receipt.FabricWorkOrderReceiptImpl</a>	74
<a href="#">avalon_sdk.connector.blockchains.fabric.fabric_wrapper.FabricWrapper</a>	79
<a href="#">FileIoExecutor</a>	80
<a href="#">avalon_sdk.http_client.http_rpc_client.HttpRpcClient</a>	85
<a href="#">tcf::error::IndexError</a>	86
<a href="#">tcf::error::IOError</a>	87
<a href="#">avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl</a>	88
<a href="#">avalon_sdk.connector.direct.jrpc.jrpc_work_order.JRPCWorkOrderImpl</a>	92
<a href="#">avalon_sdk.connector.direct.jrpc.jrpc_work_order_receipt.JRPCWorkOrderReceiptImpl</a>	95
<a href="#">json_array_t</a>	100
<a href="#">json_object_t</a>	101
<a href="#">json_value_t</a>	102
<a href="#">json_value_value</a>	103
<a href="#">avalon_sdk.connector.direct.jrpc.jrpc_util.JsonRpcErrorCode</a>	103
<a href="#">JsonValue</a>	105

tcf::error::MemoryError	106
avalon_sdk.http_client.http_jrpc_client.MessageException	107
tcf::error::OverflowError	108
tcf::crypto::pkenc::PrivateKey	109
tcf::crypto::sig::PrivateKey	111
tcf::crypto::sig::PublicKey	112
tcf::crypto::pkenc::PublicKey	114
avalon_sdk.work_order_receipt.work_order_receipt.ReceiptCreateStatus	116
avalon_sdk.registry.registry_status.RegistryStatus	118
tcf::error::RuntimeError	119
avalon_sdk.worker.worker_details.SGXWorkerDetails	120
StringArray	121
tcf::error::SystemBusyError	122
tcf::error::SystemError	123
test_ethereum_worker_registry_impl.TestEthereumWorkerRegistryImpl	124
test_ethereum_worker_registry_list_impl.TestEthereumWorkerRegistryListImpl	125
test_ethereum_work_order_impl.TestEthereumWorkOrderProxyImpl	126
test_fabric_worker_registry_impl.TestFabricWorkerRegistryImpl	129
test_fabric_worker_registry_list_impl.TestFabricWorkerRegistryListImpl	130
test_worker_registry_jrpc_impl.TestWorkerRegistryJRPCImpl	131
test_work_order_encryption_key_jrpc_impl.TestWorkOrderEncryptionKeyJRPCImpl	132
test_work_order_jrpc_impl.TestWorkOrderJRPCImpl	133
tcf::utility::Timer	133
avalon_sdk.connector.blockchains.fabric.tx_committer.TxCommitter	134
tcf::error::UnknownError	136
tcf::error::ValueError	137
avalon_sdk.worker.worker_details.WorkerDetails	138
avalon_sdk.connector.interfaces.worker_registry.WorkerRegistry	139
avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList	144
avalon_sdk.worker.worker_details.WorkerStatus	148
avalon_sdk.worker.worker_details.WorkerType	149
tcf::error::WorkloadError	150
WorkloadProcessor	151
avalon_sdk.connector.interfaces.work_order.WorkOrder	153
tcf::WorkOrderData	158
avalon_sdk.work_order.work_order_params.WorkOrderParams	158
avalon_sdk.connector.interfaces.work_order_proxy.WorkOrderProxy	166
avalon_sdk.connector.interfaces.work_order_receipt.WorkOrderReceipt	168
avalon_sdk.work_order_receipt.work_order_receipt.WorkOrderReceiptRequest	174
avalon_sdk.work_order.work_order_request_validator.WorkOrderRequestValidator	175



## Chapter 10

# File Index

### 10.1 File List

Here is a list of all documented files with brief descriptions:

/home/dano/git/avalon/common/cpp/c11_support.h	??
/home/dano/git/avalon/common/cpp/error.h	193
/home/dano/git/avalon/common/cpp/hex_string.h	194
/home/dano/git/avalon/common/cpp/json_utils.h	196
/home/dano/git/avalon/common/cpp/jsonvalue.h	196
/home/dano/git/avalon/common/cpp/tcf_error.h	200
/home/dano/git/avalon/common/cpp/timer.h	201
/home/dano/git/avalon/common/cpp/types.cpp	202
/home/dano/git/avalon/common/cpp/types.h	204
/home/dano/git/avalon/common/cpp/utils.cpp	207
/home/dano/git/avalon/common/cpp/utils.h	208
/home/dano/git/avalon/common/cpp/zero.h	210
/home/dano/git/avalon/common/cpp/crypto/crypto.h	??
/home/dano/git/avalon/common/cpp/crypto/crypto_shared.h	??
/home/dano/git/avalon/common/cpp/crypto/crypto_utils.h	177
/home/dano/git/avalon/common/cpp/crypto/pkenc.h	??
/home/dano/git/avalon/common/cpp/crypto/pkenc_private_key.cpp	179
/home/dano/git/avalon/common/cpp/crypto/pkenc_private_key.h	180
/home/dano/git/avalon/common/cpp/crypto/pkenc_public_key.cpp	181
/home/dano/git/avalon/common/cpp/crypto/pkenc_public_key.h	182
/home/dano/git/avalon/common/cpp/crypto/sig.h	??
/home/dano/git/avalon/common/cpp/crypto/sig_private_key.cpp	186
/home/dano/git/avalon/common/cpp/crypto/sig_private_key.h	188
/home/dano/git/avalon/common/cpp/crypto/sig_public_key.cpp	189
/home/dano/git/avalon/common/cpp/crypto/sig_public_key.h	190
/home/dano/git/avalon/common/cpp/crypto/skenc.cpp	191
/home/dano/git/avalon/common/cpp/crypto/skenc.h	192
/home/dano/git/avalon/common/cpp/crypto/verify_certificate.h	184
/home/dano/git/avalon/common/cpp/crypto/verify_signature.h	185
/home/dano/git/avalon/common/cpp/crypto/SAVE/verify_certificate.h	183
/home/dano/git/avalon/common/cpp/crypto/SAVE/verify_signature.h	184
/home/dano/git/avalon/common/cpp/packages/base64/base64.cpp	197
/home/dano/git/avalon/common/cpp/packages/base64/base64.h	198
/home/dano/git/avalon/common/cpp/packages/parson/parson.h	??
/home/dano/git/avalon/common/cpp/verify_ias_report/ias-certificates.h	??

/home/dano/git/avalon/common/cpp/verify_ias_report/ <b>ias_attestation_util.h</b> . . . . .	??
/home/dano/git/avalon/common/cpp/verify_ias_report/ <b>verify-report.h</b> . . . . .	??
/home/dano/git/avalon/common/sgx_workload/iohandler/ <a href="#">file_io.cpp</a> . . . . .	210
/home/dano/git/avalon/common/sgx_workload/iohandler/ <a href="#">file_io.h</a> . . . . .	211
/home/dano/git/avalon/common/sgx_workload/iohandler/ <a href="#">file_io_wrapper.cpp</a> . . . . .	212
/home/dano/git/avalon/common/sgx_workload/iohandler/ <a href="#">file_io_wrapper.h</a> . . . . .	214
/home/dano/git/avalon/common/sgx_workload/workload/ <a href="#">work_order_data.h</a> . . . . .	217
/home/dano/git/avalon/common/sgx_workload/workload/ <a href="#">workload_processor.cpp</a> . . . . .	218
/home/dano/git/avalon/common/sgx_workload/workload/ <a href="#">workload_processor.h</a> . . . . .	219
/home/dano/git/avalon/tc/sgx/trusted_worker_manager/tests/ <b>testCrypto.h</b> . . . . .	??
/home/dano/git/avalon/tc/sgx/trusted_worker_manager/tests/trusted/app/ <b>TestApp.h</b> . . . . .	??
/home/dano/git/avalon/tc/sgx/trusted_worker_manager/tests/trusted/enclave/ <b>TestEnclave.h</b> . . . . .	??

# Chapter 11

## Namespace Documentation

### 11.1 tcf::crypto::constants Namespace Reference

#### Variables

- const int **ERR\_BUF\_LEN** = 130
- const int **RSA\_KEY\_SIZE** = 2048
- const int **RSA\_PADDING\_SIZE** = 41
- const int **RSA\_PADDING\_SCHEME** = RSA\_PKCS1\_OAEP\_PADDING
- constexpr int **RSA\_PLAINTEXT\_LEN**
- const int **CURVE** = NID\_secp256k1
- const int **MAX\_SIG\_SIZE** = 72
- const int **IV\_LEN** = 12
- const int **SYM\_KEY\_LEN** = 32
- const int **TAG\_LEN** = 16

#### 11.1.1 Detailed Description

AES-GCM 256 for authenticated encryption.

#### 11.1.2 Variable Documentation

##### 11.1.2.1 IV\_LEN

```
const int tcf::crypto::constants::IV_LEN = 12
```

AES-GCM IV length (96 bits)

### 11.1.2.2 RSA\_PLAINTEXT\_LEN

```
constexpr int tcf::crypto::constants::RSA_PLAINTEXT_LEN
```

#### Initial value:

```
=
    ((RSA_KEY_SIZE - RSA_PADDING_SIZE) >> 3)
```

### 11.1.2.3 SYM\_KEY\_LEN

```
const int tcf::crypto::constants::SYM_KEY_LEN = 32
```

AES-GCM Key length (256 bits)

### 11.1.2.4 TAG\_LEN

```
const int tcf::crypto::constants::TAG_LEN = 16
```

AES-GCM TAG length (128 bits)

## 11.2 tcf::crypto::skenc Namespace Reference

### Functions

- [ByteArray GenerateKey](#) ()
- [ByteArray GenerateIV](#) (const std::string &IVstring=std::string(""))
- [ByteArray EncryptMessage](#) (const [ByteArray](#) &key, const [ByteArray](#) &iv, const [ByteArray](#) &message)
- [ByteArray EncryptMessage](#) (const [ByteArray](#) &key, const [ByteArray](#) &message)
- [ByteArray DecryptMessage](#) (const [ByteArray](#) &key, const [ByteArray](#) &iv, const [ByteArray](#) &message)
- [ByteArray DecryptMessage](#) (const [ByteArray](#) &key, const [ByteArray](#) &message)

### 11.2.1 Detailed Description

Authenticated encryption.

### 11.2.2 Function Documentation

### 11.2.2.1 DecryptMessage() [1/2]

```
ByteArray tcf::crypto::skenc::DecryptMessage (
    const ByteArray & key,
    const ByteArray & iv,
    const ByteArray & message )
```

Throws RuntimeError, ValueError, CryptoError (message authentication failure).

### 11.2.2.2 DecryptMessage() [2/2]

```
ByteArray tcf::crypto::skenc::DecryptMessage (
    const ByteArray & key,
    const ByteArray & message )
```

Throws RuntimeError, ValueError, CryptoError (message authentication failure). Expects IV prepended to message ciphertext.

### 11.2.2.3 EncryptMessage() [1/2]

```
ByteArray tcf::crypto::skenc::EncryptMessage (
    const ByteArray & key,
    const ByteArray & iv,
    const ByteArray & message )
```

Throws RuntimeError, ValueError.

### 11.2.2.4 EncryptMessage() [2/2]

```
ByteArray tcf::crypto::skenc::EncryptMessage (
    const ByteArray & key,
    const ByteArray & message )
```

Uses random IV prepended the returned ciphertext. Throws RuntimeError, ValueError.

### 11.2.2.5 GenerateIV()

```
ByteArray tcf::crypto::skenc::GenerateIV (
    const std::string & IVstring = std::string("") )
```

Throws RuntimeError.

### 11.2.2.6 GenerateKey()

```
ByteArray tcf::crypto::skenc::GenerateKey ( )
```

ByteArray here is used to encapsulate raw binary data and does not apply/assume any encoding. Throws RuntimeError.



## Chapter 12

# Class Documentation

### 12.1 `_sgx_errlist_t` Struct Reference

#### Public Attributes

- `sgx_status_t err`
- `const char * msg`
- `const char * sug`

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

- `/home/dano/git/avalon/tc/sgx/trusted_worker_manager/tests/trusted/app/TestApp.cpp`

### 12.2 `avalon_sdk.connector.direct.avalon_direct_client.AvalonDirectClient` Class Reference

#### Public Member Functions

- `def __init__ (self, config_file=None, config=None)`
- `def get\_worker\_registry\_list\_instance (self)`
- `def get\_worker\_registry\_instance (self)`
- `def get\_work\_order\_instance (self)`
- `def get\_work\_order\_receipt\_instance (self)`

#### 12.2.1 Detailed Description

This is class for the direct JSON RPC API client.  
It is used in the direct model.

1. Worker registry list interacts with the blockchain; it is optional.
2. Worker registry interacts with the JSON RPC listener.
3. Work order interacts with the JSON RPC listener.
4. Work order receipt interacts with the JSON RPC listener.

## 12.2.2 Constructor & Destructor Documentation

### 12.2.2.1 `__init__()`

```
def avalon_sdk.connector.direct.avalon_direct_client.AvalonDirectClient.__init__ (
    self,
    config_file = None,
    config = None )
```

Parameters:

`config_file` Optional configuration file path as a string  
`config` Optional dictionary loaded from `config_file`

Either one of `config_file` or `config` needs to be passed.  
If both are passed, then `config` takes precedence.

## 12.2.3 Member Function Documentation

### 12.2.3.1 `get_work_order_instance()`

```
def avalon_sdk.connector.direct.avalon_direct_client.AvalonDirectClient.get_work_order_↵
instance (
    self )
```

Return the worker's instance of the work order.  
The work order interacts with the JSON RPC listener.

### 12.2.3.2 `get_work_order_receipt_instance()`

```
def avalon_sdk.connector.direct.avalon_direct_client.AvalonDirectClient.get_work_order_↵
receipt_instance (
    self )
```

Return the worker's instance of the work order receipt.  
The work order receipt interacts with the JSON RPC listener.



## 12.2.3.3 get\_worker\_registry\_instance()

```
def avalon_sdk.connector.direct.avalon_direct_client.AvalonDirectClient.get_worker_registry_↵
instance (
    self )
```

Return the worker's instance of the registry.  
The registry interacts with the JSON RPC listener.

## 12.2.3.4 get\_worker\_registry\_list\_instance()

```
def avalon_sdk.connector.direct.avalon_direct_client.AvalonDirectClient.get_worker_registry_↵
list_instance (
    self )
```

Return the worker's instance of the registry list.  
This list is optional and interacts with the blockchain.

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

- /home/dano/git/avalon/sdk/avalon\_sdk/connector/direct/avalon\_direct\_client.py

## 12.3 avalon\_sdk.connector.blockchains.ethereum.ethereum\_listener.BlockchainInterface Class Reference

### Public Member Functions

- def **\_\_init\_\_** (self, config)
- def [newListener](#) (self, contract, event, fromBlock='latest')

### 12.3.1 Detailed Description

Ethereum blockchain interface to event processor.

### 12.3.2 Member Function Documentation

### 12.3.2.1 newListener()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_listener.BlockchainInterface.new↔
Listener (
    self,
    contract,
    event,
    fromBlock = 'latest' )
```

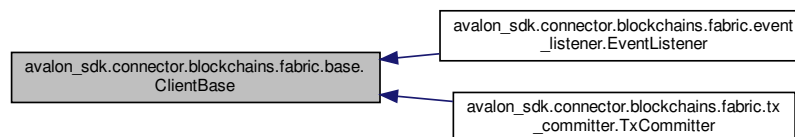
Create a filter to get events from latest block by default.

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

- /home/dano/git/avalon/sdk/avalon\_sdk/connector/blockchains/ethereum/ethereum\_listener.py

## 12.4 avalon\_sdk.connector.blockchains.fabric.base.ClientBase Class Reference

Inheritance diagram for avalon\_sdk.connector.blockchains.fabric.base.ClientBase:



### Public Member Functions

- `def __init__ (self, profile, channel_name, org_name, peer_name, user_name)`
- `def channel_name (self)`
- `def channel (self)`
- `def org_name (self)`
- `def peer_name (self)`
- `def user_name (self)`
- `def user (self)`

### Public Attributes

- `client`

### 12.4.1 Detailed Description

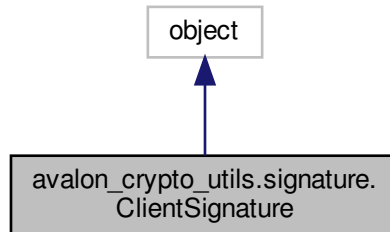
Base class for a Hyperledger Fabric client.

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

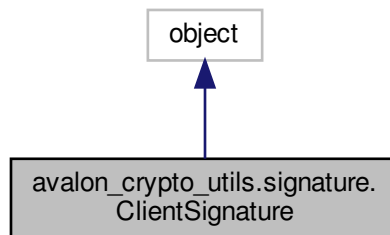
- /home/dano/git/avalon/sdk/avalon\_sdk/connector/blockchains/fabric/base.py

## 12.5 avalon\_crypto\_utils.signature.ClientSignature Class Reference

Inheritance diagram for avalon\_crypto\_utils.signature.ClientSignature:



Collaboration diagram for avalon\_crypto\_utils.signature.ClientSignature:



### Public Member Functions

- `def __init__(self)`
- `def calculate_datahash(self, data_objects)`
- `def generate_signature(self, hash, private_key)`
- `def generate_client_signature(self, input_json_str, worker, private_key, session_key, session_iv, encrypted_data, _session_key, data_key=None, data_iv=None)`
- `def verify_signature(self, input_json, verification_key)`
- `def verify_update_receipt_signature(self, input_json)`
- `def verify_create_receipt_signature(self, input_json)`
- `def calculate_request_hash(self, input_json)`

### Public Attributes

- `private_key`
- `public_key`
- `param_pool`
- `tcs_worker`

### 12.5.1 Detailed Description

Class to perform hash calculation, signature generation and verification

### 12.5.2 Member Function Documentation

#### 12.5.2.1 calculate\_datahash()

```
def avalon_crypto_utils.signature.ClientSignature.calculate_datahash (
    self,
    data_objects )
```

Function to calculate a hash value of the array concatenating dataHash, data, encryptedDataEncryptionKey, iv for each item in the inData/outData array

Parameters:

- data\_objects is each item in inData or outData part of workorder request as per Trusted Compute EEA API 6.1.7 Work Order Data Formats

#### 12.5.2.2 calculate\_request\_hash()

```
def avalon_crypto_utils.signature.ClientSignature.calculate_request_hash (
    self,
    input_json )
```

Function to create the work order request hash as defined in EEA spec 6.1.8.1

Parameters:

- input\_json is dictionary contains work order request payload as define EEA spec 6.1.1

Returns hash of work order request as string

**12.5.2.3 generate\_client\_signature()**

```
def avalon_crypto_utils.signature.ClientSignature.generate_client_signature (
    self,
    input_json_str,
    worker,
    private_key,
    session_key,
    session_iv,
    encrypted_session_key,
    data_key = None,
    data_iv = None )
```

Function to generate client signature

Parameters:

- input\_json\_str is requester Work Order Request payload in a JSON-RPC based format defined 6.1.1 Work Order Request Payload
- worker is a worker object to store all the common details of worker as per Trusted Compute EEA API 8.1 Common Data for All Worker Types
- private\_key is Client private key
- session\_key is one time session key generated by the participant submitting the work order.
- session\_iv is an initialization vector if required by the data encryption algorithm (encryptedSessionKey). The default is all zeros.
- data\_key is a one time key generated by participant used to encrypt work order indata
- data\_iv is an initialization vector used along with data\_key. Default is all zeros.
- encrypted\_session\_key is a encrypted version of session\_key.

Returns a tuple containing signature and status

**12.5.2.4 generate\_signature()**

```
def avalon_crypto_utils.signature.ClientSignature.generate_signature (
    self,
    hash,
    private_key )
```

Function to generate signature object

Parameters:

- hash is the combined array of all hashes calculated on the message
- private\_key is Client private key

Returns tuple(status, signature)

**12.5.2.5 verify\_create\_receipt\_signature()**

```
def avalon_crypto_utils.signature.ClientSignature.verify_create_receipt_signature (
    self,
    input_json )
```

Function to verify the signature of work order receipt create

Parameters:

- input\_json is dictionary contains request payload of WorkOrderReceiptRetrieve API as define EEA spec 7.2.2

Returns enum type SignatureStatus

### 12.5.2.6 verify\_signature()

```
def avalon_crypto_utils.signature.ClientSignature.verify_signature (
    self,
    input_json,
    verification_key )
```

Function to verify the signature received from the enclave

Parameters:

- input\_json is dictionary contains payload returned by the Worker Service in response to successful workorder submit request as per Trusted Compute EEA API 6.1.2 Work Order Result Payload
- verification\_key is ECDSA/SECP256K1 public key used to verify signatures created by the Enclave.

Returns enum type SignatureStatus

### 12.5.2.7 verify\_update\_receipt\_signature()

```
def avalon_crypto_utils.signature.ClientSignature.verify_update_receipt_signature (
    self,
    input_json )
```

Function to verify the signature of work order receipt update

Parameters:

- input\_json is dictionary contains payload returned by the WorkOrderReceiptUpdateRetrieve API as define EEA spec 7.2.7

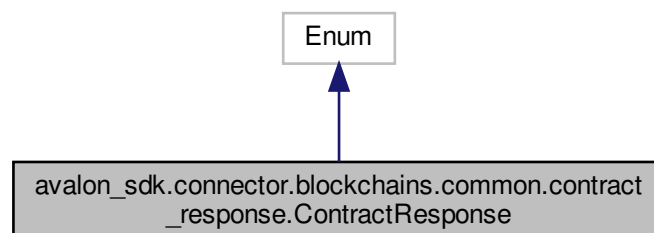
Returns enum type SignatureStatus

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

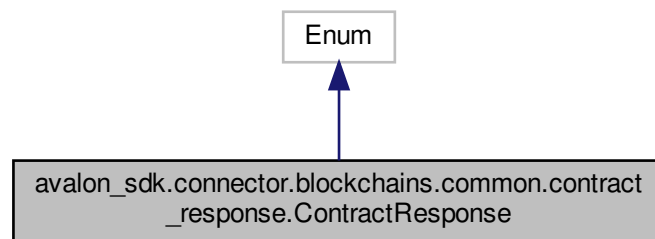
- /home/dano/git/avalon/common/crypto\_utils/avalon\_crypto\_utils/signature.py

## 12.6 avalon\_sdk.connector.blockchains.common.contract\_response.ContractResponse Class Reference

Inheritance diagram for avalon\_sdk.connector.blockchains.common.contract\_response.ContractResponse:



Collaboration diagram for avalon\_sdk.connector.blockchains.common.contract\_response.ContractResponse:



### Static Public Attributes

- int **SUCCESS** = 0
- int **ERROR** = -1

### 12.6.1 Detailed Description

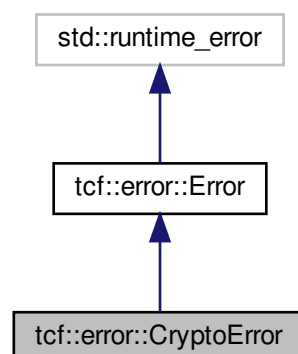
Contract response values.

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

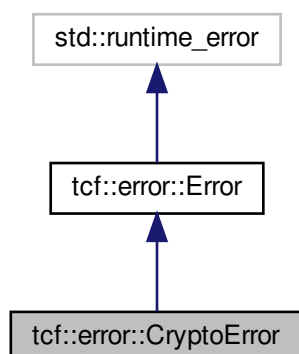
- /home/dano/git/avalon/sdk/avalon\_sdk/connector/blockchains/common/contract\_response.py

## 12.7 tcf::error::CryptoError Class Reference

Inheritance diagram for tcf::error::CryptoError:



Collaboration diagram for `tcf::error::CryptoError`:



### Public Member Functions

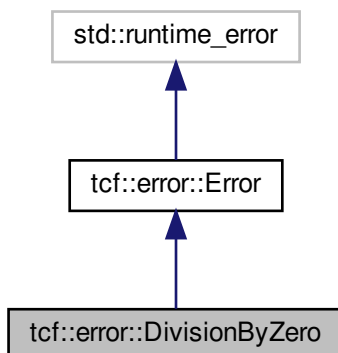
- **CryptoError** (const std::string &msg)

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

- </home/dano/git/avalon/common/cpp/error.h>

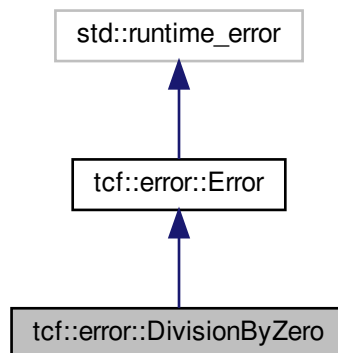
## 12.8 tcf::error::DivisionByZero Class Reference

Inheritance diagram for `tcf::error::DivisionByZero`:





Collaboration diagram for tcf::error::DivisionByZero:



### Public Member Functions

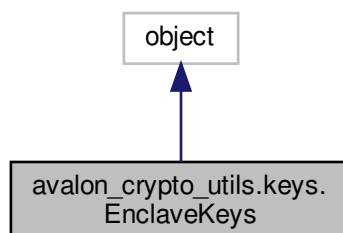
- **DivisionByZero** (const std::string &msg)

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

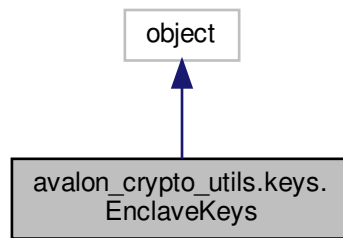
- `/home/dano/git/avalon/common/cpp/error.h`

## 12.9 avalon\_crypto\_utils.keys.EnclaveKeys Class Reference

Inheritance diagram for avalon\_crypto\_utils.keys.EnclaveKeys:



Collaboration diagram for `avalon_crypto_utils.keys.EnclaveKeys`:



## Public Member Functions

- `def \_\_init\_\_ (self, verifying_key, encryption_key)`
- `def identity (self)`
- `def hashed_identity (self)`

### 12.9.1 Detailed Description

Wrapper for managing the enclave's keys, the `verifying_key` is an ECDSA public key used to verify enclave signatures, the `encryption_key` is an RSA public key for encrypting message to the enclave.

### 12.9.2 Constructor & Destructor Documentation

#### 12.9.2.1 `__init__()`

```
def avalon_crypto_utils.keys.EnclaveKeys.__init__ (
    self,
    verifying_key,
    encryption_key )
```

Initialize the object

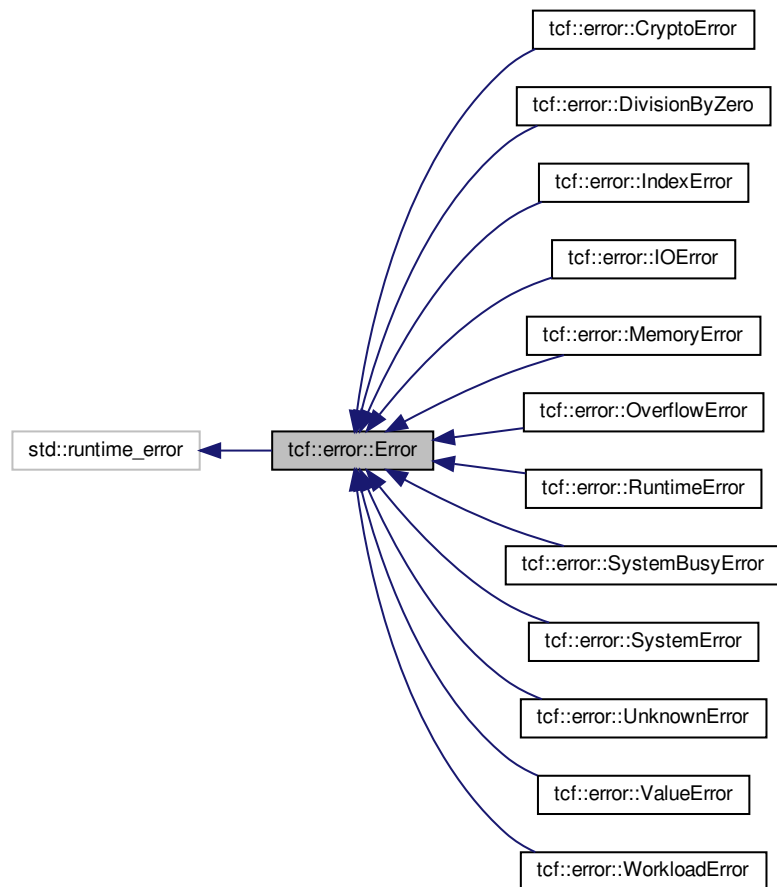
```
:param verifying_key: PEM encoded ECDSA verifying key
:param encryption_key: PEM encoded RSA encryption key
```

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

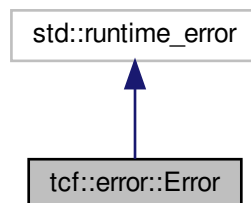
- `/home/dano/git/avalon/common/crypto_utils/avalon_crypto_utils/keys.py`

## 12.10 tcf::error::Error Class Reference

Inheritance diagram for tcf::error::Error:



Collaboration diagram for tcf::error::Error:



## Public Member Functions

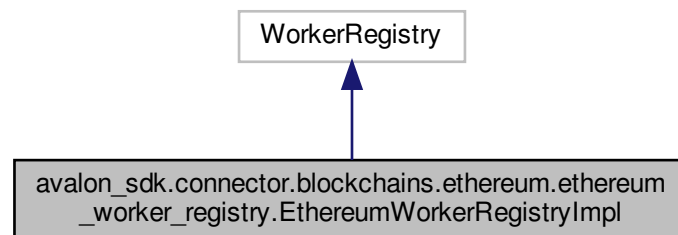
- **Error** ([tcf\\_err\\_t](#) in\_error, const std::string &msg)
- [tcf\\_err\\_t](#) **error\_code** ()

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

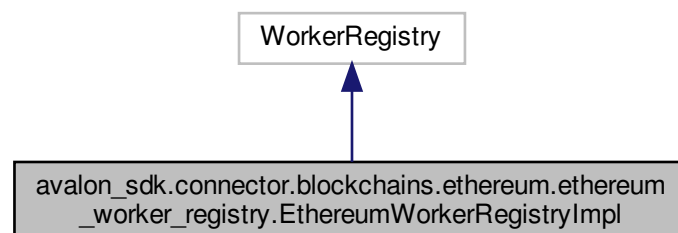
- /home/dano/git/avalon/common/cpp/[error.h](#)

## 12.11 `avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.EthereumWorkerRegistryImpl` Class Reference

Inheritance diagram for `avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.EthereumWorkerRegistryImpl`:



Collaboration diagram for `avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.EthereumWorkerRegistryImpl`:



**Public Member Functions**

---

- def `__init__` (self, config)
- def `worker_lookup` (self, worker\_type, org\_id, application\_id, id=None)
- def `worker_retrieve` (self, worker\_id, id=None)
- def `worker_lookup_next` (self, worker\_type, org\_id, application\_id, lookup\_tag)
- def `worker_register` (self, worker\_id, worker\_type, organization\_id, application\_type\_ids, details)
- def `worker_update` (self, worker\_id, details)
- def `worker_set_status` (self, worker\_id, status)

**12.11.1 Detailed Description**

This class is sets and gets worker-related information to and from the Ethereum blockchain.

Detailed method descriptions are available in the WorkerRegistry interfaces.

**12.11.2 Constructor & Destructor Documentation****12.11.2.1 `__init__()`**

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.EthereumWorkerRegistry←
Impl.__init__ (
    self,
    config )
```

Parameters:

config     Dictionary containing Ethereum-specific parameters

**12.11.3 Member Function Documentation****12.11.3.1 `worker_lookup()`**

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.EthereumWorkerRegistry←
Impl.worker_lookup (
    self,
    worker_type,
    org_id,
    application_id,
    id = None )
```

Lookup a worker identified by `worker_type`, `org_id`, and `application_id`. All fields are optional and, if present, condition should match for all fields. If none are passed it should return all workers.

If the list is too large to fit into a single response (the maximum number of entries in a single response is implementation specific), the smart contract should return the first batch of the results and provide a `lookupTag` that can be used by the caller to retrieve the next batch by calling `worker_lookup_next`.

Parameters:

`worker_type` Optional characteristic of workers for which you may wish to search  
`org_id` Optional organization ID that can be used to search for one or more workers that belong to this organization  
`application_id` Optional application type ID that is supported by the worker  
`id` Optional JSON RPC request ID

Returns:

Tuple containing workers count, lookup tag, and list of worker IDs:

`total_count` Total number of entries matching a specified lookup criteria. If this number is larger than the size of the IDs array, the caller should use `lookupTag` to call `worker_lookup_next` to retrieve the rest of the IDs

`lookup_tag` Optional parameter. If it is returned, it means that there are more matching worker IDs, which can then be retrieved by calling function `worker_lookup_next` with this tag as an input parameter

`ids` Array of the worker IDs that match the input parameters

On error returns None.

### 12.11.3.2 worker\_lookup\_next()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.EthereumWorkerRegistry←
Impl.worker_lookup_next (
    self,
    worker_type,
    org_id,
    application_id,
    lookup_tag )
```

Retrieve additional worker lookup results after calling `worker_lookup`.

Parameters:

`worker_type` Characteristic of Workers for which you may wish to search  
`org_id` Organization ID to which a Worker belongs  
`application_id` Optional application type ID that is supported by the worker  
`lookup_tag` is returned by a previous call to either this function or to `worker_lookup`  
`id` Optional Optional JSON RPC request ID

Returns:

Tuple containing the following:

`total_count` Total number of entries matching this lookup criteria. If this number is larger than the number of IDs returned so far, the caller should use

---

```

        lookupTag to call worker_lookup_next to retrieve
        the rest of the IDs
new_lookup_tag Optional parameter. If it is returned, it
        means that there are more matching worker IDs that
        can be retrieved by calling this function again with
        this tag as an input parameter
ids            Array of the worker IDs that match the input parameters

```

On error returns None.

### 12.11.3.3 worker\_register()

```

def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.EthereumWorkerRegistry←
Impl.worker_register (
    self,
    worker_id,
    worker_type,
    organization_id,
    application_type_ids,
    details )

```

Register a new worker with details of the worker.

Parameters:

```

worker_id      Worker ID value. E.g., an Ethereum address or
                a value derived from the worker's DID
worker_type    Type of Worker. Currently defined types are:
    * "TEE-SGX": an Intel SGX Trusted Execution
                Environment
    * "MPC": Multi-Party Compute
    * "ZK": Zero-Knowledge
organization_id Optional parameter representing the
                organization that hosts the Worker,
                e.g. a bank in the consortium or
                anonymous entity
application_ids Optional parameter that defines
                application types supported by the Worker
details        Detailed information about the worker in
                JSON RPC format as defined in
                https://entethalliance.github.io/trusted-computing/spec.html
                #common-data-for-all-worker-types

```

Returns:

```

Transaction receipt if registration succeeds.
None if registration does not succeed.

```

### 12.11.3.4 worker\_retrieve()

```

def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.EthereumWorkerRegistry←
Impl.worker_retrieve (
    self,
    worker_id,
    id = None )

```

Retrieve the worker identified by worker ID.

Parameters:

`worker_id` Worker ID of the registry whose details are requested  
`id` Optional JSON RPC request ID

Returns:

Tuple containing worker status (defined in `worker_set_status`),  
 worker type, organization ID, list of application IDs, and worker  
 details (JSON RPC string).

On error returns None.

### 12.11.3.5 `worker_set_status()`

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.EthereumWorkerRegistry↔
Impl.worker_set_status (
    self,
    worker_id,
    status )
```

Set the worker status identified by worker ID.

Parameters:

`worker_id` Worker ID value. E.g., an Ethereum address or  
 a value derived from the worker's DID  
`status` Worker status. The currently defined values are:  
 1 - worker is active  
 2 - worker is temporarily "off-line"  
 3 - worker is decommissioned  
 4 - worker is compromised

Returns:

Transaction receipt if registration succeeds.  
 None if registration does not succeed.

### 12.11.3.6 `worker_update()`

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry.EthereumWorkerRegistry↔
Impl.worker_update (
    self,
    worker_id,
    details )
```

Update a worker with details data.

Parameters:

`worker_id` Worker ID value. E.g., an Ethereum address or  
 a value derived from the worker's DID  
`details` Detailed information about the worker in JSON format

Returns:

Transaction receipt if registration succeeds.  
 None if registration does not succeed.

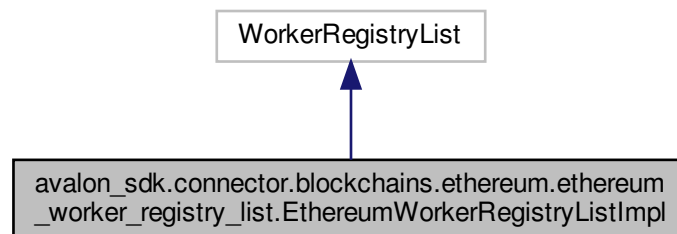
The documentation for this class was generated from the following file:

- `/home/dano/git/avalon/sdk/avalon_sdk/connector/blockchains/ethereum/ethereum_worker_registry.py`

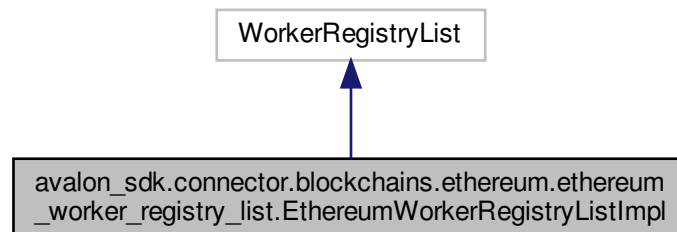


## 12.12 `avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry_list.EthereumWorkerRegistryListImpl` Class Reference

Inheritance diagram for `avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry_list.EthereumWorkerRegistryListImpl`:



Collaboration diagram for `avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry_list.EthereumWorkerRegistryListImpl`:



### Public Member Functions

- `def __init__ (self, config)`
- `def registry_lookup (self, app_type_id=None)`
- `def registry_retrieve (self, org_id)`
- `def registry_lookup_next (self, app_type_id, lookup_tag)`
- `def registry_add (self, org_id, uri, sc_addr, app_type_ids)`
- `def registry_update (self, org_id, uri, sc_addr, app_type_ids)`
- `def registry_set_status (self, org_id, status)`

### 12.12.1 Detailed Description

This class provide APIs to read/write registry entries of workers, which is stored in the Ethereum blockchain.

## 12.12.2 Member Function Documentation

### 12.12.2.1 registry\_add()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry_list.EthereumWorkerRegistryListImpl.registry_add (
    self,
    org_id,
    uri,
    sc_addr,
    app_type_ids )
```

Add a new registry.

Parameters:

`org_id` bytes[] identifies organization that hosts the registry, e.g. a bank in the consortium or an anonymous entity

`uri` String defines a URI for this registry that supports the Off-Chain Worker Registry JSON RPC API.

`sc_addr` bytes[] defines an Ethereum address that runs the Worker Registry Smart Contract API smart contract for this registry

`app_type_ids` []bytes[] is an optional parameter that defines application types supported by the worker managed by the registry

Returns:

Transaction receipt on success or None on error.

### 12.12.2.2 registry\_lookup()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry_list.EthereumWorkerRegistryListImpl.registry_lookup (
    self,
    app_type_id = None )
```

Registry Lookup identified by application type ID.

Parameters:

`app_type_id` Application type ID to lookup in the registry

Returns:

Returns tuple containing `totalCount`, `lookupTag`, `ids` on success:

`totalCount` Total number of entries matching a specified lookup criteria. If this number is larger than the size of the IDs array, the caller should use the `lookupTag` to call `workerLookupNext` to retrieve the rest of the IDs

`lookupTag` Optional parameter. If it is returned, it means that there are more matching registry IDs that can be retrieved by calling the function `registry_lookup_next` with this tag as an input parameter

`ids` Array of the registry organization IDs that match the input parameters

Returns None on error.

**12.12.2.3 registry\_lookup\_next()**

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry_list.EthereumWorkerRegistryListImpl.registry_lookup_next (
    self,
    app_type_id,
    lookup_tag )
```

Get additional registry lookup results.  
This function is called to retrieve additional results of the Registry lookup initiated by the registry\_lookup call.

**Parameters:**

app\_type\_id     Application type that has to be supported by the workers retrieved  
lookup\_tag     Returned by a previous call to either this function or to registry\_lookup

**Returns:**

Outputs tuple on success containing the following:

total\_count     Total number of entries matching the lookup criteria. If this number is larger than the number of IDs returned so far, the caller should use lookup\_tag to call registry\_lookup\_next to the rest of the ids  
new\_lookup\_tag   Optional parameter. If it is returned, it means that there are more matching registry IDs that can be retrieved by calling this function again with this tag as an input parameter  
ids             Array of the registry IDs that match the input parameters

Returns None on error.

**12.12.2.4 registry\_retrieve()**

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry_list.EthereumWorkerRegistryListImpl.registry_retrieve (
    self,
    org_id )
```

Retrieving Registry Information identified by organization ID.

**Parameters:**

org\_id     Organization ID to lookup

**Returns:**

Tuple containing following on success:

uri             string defining a URI for this registry that supports the Off-Chain Worker Registry JSON RPC API. It will be None for the proxy model  
sc\_addr         Ethereum address for worker registry smart contract address  
application\_type\_ids   List of application ids(array of byte[])  
status             Status of the registry

Returns None on error.

### 12.12.2.5 registry\_set\_status()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry_list.EthereumWorkerRegistryListImpl.registry_set_status (
    self,
    org_id,
    status )
```

Set registry status.

**Parameters:**

org\_id bytes[] identifies organization that hosts the registry  
 status Defines registry status to set.  
 The currently defined values are:  
 1 - the registry is active  
 2 - the registry is temporarily "off-line"  
 3 - the registry is decommissioned

**Returns:**

Transaction receipt on success or None on error.

### 12.12.2.6 registry\_update()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_worker_registry_list.EthereumWorkerRegistryListImpl.registry_update (
    self,
    org_id,
    uri,
    sc_addr,
    app_type_ids )
```

Update a registry.

**Parameters:**

org\_id bytes[] identifies organization that hosts the registry, e.g. a bank in the consortium or an anonymous entity  
 uri string defines a URI for this registry that supports the Off-Chain Worker Registry JSON RPC API  
 sc\_addr bytes[] defines an Ethereum address that runs a Worker Registry Smart Contract API smart contract for this registry  
 app\_type\_ids []bytes[] is an optional parameter that defines application types supported by the worker managed by the registry

**Returns:**

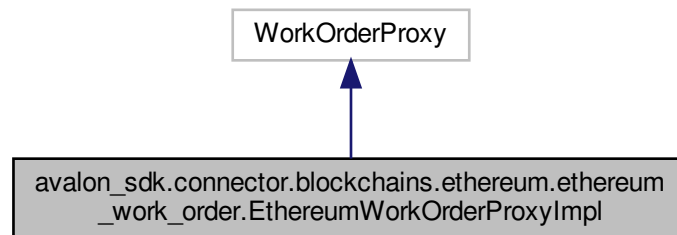
Transaction receipt on success or None on error.

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

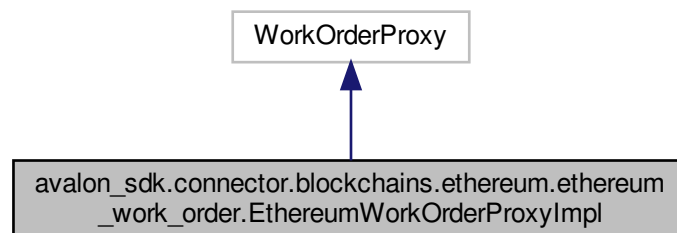
- /home/dano/git/avalon/sdk/avalon\_sdk/connector/blockchains/ethereum/ethereum\_worker\_registry\_list.py

## 12.13 avalon\_sdk.connector.blockchains.ethereum.ethereum\_work\_order.EthereumWorkOrderProxyImpl Class Reference

Inheritance diagram for avalon\_sdk.connector.blockchains.ethereum.ethereum\_work\_order.EthereumWorkOrderProxyImpl:



Collaboration diagram for avalon\_sdk.connector.blockchains.ethereum.ethereum\_work\_order.EthereumWorkOrderProxyImpl:



### Public Member Functions

- def **\_\_init\_\_** (self, config)
- def **work\_order\_submit** (self, work\_order\_id, worker\_id, requester\_id, work\_order\_request, id=None)
- def **work\_order\_complete** (self, work\_order\_id, work\_order\_response)
- def **work\_order\_get\_result** (self, work\_order\_id, id=None)
- def **encryption\_key\_retrieve** (self, worker\_id, last\_used\_key\_nonce, tag, requester\_id, signature\_nonce=None, signature=None, id=None)
- def **encryption\_key\_start** (self, tag, id=None)
- def **encryption\_key\_set** (self, worker\_id, encryption\_key, encryption\_nonce, tag, signature, id=None)
- def **encryption\_key\_get** (self, worker\_id, requester\_id, last\_used\_key\_nonce=None, tag=None, signature\_nonce=None, signature=None, id=None)

### 12.13.1 Detailed Description

This class is meant to write work order-related data to the Ethereum blockchain.  
Detailed method descriptions are available in the interfaces.

### 12.13.2 Member Function Documentation

#### 12.13.2.1 encryption\_key\_get()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_work_order.EthereumWorkOrderProxy↔
Impl.encryption_key_get (
    self,
    worker_id,
    requester_id,
    last_used_key_nonce = None,
    tag = None,
    signature_nonce = None,
    signature = None,
    id = None )
```

Get Encryption Key Request Payload.  
Not supported for Ethereum.

#### 12.13.2.2 encryption\_key\_retrieve()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_work_order.EthereumWorkOrderProxy↔
Impl.encryption_key_retrieve (
    self,
    worker_id,
    last_used_key_nonce,
    tag,
    requester_id,
    signature_nonce = None,
    signature = None,
    id = None )
```

Get Encryption Key Request Payload.  
Not supported for Ethereum.

### 12.13.2.3 encryption\_key\_set()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_work_order.EthereumWorkOrderProxy↔
Impl.encryption_key_set (
    self,
    worker_id,
    encryption_key,
    encryption_nonce,
    tag,
    signature,
    id = None )
```

Set Encryption Key Request Payload.  
Not supported for Ethereum.

### 12.13.2.4 encryption\_key\_start()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_work_order.EthereumWorkOrderProxy↔
Impl.encryption_key_start (
    self,
    tag,
    id = None )
```

Inform the Worker that it should start  
encryption key generation for this requester.  
Not supported for Ethereum.

### 12.13.2.5 work\_order\_complete()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_work_order.EthereumWorkOrderProxy↔
Impl.work_order_complete (
    self,
    work_order_id,
    work_order_response )
```

This function is called by the Worker Service to  
complete a work order successfully or in error.  
This API is for the proxy model.

Parameters:

work_order_id	Unique ID to identify the work order request
work_order_response	Work order response data in a string

Returns:

errorCode	0 on success or non-zero on error.
-----------	------------------------------------

**12.13.2.6 work\_order\_get\_result()**

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_work_order.EthereumWorkOrderProxy←
Impl.work_order_get_result (
    self,
    work_order_id,
    id = None )
```

Query blockchain to get a work order result.  
This function starts an event handler for handling the  
workOrderCompleted event from the Ethereum blockchain.

**Parameters:**

work\_order\_id Work Order ID that was sent in the  
corresponding work\_order\_submit request  
id Optional JSON RPC request ID

**Returns:**

Tuple containing work order status, worker id, work order request,  
work order response, and error code.  
None on error.

**12.13.2.7 work\_order\_submit()**

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_work_order.EthereumWorkOrderProxy←
Impl.work_order_submit (
    self,
    work_order_id,
    worker_id,
    requester_id,
    work_order_request,
    id = None )
```

Submit work order request to the Ethereum block chain.

**Parameters:**

work\_order\_id Unique ID of the work order request  
worker\_id Identifier for the worker  
requester\_id Unique id to identify the requester  
work\_order\_request JSON RPC string work order request.  
Complete definition at work\_order.py and  
defined in EEA specification 6.1.1.  
id Optional JSON RPC request ID

**Returns:**

0 on success and non-zero on error.

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

- /home/dano/git/avalon/sdk/avalon\_sdk/connector/blockchains/ethereum/ethereum\_work\_order.py



## 12.14 avalon\_sdk.connector.blockchains.ethereum.ethereum\_wrapper.EthereumWrapper Class Reference

### Public Member Functions

- def `__init__` (self, config)
- def `compile_source_file` (self, file\_path)
- def `deploy_contract` (self, contract\_interface)
- def `sign_execute_raw_transaction` (self, tx\_dict)
- def `execute_unsigned_transaction` (self, tx\_dict)
- def `execute_transaction` (self, tx\_dict)
- def `get_chain_id` (self)
- def `get_gas_limit` (self)
- def `get_gas_price` (self)
- def `get_account_address` (self)
- def `get_contract_instance` (self, contract\_file\_name, contract\_address)
- def `get_contract_instance_from_json` (self, json\_file\_name, contract\_address)
- def `get_txn_nonce` (self)
- def `get_transaction_params` (self)
- def `get_bytes_from_hex` (self, hex\_str)

### 12.14.1 Detailed Description

Ethereum wrapper class to interact with the Ethereum blockchain to deploy compile contract code, deploy contract code, and execute contract code.

### 12.14.2 Member Function Documentation

#### 12.14.2.1 `compile_source_file()`

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.compile_↵  
source_file (   
    self,   
    file_path )
```

Compile a Solidity contract file and returns contract instance object.

Parameters:

`file_path`     Path to Solidity contract file

Returns:

Solidity contract instance object.

#### 12.14.2.2 `deploy_contract()`

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.deploy_contract  
(  
    self,  
    contract_interface )
```

Deploys a Solidity contract to an Ethereum network identified by `chain_id`.

Parameters:  
`contract_interface` Solidity contract interface

Returns:  
Solidity contract address.

#### 12.14.2.3 `execute_transaction()`

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.execute_↵  
transaction (   
    self,  
    tx_dict )
```

Wrapper function to choose appropriate function to execute a transaction based on provider (Ropsten vs other).

Parameters:  
`tx_dict` Transaction to execute

#### 12.14.2.4 `execute_unsigned_transaction()`

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.execute_↵  
unsigned_transaction (   
    self,  
    tx_dict )
```

Send a transaction to be executed only with the account address, and wait for receipts.

Parameters:  
`tx_dict` Unsigned transaction to execute

Returns:  
Transaction receipt on success or None on error.

#### 12.14.2.5 get\_account\_address()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.get_account_↵
address (
    self )
```

Retrieve account address.

#### 12.14.2.6 get\_bytes\_from\_hex()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.get_bytes_↵
from_hex (
    self,
    hex_str )
```

Convert a hex string to bytes.

#### 12.14.2.7 get\_chain\_id()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.get_chain_id (
    self )
```

Retrieve chain ID.

#### 12.14.2.8 get\_contract\_instance()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.get_contract_↵
instance (
    self,
    contract_file_name,
    contract_address )
```

This function returns two contract instances.  
The first is meant for committing transactions or reading from  
a blockchain.  
The second one is specifically meant for event listening.

Parameters:

contract_file_name	Contract filename
contract_address	Ethereum contract address

Returns:

Two contract instances as explained above.

#### 12.14.2.9 `get_contract_instance_from_json()`

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.get_contract_↵  
instance_from_json (   
    self,   
    json_file_name,   
    contract_address )
```

Return two contract instances from a JSON file.  
The first is meant for committing transactions or reading from  
a blockchain.  
The second one is specifically meant for event listening.

Parameters:

json\_file\_name     JSON filename  
contract\_address   Ethereum contract address

Returns:

Two contract instances as explained above.

#### 12.14.2.10 `get_gas_limit()`

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.get_gas_limit (   
    self )
```

Retrieve gas limit.

#### 12.14.2.11 `get_gas_price()`

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.get_gas_price (   
    self )
```

Retrieve gas price.

#### 12.14.2.12 `get_transaction_params()`

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.get_transaction_↵  
_params (   
    self )
```

Construct a dictionary with required parameters  
to submit the transaction.  
Return dict containing chain id, gas, gas limit, and nonce.

### 12.14.2.13 get\_txn\_nonce()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.get_txn_nonce (
    self )
```

Return a transaction nonce. Derived from the transaction address.

### 12.14.2.14 sign\_execute\_raw\_transaction()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_wrapper.EthereumWrapper.sign_execute_↵
raw_transaction (
    self,
    tx_dict )
```

Sign the raw transaction with a private key, send it,  
and wait for receipts.

Parameters:

tx\_dict      Raw transaction to sign

Returns:

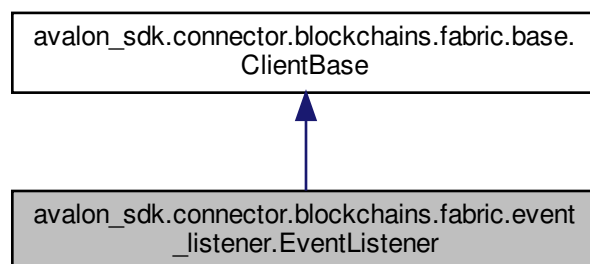
Transaction receipt on success or None on error.

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

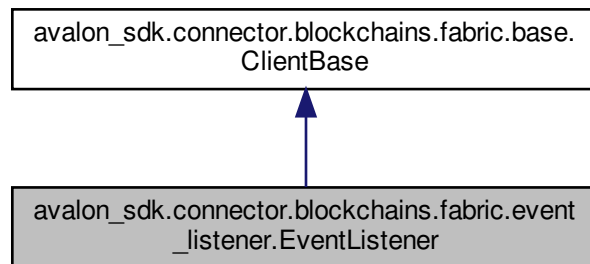
- /home/dano/git/avalon/sdk/avalon\_sdk/connector/blockchains/ethereum/ethereum\_wrapper.py

## 12.15 avalon\_sdk.connector.blockchains.fabric.event\_listener.EventListener Class Reference

Inheritance diagram for avalon\_sdk.connector.blockchains.fabric.event\_listener.EventListener:



Collaboration diagram for `avalon_sdk.connector.blockchains.fabric.event_listener.EventListener`:



## Public Member Functions

- `def __init__ (self, profile, channel_name, org_name, peer_name, user_name)`
- `def handler (self)`
- `def handler (self, handler)`
- `def chaincode (self)`
- `def chaincode (self, chaincode)`
- `def event (self)`
- `def event (self, event)`
- `def config (self)`
- `def config (self, config)`
- `def start_event_handling (self)`
- `def stop_event_handling (self, seconds=0)`
- `def get_single_event (self)`

## Additional Inherited Members

### 12.15.1 Detailed Description

Utility class to listen to Fabric block chain events.

### 12.15.2 Member Function Documentation

#### 12.15.2.1 config()

```
def avalon_sdk.connector.blockchains.fabric.event_listener.EventListener.config (
    self,
    config )
```

Open configuration file named config.

### 12.15.2.2 get\_single\_event()

```
def avalon_sdk.connector.blockchains.fabric.event_listener.EventListener.get_single_event (
    self )
```

Start event listener and listen for particular event.  
Once we got the particular event then unregister and  
close the event listenerhub.

### 12.15.2.3 start\_event\_handling()

```
def avalon_sdk.connector.blockchains.fabric.event_listener.EventListener.start_event_handling
(
    self )
```

Start event listener and listen for events forever  
Only way to stop the event listener by calling  
stop\_event\_listener()

### 12.15.2.4 stop\_event\_handling()

```
def avalon_sdk.connector.blockchains.fabric.event_listener.EventListener.stop_event_handling (
    self,
    seconds = 0 )
```

Stop event listener.

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

- /home/dano/git/avalon/sdk/avalon\_sdk/connector/blockchains/fabric/event\_listener.py

## 12.16 avalon\_sdk.connector.blockchains.ethereum.ethereum\_listener.EventProcessor Class Reference

### Public Member Functions

- def `__init__` (self, config)
- def `listener` (self, event\_filter)
- def `handler` (self, callback, kargs, kwargs)
- def `sync_handler` (self, check\_event\_callback=None, kargs, kwargs)
- def `start` (self, event\_filter, callback, kargs, kwargs)
- def `get_event_synchronously` (self, event\_filter, callback, kargs, kwargs)
- def `stop` (self)

## Public Attributes

- **queue**
- **listeners**
- **handlers**

### 12.16.1 Detailed Description

This class provides an event processor to capture events then send these events to event listeners.

### 12.16.2 Member Function Documentation

#### 12.16.2.1 `get_event_synchronously()`

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_listener.EventProcessor.get_event_↵  
synchronously (   
    self,   
    event_filter,   
    callback,   
    kargs,   
    kwargs )
```

Get a single event synchronously using the event\_filter provided.

Returns an event received for the event\_filter used.

#### 12.16.2.2 `handler()`

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_listener.EventProcessor.handler (   
    self,   
    callback,   
    kargs,   
    kwargs )
```

Start event handler to handle events.



### 12.16.2.3 listener()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_listener.EventProcessor.listener (
    self,
    event_filter )
```

Listen to new events since the last poll on this filter.  
Although this method uses events, it is not fully asynchronous.

### 12.16.2.4 start()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_listener.EventProcessor.start (
    self,
    event_filter,
    callback,
    kargs,
    kwargs )
```

Start event processor in an infinite loop.

### 12.16.2.5 stop()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_listener.EventProcessor.stop (
    self )
```

Stop the event processor that was started with start().

### 12.16.2.6 sync\_handler()

```
def avalon_sdk.connector.blockchains.ethereum.ethereum_listener.EventProcessor.sync_handler (
    self,
    check_event_callback = None,
    kargs,
    kwargs )
```

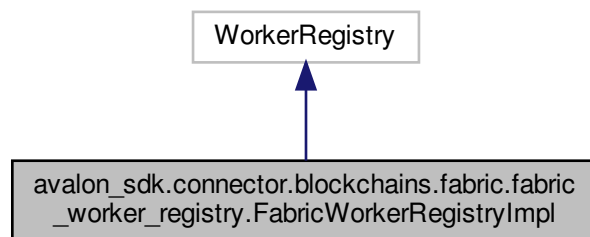
Start a synchronous event handler to handle an event.

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

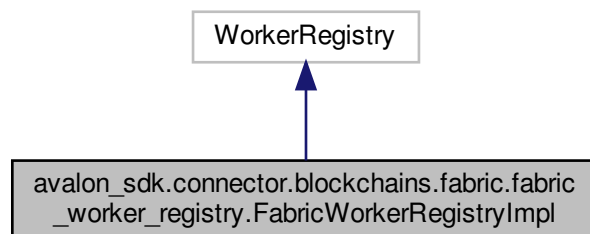
- /home/dano/git/avalon/sdk/avalon\_sdk/connector/blockchains/ethereum/ethereum\_listener.py

## 12.17 `avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorkerRegistry`↔ RegistryImpl Class Reference

Inheritance diagram for `avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorkerRegistry`↔  
Impl:



Collaboration diagram for `avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorkerRegistry`↔  
Impl:



### Public Member Functions

- `def __init__ (self, config)`
- `def worker_lookup (self, worker_type=None, org_id=None, application_id=None, id=None)`
- `def worker_retrieve (self, worker_id, id=None)`
- `def worker_lookup_next (self, worker_type, org_id, application_id, lookup_tag, id=None)`
- `def worker_register (self, worker_id, worker_type, org_id, application_ids, details, id=None)`
- `def worker_set_status (self, worker_id, status, id=None)`
- `def worker_update (self, worker_id, details, id=None)`

### Public Attributes

- `CHAIN_CODE`

### 12.17.1 Detailed Description

This class provide worker APIs which interact with the Hyperledger Fabric blockchain. Detailed method descriptions are available in the WorkerRegistry interface.

### 12.17.2 Constructor & Destructor Documentation

#### 12.17.2.1 \_\_init\_\_()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorkerRegistryImpl.↵
__init__ (
    self,
    config )
```

Parameters:

config     Dictionary containing Fabric-specific parameters

### 12.17.3 Member Function Documentation

#### 12.17.3.1 worker\_lookup()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorkerRegistryImpl.↵
worker_lookup (
    self,
    worker_type = None,
    org_id = None,
    application_id = None,
    id = None )
```

Lookup a worker identified worker\_type, org\_id, and application\_id. All fields are optional and, if present, condition should match for all fields. If none are passed it should return all workers.

If the list is too large to fit into a single response (the maximum number of entries in a single response is implementation specific), the smart contract should return the first batch of the results and provide a lookup\_tag that can be used by the caller to retrieve the next batch by calling worker\_lookup\_next.

Parameters:

worker_type	Optional characteristic of workers for which you may wish to search
org_id	Optional organization ID to which a worker belongs
application_id	Optional application type ID that is supported by the worker
id	Optional JSON RPC request ID

Returns:

Tuple containing workers count, lookup tag, and list of worker IDs:

`total_count` Total number of entries matching a specified lookup criteria. If this number is larger than the size of the IDs array, the caller should use `lookupTag` to call `worker_lookup_next` to retrieve the rest of the IDs

`lookup_tag` Optional parameter. If it is returned, it means that there are more matching worker IDs, which can then be retrieved by calling function `worker_lookup_next` with this tag as an input parameter

`ids` Array of the worker IDs that match the input parameters

On error returns None.

### 12.17.3.2 `worker_lookup_next()`

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorkerRegistryImpl.↵
worker_lookup_next (
    self,
    worker_type,
    org_id,
    application_id,
    lookup_tag,
    id = None )
```

Retrieve additional worker lookup results after calling `worker_lookup`.

Parameters:

`worker_type` Characteristic of Workers for which you may wish to search.

`org_id` Organization ID to which a worker belongs

`application_id` Optional application type ID that is supported by the worker

`lookup_tag` is returned by a previous call to either this function or to `worker_lookup`

`id` Optional Optional JSON RPC request ID

Returns:

Tuple containing the following:

`total_count` Total number of entries matching this lookup criteria. If this number is larger than the number of IDs returned so far, the caller should use `lookupTag` to call `worker_lookup_next` to retrieve the rest of the IDs

`new_lookup_tag` Optional parameter. If it is returned, it means that there are more matching worker IDs that can be retrieved by calling this function again with this tag as an input parameter

`ids` Array of the worker IDs that match the input parameters

On error returns None.

### 12.17.3.3 worker\_register()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorkerRegistryImpl.↵
worker_register (
    self,
    worker_id,
    worker_type,
    org_id,
    application_ids,
    details,
    id = None )
```

Register a new worker with details of the worker.

Parameters:

worker\_id            Worker ID value. E.g., a Fabric address  
worker\_type          Type of Worker. Currently defined types are:  
    \* "TEE-SGX": an Intel SGX Trusted Execution Environment  
    \* "MPC": Multi-Party Compute  
    \* "ZK": Zero-Knowledge  
org\_id               Optional parameter representing the organization that hosts the Worker, e.g. a bank in the consortium or anonymous entity  
application\_ids      Optional parameter that defines application types supported by the Worker  
details               Detailed information about the worker in JSON RPC format as defined in <https://entethalliance.github.io/trusted-computing/spec.html#common-data-for-all-worker-types>  
id                    Optional Optional JSON RPC request ID

Returns:

ContractResponse.SUCCESS on success or  
ContractResponse.ERROR on error.

### 12.17.3.4 worker\_retrieve()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorkerRegistryImpl.↵
worker_retrieve (
    self,
    worker_id,
    id = None )
```

Retrieve the worker identified by worker ID.

Parameters:

worker\_id           Worker ID of the registry whose details are requested  
id                    Optional Optional JSON RPC request ID

Returns:

Tuple containing worker status (defined in worker\_set\_status), worker type, organization ID, list of application IDs, and worker details (JSON RPC string).

On error returns None.

### 12.17.3.5 worker\_set\_status()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorkerRegistryImpl.↵
worker_set_status (
    self,
    worker_id,
    status,
    id = None )
```

Set the registry status identified by worker ID

Parameters:

worker\_id Worker ID value. E.g., a Fabric address  
status Worker status. The currently defined values are:  
1 - worker is active  
2 - worker is temporarily "off-line"  
3 - worker is decommissioned  
4 - worker is compromised  
id Optional Optional JSON RPC request ID

Returns:

ContractResponse.SUCCESS on success  
or ContractResponse.ERROR on error.

### 12.17.3.6 worker\_update()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry.FabricWorkerRegistryImpl.↵
worker_update (
    self,
    worker_id,
    details,
    id = None )
```

Update a worker with details data.

Parameters:

worker\_id Worker ID, e.g. a Fabric address  
details Detailed information about the worker in JSON format  
id Optional Optional JSON RPC request ID

Returns:

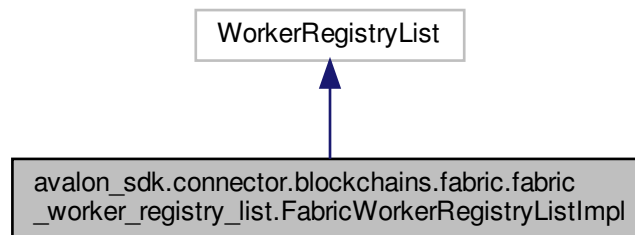
ContractResponse.SUCCESS on success  
or ContractResponse.ERROR on error.

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

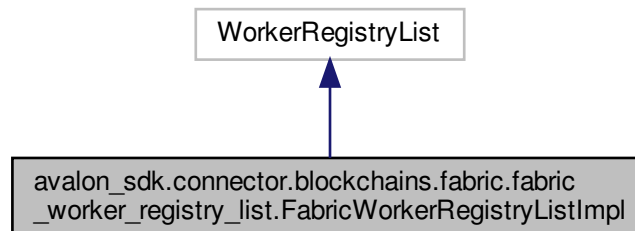
- /home/dano/git/avalon/sdk/avalon\_sdk/connector/blockchains/fabric/fabric\_worker\_registry.py

## 12.18 `avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistryListImpl` Class Reference

Inheritance diagram for `avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistryListImpl`:



Collaboration diagram for `avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistryListImpl`:



### Public Member Functions

- `def __init__ (self, config)`
- `def registry_lookup (self, app_type_id=None)`
- `def registry_retrieve (self, org_id)`
- `def registry_lookup_next (self, app_type_id, lookup_tag)`
- `def registry_add (self, org_id, uri, sc_addr, app_type_ids)`
- `def registry_update (self, org_id, uri, sc_addr, app_type_ids)`
- `def registry_set_status (self, org_id, status)`

### Public Attributes

- `CHAIN_CODE`

### 12.18.1 Detailed Description

This class provide APIs to read/write registry entries of workers, which is stored in the Hyperledger Fabric blockchain.

### 12.18.2 Constructor & Destructor Documentation

#### 12.18.2.1 `__init__()`

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistry←
ListImpl.__init__ (
    self,
    config )
```

Parameters:

`config` Dictionary containing Fabric-specific parameters.

### 12.18.3 Member Function Documentation

#### 12.18.3.1 `registry_add()`

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistry←
ListImpl.registry_add (
    self,
    org_id,
    uri,
    sc_addr,
    app_type_ids )
```

Add a new registry.

Parameters:

`org_id` bytes[] identifies organization that hosts the registry, e.g. a bank in the consortium or an anonymous entity

`uri` String defining a URI for this registry that supports the Off-Chain Worker Registry JSON RPC API

`sc_addr` bytes[] defines a Fabric chain code name that runs the Worker Registry Smart Contract API smart contract for this registry

`app_type_ids` []bytes[] is an optional parameter that defines application types supported by the worker managed by the registry

Returns:

Transaction receipt on success or None on error.



### 12.18.3.2 registry\_lookup()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistryListImpl.registry_lookup (
    self,
    app_type_id = None )
```

Registry Lookup identified by application type ID

Parameters:

app\_type\_id Application type ID to lookup in the registry

Returns:

Tuple containing totalCount, lookupTag, and ids on success:  
 totalCount Total number of entries matching a specified lookup  
 criteria. If this number is larger than the size of the  
 ids array, the caller should use the lookupTag to call  
 registry\_lookup\_next to retrieve the rest of the IDs  
 lookupTag Optional parameter. If it is returned, it means that  
 there are more matching registry IDs that can be  
 retrieved by calling the function registry\_lookup\_next  
 with this tag as an input parameter.  
 ids Array of the registry organization ids that match the  
 input parameters.

Returns None on error.

### 12.18.3.3 registry\_lookup\_next()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistryListImpl.registry_lookup_next (
    self,
    app_type_id,
    lookup_tag )
```

Get additional registry lookup results.  
 This function is called to retrieve additional results of the  
 Registry lookup initiated by the registry\_lookup call.

Parameters:

app\_type\_id Application type ID that has to be  
 supported by the workers retrieved  
 lookup\_tag Returned by a previous call to either this function  
 or to registry\_lookup

Returns:

Outputs a tuple on success containing the following:  
 total\_count Total number of entries matching the lookup  
 criteria. If this number is larger than the number  
 of IDs returned so far, the caller should use  
 lookup\_tag to call registry\_lookup\_next to  
 retrieve the rest of the IDs  
 new\_lookup\_tag is an optional parameter. If it is returned, it means  
 that there are more matching registry IDs that can be  
 retrieved by calling this function again with this tag  
 as an input parameter  
 ids Array of the registry IDs that match the input  
 parameters

Returns None on error.

**12.18.3.4 registry\_retrieve()**

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistry←
ListImpl.registry_retrieve (
    self,
    org_id )
```

Retrieve registry information identified by the organization ID.

Parameters:

org\_id                      Organization ID to lookup

Returns:

Tuple containing following on success:

uri	String defines a URI for this registry that supports the Off-Chain Worker Registry JSON RPC API. It will be None for the proxy model
sc_addr	Fabric address for worker registry smart contract address
application_type_ids	List of application ids (array of byte[])
status	Status of the registry

Returns None on error.

**12.18.3.5 registry\_set\_status()**

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistry←
ListImpl.registry_set_status (
    self,
    org_id,
    status )
```

Set registry status.

Parameters:

org\_id bytes[] identifies organization that hosts the registry, e.g. a bank in the consortium or an anonymous entity

status Defines the registry status to set. The currently defined values are:

- 1 - the registry is active
- 2 - the registry is temporarily "off-line"
- 3 - the registry is decommissioned

Returns:

Transaction receipt on success or None on error.

## 12.18.3.6 registry\_update()

```
def avalon_sdk.connector.blockchains.fabric.fabric_worker_registry_list.FabricWorkerRegistry←
ListImpl.registry_update (
    self,
    org_id,
    uri,
    sc_addr,
    app_type_ids )
```

Update a registry.

Parameters:

org\_id bytes[] identifies organization that hosts the registry, e.g. a bank in the consortium or an anonymous entity

uri string that defines a URI for this registry that supports the Off-Chain Worker Registry JSON RPC API

sc\_addr bytes[] defines a Fabric chain code name that runs the Worker Registry Smart Contract API smart contract for this registry

app\_type\_ids []bytes[] is an optional parameter that defines application types supported by the worker managed by the registry

Returns:

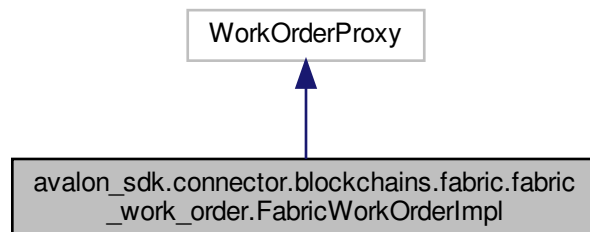
Transaction receipt on success or None on error.

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

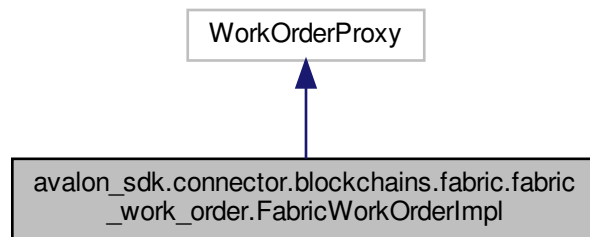
- /home/dano/git/avalon/sdk/avalon\_sdk/connector/blockchains/fabric/fabric\_worker\_registry\_list.py

## 12.19 avalon\_sdk.connector.blockchains.fabric.fabric\_work\_order.FabricWorkOrderImpl Class Reference

Inheritance diagram for avalon\_sdk.connector.blockchains.fabric.fabric\_work\_order.FabricWorkOrderImpl:



Collaboration diagram for `avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl`:



## Public Member Functions

- `def __init__ (self, config)`
- `def work_order_submit (self, work_order_id, worker_id, requester_id, work_order_request, id=None)`
- `def work_order_get_result (self, work_order_id, id=None)`
- `def work_order_complete (self, work_order_id, work_order_response)`
- `def encryption_key_start (self, tag)`
- `def encryption_key_get (self, worker_id, requester_id, last_used_key_nonce=None, tag=None, signature_nonce=None, signature=None)`
- `def encryption_key_set (self, worker_id, encryption_key, encryption_nonce, tag, signature)`
- `def get_work_order_submitted_event_handler (self, handler_func)`
- `def get_work_order_completed_event_handler (self, handler_func)`

## Public Attributes

- **CHAIN\_CODE**
- **WORK\_ORDER\_SUBMITTED\_EVENT\_NAME**
- **WORK\_ORDER\_COMPLETED\_EVENT\_NAME**

### 12.19.1 Detailed Description

This class provides work order management APIs which interact with the Fabric blockchain. Detail method descriptions are available in `WorkOrder` interface.

### 12.19.2 Constructor & Destructor Documentation

**12.19.2.1 \_\_init\_\_()**

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl.__init__ (
    self,
    config )
```

Parameters:

config     Dictionary containing Fabric-specific parameters

**12.19.3 Member Function Documentation****12.19.3.1 encryption\_key\_get()**

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl.encryption_key_get (
    self,
    worker_id,
    requester_id,
    last_used_key_nonce = None,
    tag = None,
    signature_nonce = None,
    signature = None )
```

Get worker's key from Fabric blockchain.  
Not supported for Fabric.

**12.19.3.2 encryption\_key\_set()**

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl.encryption_key_set (
    self,
    worker_id,
    encryption_key,
    encryption_nonce,
    tag,
    signature )
```

Set worker's encryption key.  
Not supported for Fabric.

### 12.19.3.3 encryption\_key\_start()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl.encryption_key_start (
    self,
    tag )
```

Initiate setting the encryption key of the worker.  
Not supported for Fabric.

### 12.19.3.4 get\_work\_order\_completed\_event\_handler()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl.get_work_order_completed_event_handler (
    self,
    handler_func )
```

Start event handler loop for a workOrderCompleted event.

Parameters:  
handler\_func Callback function name as a string

### 12.19.3.5 get\_work\_order\_submitted\_event\_handler()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl.get_work_order_submitted_event_handler (
    self,
    handler_func )
```

Start event handler loop for a workOrderSubmitted event.

Parameters:  
handler\_func Callback function name as a string

Returns:  
Event handler object.

### 12.19.3.6 work\_order\_complete()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl.work_order←
    _complete (
        self,
        work_order_id,
        work_order_response )
```

This function is called by the Worker Service to complete a work order successfully or in error. This API is for the proxy model.

Parameters:

work\_order\_id Unique ID to identify the work order request  
work\_order\_response Work order response data in a string

Returns:

errorCode 0 on success or non-zero on error.

### 12.19.3.7 work\_order\_get\_result()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl.work_order←
    _get_result (
        self,
        work_order_id,
        id = None )
```

Query blockchain to get work order result.

Parameters:

work\_order\_id Work Order ID that was sent in the corresponding work\_order\_submit request  
id Optional JSON RPC request ID

Returns:

Tuple containing work order status, worker id, work order request, work order response, and error code.  
None on error.

### 12.19.3.8 work\_order\_submit()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order.FabricWorkOrderImpl.work_order←
    _submit (
        self,
        work_order_id,
        worker_id,
        requester_id,
        work_order_request,
        id = None )
```

Submit work order request to the Fabric block chain.

Parameters:

work\_order\_id      Unique ID of the work order request  
 worker\_id          Identifier for the worker  
 requester\_id      Unique id to identify the requester  
 work\_order\_request JSON RPC string work order request.  
                     Complete definition at work\_order.py and  
                     defined in EEA specification 6.1.1  
 id                  Optional JSON RPC request ID

Returns:

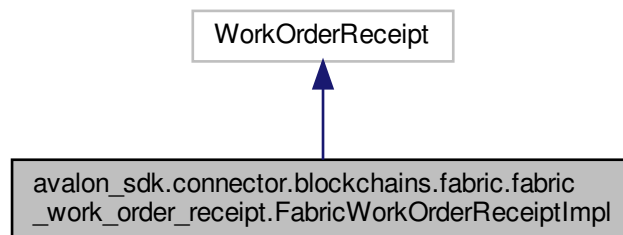
0 on success and non-zero on error.

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

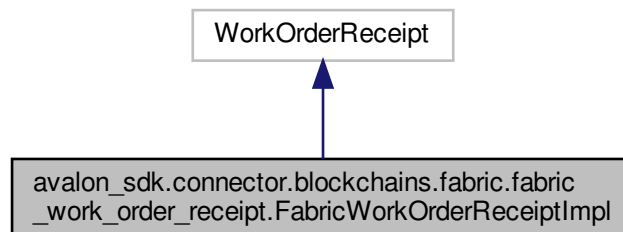
- /home/dano/git/avalon/sdk/avalon\_sdk/connector/blockchains/fabric/fabric\_work\_order.py

## 12.20 avalon\_sdk.connector.blockchains.fabric.fabric\_work\_order\_receipt.FabricWorkOrderReceiptImpl Class Reference

Inheritance diagram for avalon\_sdk.connector.blockchains.fabric.fabric\_work\_order\_receipt.FabricWorkOrderReceiptImpl:



Collaboration diagram for avalon\_sdk.connector.blockchains.fabric.fabric\_work\_order\_receipt.FabricWorkOrderReceiptImpl:





## Public Member Functions

- `def __init__ (self, config)`
- `def work_order_receipt_create (self, work_order_id, worker_id, worker_service_id, requester_id, receipt_  
create_status, work_order_request_hash)`
- `def work_order_receipt_update (self, work_order_id, updater_id, update_type, update_data, update_  
signature=None, signature_rules=None)`
- `def work_order_receipt_retrieve (self, work_order_id)`
- `def work_order_receipt_update_retrieve (self, work_order_id, updater_id, update_index)`
- `def work_order_receipt_lookup (self, worker_service_id, worker_id, requester_id, receipt_status)`
- `def work_order_receipt_lookup_next (self, worker_service_id, worker_id, requester_id, receipt_status, last_  
lookup_tag)`

## Public Attributes

- **CHAIN\_CODE**

### 12.20.1 Detailed Description

This class provides work order receipt management APIs which interact with the Fabric blockchain. Detailed method descriptions are available in the WorkOrderReceipt interface.

### 12.20.2 Constructor & Destructor Documentation

#### 12.20.2.1 \_\_init\_\_()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order_receipt.FabricWorkOrderReceiptImpl.__init__ (
    self,
    config )
```

Parameters:  
config Dict containing Fabric-specific parameters.

### 12.20.3 Member Function Documentation

**12.20.3.1 work\_order\_receipt\_create()**

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order_receipt.FabricWorkOrderReceipt←
Impl.work_order_receipt_create (
    self,
    work_order_id,
    worker_id,
    worker_service_id,
    requester_id,
    receipt_create_status,
    work_order_request_hash )
```

Create work order receipt in the Fabric block chain.

Parameters:

work\_order\_id                    ID of the Work Order  
worker\_id                        Worker id that should execute the Work Order  
worker\_service\_id                ID of the Worker Service that  
                                  hosts the Worker  
requester\_id                    ID of the requester  
receipt\_create\_status          Initial receipt status defined  
                                  in EEA spec 7.1.1  
work\_order\_request\_hash        Hash value of the work order request as  
                                  defined in EEA spec 6.7

Returns:

0 on success and -1 on error.

**12.20.3.2 work\_order\_receipt\_lookup()**

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order_receipt.FabricWorkOrderReceipt←
Impl.work_order_receipt_lookup (
    self,
    worker_service_id,
    worker_id,
    requester_id,
    receipt_status )
```

Lookup a work order receipt.

Parameters:

worker\_service\_id               Worker Service ID whose receipts will be  
                                  retrieved  
worker\_id                        Worker Id whose receipts are requested  
requester\_id                    ID of the entity requesting receipts  
receipt\_status                 Defines the status of the receipts retrieve  
id                                Optional JSON RPC request ID

Returns:

Tuple containing total count, last\_lookup\_tag, and  
list of work order IDs, on success:  
total\_count                    Total number of receipts matching the lookup criteria.  
                                  If this number is bigger than the size of the ids  
                                  array, the caller should use a lookup\_tag to call  
                                  work\_order\_receipt\_lookup\_next() to retrieve the rest  
                                  of the receipt IDs.  
last\_lookup\_tag                Optional lookup\_tag when the receipts exceed the ids  
                                  array size  
ids                              Array of work order receipt ids that match the input

On error, returns -1.

### 12.20.3.3 work\_order\_receipt\_lookup\_next()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order_receipt.FabricWorkOrderReceiptImpl.work_order_receipt_lookup_next (
    self,
    worker_service_id,
    worker_id,
    requester_id,
    receipt_status,
    last_lookup_tag )
```

Retrieve subsequent work order receipts after calling  
work\_order\_receipt\_lookup().

**Parameters:**

worker\_service\_id Worker Service ID  
worker\_id Worker ID value derived from the worker's DID  
requester\_id Requester ID  
last\_lookup\_tag One of the output parameters for function  
work\_order\_receipt\_lookup()  
id Optional JSON RPC request ID

**Returns:**

On success, return a tuple containing total count, look up tag, and  
list of work order IDs:

total\_count Total number of receipts matching the lookup  
criteria  
lookup\_tag Optional parameter. If it is returned, it means  
that there are more matching receipts that can be  
retrieved by calling this function again and with  
this tag as an input parameter.  
ids Array of the Work Order receipt IDs that match the  
input criteria from the corresponding call to  
work\_order\_receipt\_lookup().  
Return -1 on error.

### 12.20.3.4 work\_order\_receipt\_retrieve()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order_receipt.FabricWorkOrderReceiptImpl.work_order_receipt_retrieve (
    self,
    work_order_id )
```

Retrieve a Work Order Receipt.

**Parameters:**

work\_order\_id ID of the Work Order to be retrieved  
id Optional JSON RPC request ID

**Returns:**

worker\_service\_id, requester\_id, work\_order\_id, receipt\_create\_status,  
and work\_order\_request\_hash, as defined in work\_order\_receipt\_create().  
Return -1 on error.

### 12.20.3.5 work\_order\_receipt\_update()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order_receipt.FabricWorkOrderReceipt←
Impl.work_order_receipt_update (
    self,
    work_order_id,
    updater_id,
    update_type,
    update_data,
    update_signature = None,
    signature_rules = None )
```

Update a Work Order Receipt.

Parameters:

work\_order\_id Work Order ID that was sent in the  
corresponding work\_order\_submit request  
updater\_id ID of the updating entity. It is optional if it  
is the same as the transaction sender address  
update\_type Type of the Work Order update that defines  
how the update should be handled  
update\_data Update-specific data that depends on the  
updater type defined in EEA spec 7.1.2  
update\_signature Optional signature of concatenated  
work\_order\_id, update\_type, and update\_data  
signature\_rules Defines hashing and signing algorithms,  
that are separated by forward slash '/'

Returns:

0 on success, -1 on error.

### 12.20.3.6 work\_order\_receipt\_update\_retrieve()

```
def avalon_sdk.connector.blockchains.fabric.fabric_work_order_receipt.FabricWorkOrderReceipt←
Impl.work_order_receipt_update_retrieve (
    self,
    work_order_id,
    updater_id,
    update_index )
```

Retrieve an update to a work order receipt.

Parameters:

work\_order\_id Work Order ID that was sent in the  
corresponding work\_order\_submit request  
updater\_id ID of the updating entity. Ignored if null  
update\_index Index of the update to retrieve  
Value "0xFFFFFFFF" is reserved to retrieve the  
last received update

Returns:

On success, return updater\_id, update\_type, update\_data,  
update\_signature, signature\_rules, as defined in  
work\_order\_receipt\_update(), and update\_count.  
On error, return -1.

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

- /home/dano/git/avalon/sdk/avalon\_sdk/connector/blockchains/fabric/fabric\_work\_order\_receipt.py

## 12.21 avalon\_sdk.connector.blockchains.fabric.fabric\_wrapper.FabricWrapper Class Reference

### Public Member Functions

- def `__init__` (self, config)
- def `invoke_chaincode` (self, chaincode\_name, method\_name, params)
- def `get_event_handler` (self, event\_name, chain\_code, handler\_func)

### 12.21.1 Detailed Description

Fabric wrapper class to interact with Fabric blockchain.  
It provides wrapper functions to invoke and query chain code.

### 12.21.2 Constructor & Destructor Documentation

#### 12.21.2.1 `__init__()`

```
def avalon_sdk.connector.blockchains.fabric.fabric_wrapper.FabricWrapper.__init__ (
    self,
    config )
```

Constructor to initialize wrapper with required parameter.

Parameters:  
config     Dictionary containing parameters for Fabric.  
          These parameters are read from a .toml file

### 12.21.3 Member Function Documentation

#### 12.21.3.1 `get_event_handler()`

```
def avalon_sdk.connector.blockchains.fabric.fabric_wrapper.FabricWrapper.get_event_handler (
    self,
    event_name,
    chain_code,
    handler_func )
```

Create event handler object.

Parameters:  
event\_name     String to identify the event name  
chain\_code     Chain code name as string  
handler\_func   Callback function name

Returns:  
Event object

### 12.21.3.2 invoke\_chaincode()

```
def avalon_sdk.connector.blockchains.fabric.fabric_wrapper.FabricWrapper.invoke_chaincode (
    self,
    chaincode_name,
    method_name,
    params )
```

This is wrapper method to invoke chain code.

**Parameters:**

chaincode\_name Name of the chain code  
 method\_name Chain code method name  
 params List of arguments to method

**Returns:**

If the call to chain code query, then it returns the payload of the chain code response on success or None on error.  
 If the call is invoking chain code, then it returns ContractResponse.SUCCESS on success and ContractResponse.ERROR on failure.

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

- /home/dano/git/avalon/sdk/avalon\_sdk/connector/blockchains/fabric/fabric\_wrapper.py

## 12.22 FileIoExecutor Class Reference

### Public Member Functions

- void **SetIoHandlerId** (uint32\_t handler\_id)
- void **SetFileName** (std::string file\_name)
- uint32\_t **GetIoHandlerId** (const char \*handlerName)
- size\_t **GetMaxFileSize** ()
- size\_t **GetMaxIoResultSize** ()
- uint32\_t **FileOpen** (uint8\_t \*result, size\_t result\_size)
- uint32\_t **FileClose** (uint8\_t \*result, size\_t result\_size)
- uint32\_t **FileRead** (uint8\_t \*result, size\_t result\_size, uint8\_t \*out\_buf, size\_t out\_buf\_size)
- uint32\_t **FileWrite** (uint8\_t \*result, size\_t result\_size, const uint8\_t \*in\_buf, size\_t in\_buf\_size)
- uint32\_t **FileTell** (uint8\_t \*result, size\_t result\_size, uint8\_t \*out\_buf, size\_t out\_buf\_size)
- uint32\_t **FileSeek** (size\_t position, uint8\_t \*result, size\_t result\_size)
- uint32\_t **FileDelete** (uint8\_t \*result, size\_t result\_size)

### 12.22.1 Member Function Documentation

#### 12.22.1.1 FileClose()

```
uint32_t FileIoExecutor::FileClose (
    uint8_t * result,
    size_t result_size )
```

Closes given file and updates status in the result buffer.

**Parameters**

<i>result</i>	Status of file close operation (0 is success, non-0 is failure)
<i>result_size</i>	Maximum size of the result buffer in bytes

**Returns**

Status of operation (0 on success, non-0 on failure)

**12.22.1.2 FileDelete()**

```
uint32_t FileIoExecutor::FileDelete (
    uint8_t * result,
    size_t result_size )
```

Deletes the file whose name is stored in the [FileIoExecutor](#) instance.

**Parameters**

<i>result</i>	Status of file delete operation (0 is success, non-0 is failure)
<i>result_size</i>	Maximum size of the result buffer in bytes

**Returns**

Status of operation (0 on success, non-0 on failure)

**12.22.1.3 FileOpen()**

```
uint32_t FileIoExecutor::FileOpen (
    uint8_t * result,
    size_t result_size )
```

Opens given file and updates status in the result buffer.

**Parameters**

<i>result</i>	Status of file open operation (0 is success, non-0 is failure)
<i>result_size</i>	Maximum size of the result buffer in bytes

**Returns**

Status of operation (0 on success, non-0 on failure)

#### 12.22.1.4 FileRead()

```
uint32_t FileIoExecutor::FileRead (
    uint8_t * result,
    size_t result_size,
    uint8_t * out_buf,
    size_t out_buf_size )
```

Reads given file, stores content in out buffer and updates status in result buffer.

##### Parameters

<i>result</i>	Status of file read operation (0 is success, non-0 is failure)
<i>result_size</i>	Maximum size of the result buffer in bytes
<i>out_buf</i>	Buffer to hold file content
<i>out_buf_size</i>	Maximum size of out_buf to contain the file contents in bytes

##### Returns

Status of operation (0 on success, non-0 on failure)

#### 12.22.1.5 FileSeek()

```
uint32_t FileIoExecutor::FileSeek (
    size_t position,
    uint8_t * result,
    size_t result_size )
```

Moves the file position the file to the given position and updates the status in result buffer.

##### Parameters

<i>position</i>	Byte offset of new file position
<i>result</i>	Status of file seek operation (0 is success, non-0 is failure)
<i>result_size</i>	Maximum size of the result buffer in bytes

##### Returns

Status of operation (0 on success, non-0 on failure)

#### 12.22.1.6 FileTell()

```
uint32_t FileIoExecutor::FileTell (
    uint8_t * result,
```



```
size_t result_size,  
uint8_t * out_buf,  
size_t out_buf_size )
```

Gets the current position of the file, stores it in buffer out\_buf, and updates status in result buffer.

#### Parameters

<i>result</i>	status of file tell operation (0 is success, non-0 is failure)
<i>result_size</i>	Maximum size of the result buffer in bytes
<i>out_buf</i>	Buffer to hold file position
<i>out_buf_size</i>	Maximum size of out_buf to contain the file position in bytes

#### Returns

Status of operation (0 on success, non-0 on failure)

#### 12.22.1.7 FileWrite()

```
uint32_t FileIoExecutor::FileWrite (  
    uint8_t * result,  
    size_t result_size,  
    const uint8_t * in_buf,  
    size_t in_buf_size )
```

Writes given file with content in input buffer and updates status in result buffer.

#### Parameters

<i>result</i>	Status of file write operation (0 is success, non-0 is failure)
<i>result_size</i>	Maximum size of the result buffer in bytes
<i>in_buf</i>	Buffer with content to be written to the file
<i>in_buf_size</i>	Maximum size of in_buf to write to the file in bytes

#### Returns

Status of operation (0 on success, non-0 on failure)

#### 12.22.1.8 GetIoHandlerId()

```
uint32_t FileIoExecutor::GetIoHandlerId (  
    const char * handler_name )
```

Get the I/O handler ID corresponding to IoHandler handler\_name.

**Parameters**

<i>handlerName</i>	Name of handler
--------------------	-----------------

**Returns**

I/O handler ID. That is, 1 for handler "tcf-base-file-io"  
0 on error

**12.22.1.9 GetMaxFileSize()**

```
size_t FileIoExecutor::GetMaxFileSize ( )
```

Get the maximum size of the buffer used for file I/O.

**Returns**

Maximum buffer size in bytes

**12.22.1.10 GetMaxIoResultSize()**

```
size_t FileIoExecutor::GetMaxIoResultSize ( )
```

Get the maximum size of the result buffer used to store the I/O status.

**Returns**

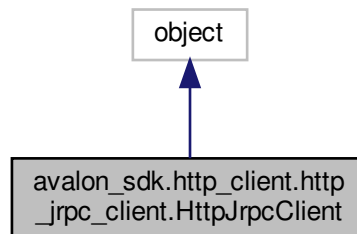
Maximum result buffer size in bytes

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

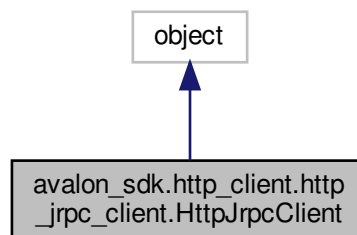
- [/home/dano/git/avalon/common/sgx\\_workload/iohandler/file\\_io.h](#)
- [/home/dano/git/avalon/common/sgx\\_workload/iohandler/file\\_io.cpp](#)

## 12.23 avalon\_sdk.http\_client.http\_rpc\_client.HttpRpcClient Class Reference

Inheritance diagram for avalon\_sdk.http\_client.http\_rpc\_client.HttpRpcClient:



Collaboration diagram for avalon\_sdk.http\_client.http\_rpc\_client.HttpRpcClient:



### Public Member Functions

- `def __init__(self, url)`

### Public Attributes

- **ServiceURL**
- **ProxyHandler**

### 12.23.1 Detailed Description

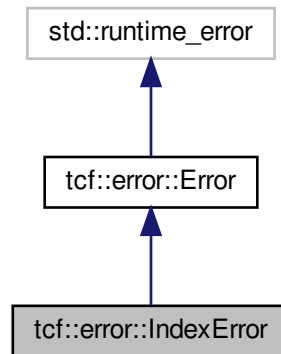
Class to handle HTTP JSON RPC communication by the client.

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

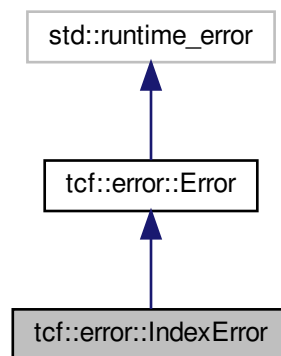
- `/home/dano/git/avalon/sdk/avalon_sdk/http_client/http_rpc_client.py`

## 12.24 tcf::error::IndexError Class Reference

Inheritance diagram for tcf::error::IndexError:



Collaboration diagram for tcf::error::IndexError:



### Public Member Functions

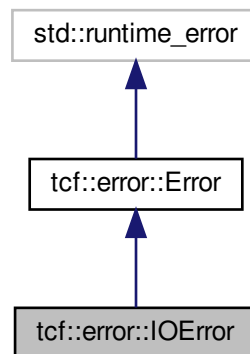
- **IndexError** (const std::string &msg)

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

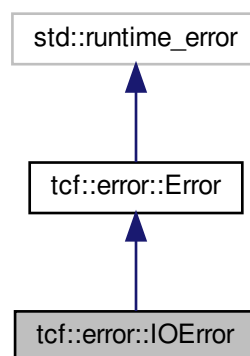
- `/home/dano/git/avalon/common/cpp/error.h`

## 12.25 tcf::error::IOError Class Reference

Inheritance diagram for tcf::error::IOError:



Collaboration diagram for tcf::error::IOError:



### Public Member Functions

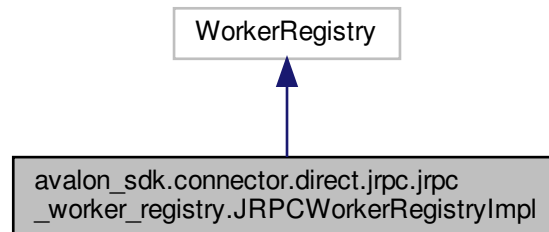
- **IOError** (const std::string &msg)

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

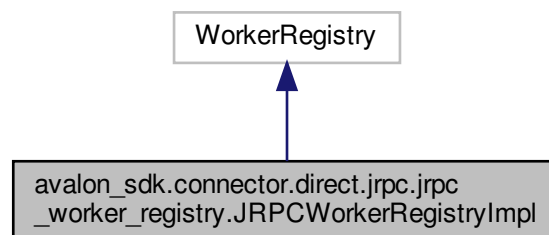
- `/home/dano/git/avalon/common/cpp/error.h`

## 12.26 `avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl` Class Reference

Inheritance diagram for `avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl`:



Collaboration diagram for `avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl`:



### Public Member Functions

- `def __init__(self, config)`
- `def worker_retrieve(self, worker_id, id=None)`
- `def worker_lookup(self, worker_type=None, organization_id=None, application_type_id=None, id=None)`
- `def worker_lookup_next(self, lookup_tag, worker_type=None, organization_id=None, application_type_id=None, id=None)`
- `def worker_register(self, worker_id, worker_type, org_id, application_type_ids, details, id=None)`
- `def worker_update(self, worker_id, details, id=None)`
- `def worker_set_status(self, worker_id, status, id=None)`

### 12.26.1 Detailed Description

This class is to read the worker registry to get the more details of worker.

## 12.26.2 Member Function Documentation

### 12.26.2.1 worker\_lookup()

```
def avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl.worker_lookup
(
    self,
    worker_type = None,
    organization_id = None,
    application_type_id = None,
    id = None )
```

Worker lookup based on worker type, organization ID, and application ID.  
All fields are optional and, if present, condition should match for all fields. If none are passed it should return all workers.

**Parameters:**

**worker\_type** Optional characteristic of Workers for which you may wish to search. Currently defined types are:  
 \* "TEE-SGX": an Intel SGX Trusted Execution Environment  
 \* "MPC": Multi-Party Compute  
 \* "ZK": Zero-Knowledge  
**organization\_id** Optional parameter representing the organization that hosts the Worker, e.g. a bank in the consortium or anonymous entity  
**application\_type\_id** Optional application type that has to be supported by the worker  
**id** Optional JSON RPC request ID

**Returns:**

JRPC response containing number of workers, lookup tag, and list of worker IDs.

### 12.26.2.2 worker\_lookup\_next()

```
def avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl.worker_↵
lookup_next (
    self,
    lookup_tag,
    worker_type = None,
    organization_id = None,
    application_type_id = None,
    id = None )
```

Retrieve subsequent Worker lookup results based on worker type, organization ID, and application ID.  
Similar to workerLookup with additional parameter lookup\_tag.

**Parameters:**

**lookup\_tag** Used to lookup subsequent results after calling

```

        worker_lookup
worker_type      Optional characteristic of Workers for which you
                  may wish to search. Currently defined types are:
                  * "TEE-SGX": an Intel SGX Trusted Execution
                    Environment
                  * "MPC": Multi-Party Compute
                  * "ZK": Zero-Knowledge
organization_id  Optional parameter representing the
                  organization that hosts the Worker,
                  e.g. a bank in the consortium or
                  anonymous entity
application_type_id Optional application type that has to be supported
                  by the worker
id              Optional JSON RPC request ID

```

Returns:

JRPC response containing number of workers,  
lookup tag, and list of worker IDs.

### 12.26.2.3 worker\_register()

```

def avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl.worker_↵
register (
    self,
    worker_id,
    worker_type,
    org_id,
    application_type_ids,
    details,
    id = None )

```

Adds worker details to registry

Parameters:

```

worker_id      Worker ID value derived from the worker's DID
worker_type    Type of Worker. Currently defined types are:
                * "TEE-SGX": an Intel SGX Trusted Execution
                  Environment
                * "MPC": Multi-Party Compute
                * "ZK": Zero-Knowledge
org_id         Organization that hosts the Worker,
                e.g. a bank in the consortium or
                anonymous entity
application_type_ids Application types supported by the worker
id             Optional JSON RPC request ID

```

Returns:

JRPC response with worker registry status.

### 12.26.2.4 worker\_retrieve()

```

def avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl.worker_↵
retrieve (
    self,
    worker_id,
    id = None )

```



Retrieve the worker identified by worker ID.

Parameters:

worker\_id Worker ID value derived from the worker's DID  
id Optional Optional JSON RPC request ID

Returns:

JRPC response containing:  
organization ID, application ID, worker status,  
and worker details.

### 12.26.2.5 worker\_set\_status()

```
def avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl.worker_set_status (
    self,
    worker_id,
    status,
    id = None )
```

Set the worker status to active, offline,  
decommissioned, or compromised state.

Parameters:

worker\_id Worker ID value derived from the worker's DID  
status Worker status value to set  
id Optional JSON RPC request ID

Returns:

JRPC response with status.

### 12.26.2.6 worker\_update()

```
def avalon_sdk.connector.direct.jrpc.jrpc_worker_registry.JRPCWorkerRegistryImpl.worker_update (
    self,
    worker_id,
    details,
    id = None )
```

Update worker with new information.

Parameters:

worker\_id Worker ID value derived from the worker's DID  
details Detailed information about the worker in  
JSON RPC format as defined in  
id Optional JSON RPC request ID

Returns:

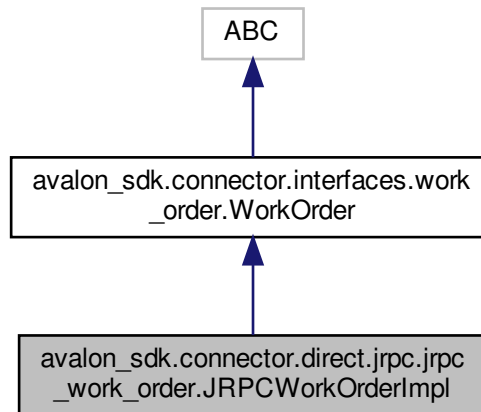
JRPC response with update status.

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

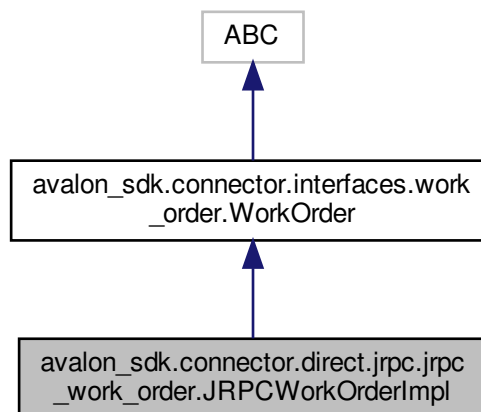
- /home/dano/git/avalon/sdk/avalon\_sdk/connector/direct/jrpc/jrpc\_worker\_registry.py

## 12.27 avalon\_sdk.connector.direct.jrpc.jrpc\_work\_order.JRPCWorkOrderImpl Class Reference

Inheritance diagram for avalon\_sdk.connector.direct.jrpc.jrpc\_work\_order.JRPCWorkOrderImpl:



Collaboration diagram for avalon\_sdk.connector.direct.jrpc.jrpc\_work\_order.JRPCWorkOrderImpl:



### Public Member Functions

- `def __init__(self, config)`
- `def work\_order\_submit(self, work_order_id, worker_id, requester_id, work_order_request, id=None)`

- def [work\\_order\\_get\\_result\\_nonblocking](#) (self, work\_order\_id, id=None)
- def [work\\_order\\_get\\_result](#) (self, work\_order\_id, id=None)
- def [encryption\\_key\\_get](#) (self, worker\_id, requester\_id, last\_used\_key\_nonce=None, tag=None, signature\_nonce=None, signature=None, id=None)
- def [encryption\\_key\\_set](#) (self, worker\_id, encryption\_key, encryption\_nonce, tag, signature\_nonce, signature, id=None)

### 12.27.1 Detailed Description

This class is for to manage to the work orders from client side.

### 12.27.2 Member Function Documentation

#### 12.27.2.1 [encryption\\_key\\_get\(\)](#)

```
def avalon_sdk.connector.direct.jrpc.jrpc_work_order.JRPCWorkOrderImpl.encryption_key_get (
    self,
    worker_id,
    requester_id,
    last_used_key_nonce = None,
    tag = None,
    signature_nonce = None,
    signature = None,
    id = None )
```

API to receive a worker's key.

Parameters:

worker_id	Worker ID of the worker whose encryption key is requested
last_used_key_nonce	Optional nonce associated with the last retrieved key. If it is provided, the key retrieved should be newer than this one. Otherwise any key can be retrieved
tag	Tag that should be associated with the returned key, e.g. the requester ID. This is an optional parameter. If it is not provided, requester_id is used as a key
requester_id	ID of the requester that plans to use the returned key to submit one or more work orders using this key
signature_nonce	Optional nonce associated with the signature and is used only if signature below is also provided
signature	Optional signature of worker_id, last_used_key_nonce, tag, and signature_nonce.
id	Optional JSON RPC request ID

**12.27.2.2 encryption\_key\_set()**

```
def avalon_sdk.connector.direct.jrpc.jrpc_work_order.JRPCWorkOrderImpl.encryption_key_set (
    self,
    worker_id,
    encryption_key,
    encryption_nonce,
    tag,
    signature_nonce,
    signature,
    id = None )
```

API called by a Worker or Worker Service to receive a Worker's key.

**Parameters:**

worker\_id ID of the worker to set an encryption key  
 encryption\_key Encryption key to set  
 encryption\_nonce Nonce associated with the key  
 tag Tag that should be associated with the returned key,  
 e.g. requester ID.  
 signature\_nonce Nonce associated with the signature  
 signature Signature generated by the worker on the worker\_id,  
 tag and encryption\_nonce  
 id Optional JSON RPC request ID

**Returns:**

JRPC response with the result of the operation.

**12.27.2.3 work\_order\_get\_result()**

```
def avalon_sdk.connector.direct.jrpc.jrpc_work_order.JRPCWorkOrderImpl.work_order_get_result (
    self,
    work_order_id,
    id = None )
```

Get the work order result in a blocking way until it gets a result or error.

**Parameters:**

work\_order\_id Work order ID  
 id Optional JSON RPC request ID

**Returns:**

JSON RPC response of dictionary type

**12.27.2.4 work\_order\_get\_result\_nonblocking()**

```
def avalon_sdk.connector.direct.jrpc.jrpc_work_order.JRPCWorkOrderImpl.work_order_get_result↔
_nonblocking (
    self,
    work_order_id,
    id = None )
```

Get the work order result in non-blocking way.

Parameters:

work\_order\_id      Work order ID  
id                    Optional JSON RPC request ID

Returns:

JSON RPC response of dictionary type

### 12.27.2.5 work\_order\_submit()

```
def avalon_sdk.connector.direct.jrpc.jrpc_work_order.JRPCWorkOrderImpl.work_order_submit (
    self,
    work_order_id,
    worker_id,
    requester_id,
    work_order_request,
    id = None )
```

Submit a work order request to an Avalon listener.

Parameters:

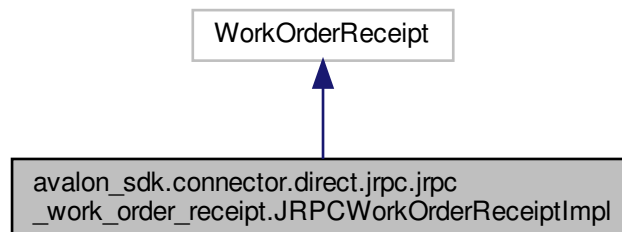
work\_order\_id      Work order ID  
worker\_id           Worker ID value derived from the worker's DID  
requester\_id        Requester ID  
work\_order\_request Work order request in JSON RPC string format  
id                    Optional JSON RPC request ID

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

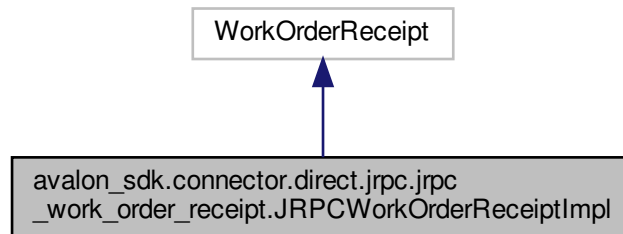
- /home/dano/git/avalon/sdk/avalon\_sdk/connector/direct/jrpc/jrpc\_work\_order.py

## 12.28 avalon\_sdk.connector.direct.jrpc.jrpc\_work\_order\_receipt.JRPCWorkOrderReceiptImpl Class Reference

Inheritance diagram for avalon\_sdk.connector.direct.jrpc.jrpc\_work\_order\_receipt.JRPCWorkOrderReceiptImpl:



Collaboration diagram for `avalon_sdk.connector.direct.jrpc.jrpc_work_order_receipt.JRPCWorkOrderReceiptImpl`:



## Public Member Functions

- `def __init__ (self, config)`
- `def work\_order\_receipt\_create (self, work_order_id, worker_service_id, worker_id, requester_id, receipt_create_status, work_order_request_hash, requester_nonce, requester_signature, signature_rules, receipt_verification_key, id=None)`
- `def work\_order\_receipt\_update (self, work_order_id, updater_id, update_type, update_data, update_signature, signature_rules, id=None)`
- `def work\_order\_receipt\_retrieve (self, work_order_id, id=None)`
- `def work\_order\_receipt\_update\_retrieve (self, work_order_id, updater_id, update_index, id=None)`
- `def work\_order\_receipt\_lookup (self, worker_service_id=None, worker_id=None, requester_id=None, receipt_status=None, id=None)`
- `def work\_order\_receipt\_lookup\_next (self, last_lookup_tag, worker_service_id=None, worker_id=None, requester_id=None, receipt_status=None, id=None)`

### 12.28.1 Detailed Description

This class is an implementation of `WorkOrderReceiptInterface` to manage work order receipts from the client side.

### 12.28.2 Member Function Documentation

#### 12.28.2.1 `work_order_receipt_create()`

```

def avalon_sdk.connector.direct.jrpc.jrpc_work_order_receipt.JRPCWorkOrderReceiptImpl.work_order_receipt_create (
    self,
    work_order_id,
    worker_service_id,
    worker_id,

```

```

    requester_id,
    receipt_create_status,
    work_order_request_hash,
    requester_nonce,
    requester_signature,
    signature_rules,
    receipt_verification_key,
    id = None )

```

Create a Work Order Receipt JSON RPC request and submit to an Avalon listener.

Parameters:

work_order_id	Work order ID
worker_service_id	Worker service ID
worker_id	Worker ID value derived from the worker's DID
requester_id	Requester ID
receipt_create_status	Receipt creation status
work_order_request_hash	Work order request hash value
requester_nonce	Requester generated nonce
requester_signature	Signature generated by the requester
signature_rules	Defines hashing and signing algorithms; separated by forward slash '/'
receipt_verification_key	Receipt verification key
id	Optional JSON RPC request ID

### 12.28.2.2 work\_order\_receipt\_lookup()

```

def avalon_sdk.connector.direct.jrpc.jrpc_work_order_receipt.JRPCWorkOrderReceiptImpl.work_order_receipt_lookup (
    self,
    worker_service_id = None,
    worker_id = None,
    requester_id = None,
    receipt_status = None,
    id = None )

```

Work Order Receipt Lookup

All fields are optional and, if present, condition should match for all fields. If none are passed it should return all work order receipts.

Parameters:

worker_service_id	Optional worker service ID to lookup
worker_id	Optional worker ID value derived from the worker's DID
requester_id	Optional requester ID to lookup
receipt_status	Optional receipt status
id	Optional JSON RPC request ID

### 12.28.2.3 work\_order\_receipt\_lookup\_next()

```
def avalon_sdk.connector.direct.jrpc.jrpc_work_order_receipt.JRPCWorkOrderReceiptImpl.work_order_receipt_lookup_next (
    self,
    last_lookup_tag,
    worker_service_id = None,
    worker_id = None,
    requester_id = None,
    receipt_status = None,
    id = None )
```

Work Order Receipt Lookup Next.  
Call to retrieve subsequent results after calling  
work\_order\_receipt\_lookup or

Parameters:

last_lookup_tag	Last lookup tag returned by work_order_receipt_lookup
worker_service_id	Optional worker service ID to lookup
worker_id	Optional worker ID value derived from the worker's DID
requester_id	Optional requester ID to lookup
receipt_status	Optional receipt status
id	Optional JSON RPC request ID

### 12.28.2.4 work\_order\_receipt\_retrieve()

```
def avalon_sdk.connector.direct.jrpc.jrpc_work_order_receipt.JRPCWorkOrderReceiptImpl.work_order_receipt_retrieve (
    self,
    work_order_id,
    id = None )
```

Retrieve a work order receipt JSON RPC request and submit to an  
Avalon listener.

Parameters:

work_order_id	Work order ID
id	Optional Optional JSON RPC request ID

### 12.28.2.5 work\_order\_receipt\_update()

```
def avalon_sdk.connector.direct.jrpc.jrpc_work_order_receipt.JRPCWorkOrderReceiptImpl.work_order_receipt_update (
    self,
    work_order_id,
    updater_id,
    update_type,
    update_data,
    update_signature,
    signature_rules,
    id = None )
```



Update a Work Order Receipt JSON RPC request and submit an Avalon listener.

Parameters:

<code>work_order_id</code>	Work Order ID
<code>updater_id</code>	Updater ID
<code>update_type</code>	Updater type
<code>update_data</code>	Receipt update data
<code>update_signature</code>	Signature of the update
<code>signature_rules</code>	Defines hashing and signing algorithms; separated by forward slash '/'
<code>id</code>	Optional JSON RPC request ID

#### 12.28.2.6 `work_order_receipt_update_retrieve()`

```
def avalon_sdk.connector.direct.jrpc.jrpc_work_order_receipt.JRPCWorkOrderReceiptImpl.work_order_receipt_update_retrieve (
    self,
    work_order_id,
    updater_id,
    update_index,
    id = None )
```

Retrieve a work order receipt update JSON RPC request and submit to an Avalon listener.

Parameters:

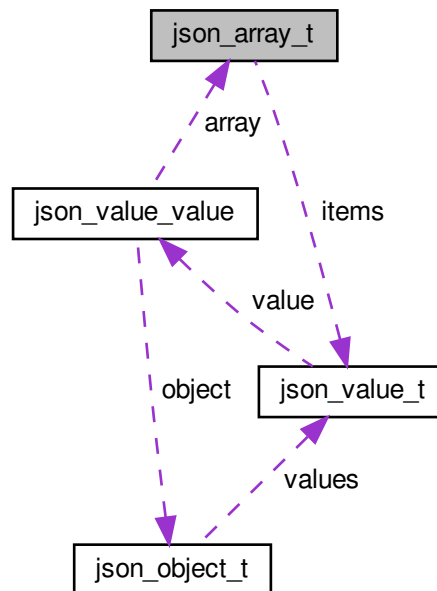
<code>work_order_id</code>	Work order ID
<code>id</code>	Optional Optional JSON RPC request ID

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

- `/home/dano/git/avalon/sdk/avalon_sdk/connector/direct/jrpc/jrpc_work_order_receipt.py`

## 12.29 json\_array\_t Struct Reference

Collaboration diagram for json\_array\_t:



### Public Attributes

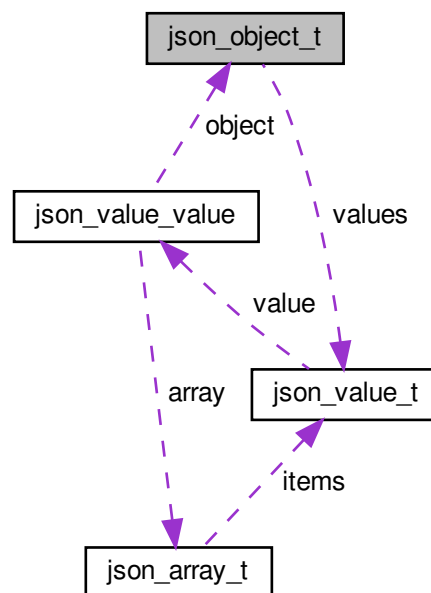
- `JSON_Value ** items`
- `size_t count`
- `size_t capacity`

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

- `/home/dano/git/avalon/common/cpp/packages/parson/parson.cpp`

## 12.30 json\_object\_t Struct Reference

Collaboration diagram for json\_object\_t:



### Public Attributes

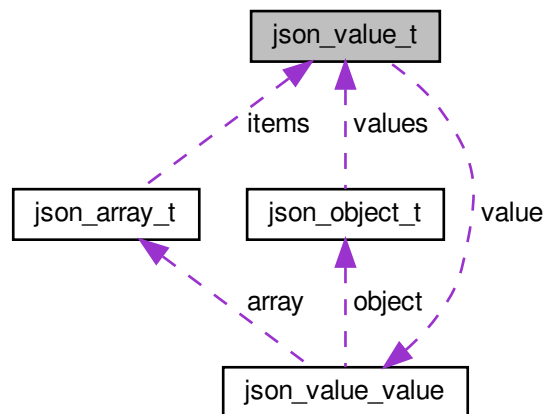
- `char ** names`
- `JSON_Value ** values`
- `size_t count`
- `size_t capacity`

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

- `/home/dano/git/avalon/common/cpp/packages/parson/parson.cpp`

## 12.31 json\_value\_t Struct Reference

Collaboration diagram for json\_value\_t:



### Public Attributes

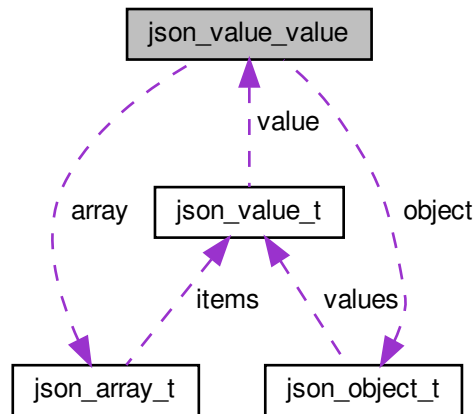
- `JSON_Value_Type` **type**
- `JSON_Value_Value` **value**

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

- `/home/dano/git/avalon/common/cpp/packages/parson/parson.cpp`

## 12.32 json\_value\_value Union Reference

Collaboration diagram for json\_value\_value:



### Public Attributes

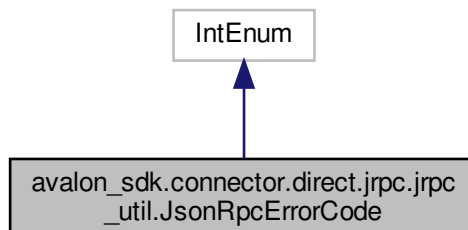
- char \* **string**
- double **number**
- [JSON\\_Object](#) \* **object**
- [JSON\\_Array](#) \* **array**
- int **boolean**
- int **null**

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

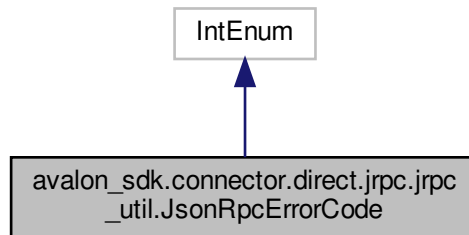
- /home/dano/git/avalon/common/cpp/packages/parson/parson.cpp

## 12.33 avalon\_sdk.connector.direct.jrpc.jrpc\_util.JsonRpcErrorCode Class Reference

Inheritance diagram for `avalon_sdk.connector.direct.jrpc.jrpc_util.JsonRpcErrorCode`:



Collaboration diagram for `avalon_sdk.connector.direct.jrpc.jrpc_util.JsonRpcErrorCode`:



### Static Public Attributes

- `int SUCCESS = 0`
- `int UNKNOWN_ERROR = 1`
- `int INVALID_PARAMETER = 2`
- `int ACCESS_DENIED = 3`
- `int INVALID_SIGNATURE = 4`
- `int NO_LOOKUP_RESULTS = 5`
- `int UNSUPPORTED_MODE = 6`

### 12.33.1 Detailed Description

JSON RPC error code values:

```
0 - SUCCESS
1 - UNKNOWN_ERROR
2 - INVALID_PARAMETER format or value
3 - ACCESS_DENIED
4 - INVALID_SIGNATURE
5 - NO_LOOKUP_RESULTS no more lookup results remaining
6 - UNSUPPORTED_MODE (e.g. synchronous, asynchronous, poll,
    or notification)
-32768 to -32000 - reserved for pre-defined errors in the JSON RPC spec.
```

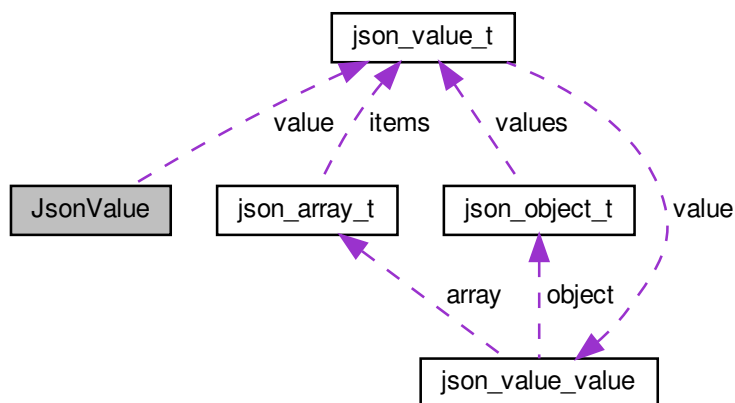
From EEA spec 4.1.1.

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

- `/home/dano/git/avalon/sdk/avalon_sdk/connector/direct/jrpc/jrpc_util.py`

## 12.34 JsonValue Class Reference

Collaboration diagram for JsonValue:



### Public Member Functions

- **JsonValue** ([JSON\\_Value](#) \*value=nullptr)
- **operator JSON\_Value** \* ()
- **operator const JSON\_Value** \* () const

### Public Attributes

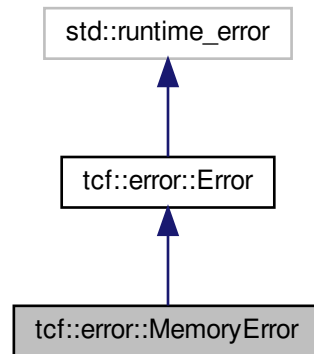
- [JSON\\_Value](#) \* **value**

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

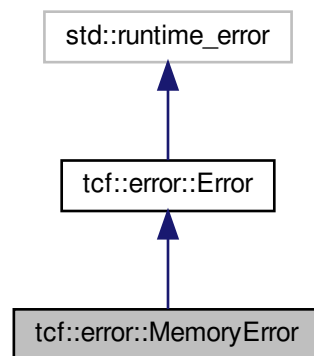
- [/home/dano/git/avalon/common/cpp/jsonvalue.h](#)

## 12.35 tcf::error::MemoryError Class Reference

Inheritance diagram for tcf::error::MemoryError:



Collaboration diagram for tcf::error::MemoryError:



### Public Member Functions

- **MemoryError** (const std::string &msg)

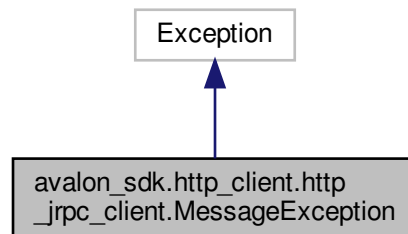
The documentation for this class was generated from the following file:

- /home/dano/git/avalon/common/cpp/[error.h](#)

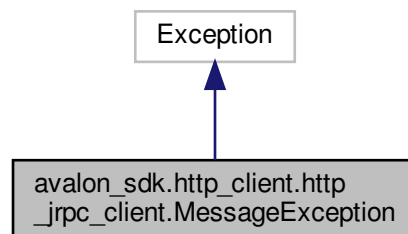


## 12.36 avalon\_sdk.http\_client.http\_rpc\_client.MessageException Class Reference

Inheritance diagram for avalon\_sdk.http\_client.http\_rpc\_client.MessageException:



Collaboration diagram for avalon\_sdk.http\_client.http\_rpc\_client.MessageException:



### 12.36.1 Detailed Description

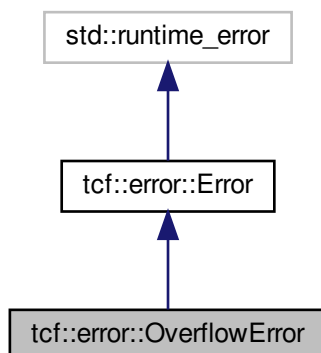
A class to capture communication exceptions when communicating with services.

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

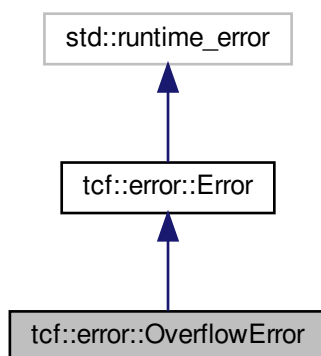
- `/home/dano/git/avalon/sdk/avalon_sdk/http_client/http_rpc_client.py`

## 12.37 tcf::error::OverflowError Class Reference

Inheritance diagram for tcf::error::OverflowError:



Collaboration diagram for tcf::error::OverflowError:



### Public Member Functions

- **OverflowError** (const std::string &msg)

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

- /home/dano/git/avalon/common/cpp/[error.h](#)

## 12.38 tcf::crypto::pkenc::PrivateKey Class Reference

### Public Member Functions

- **PrivateKey** (const [PrivateKey](#) &privateKey)
- **PrivateKey** ([PrivateKey](#) &&privateKey)
- [PrivateKey](#) (const std::string &encoded)
- [~PrivateKey](#) ()
- [PrivateKey](#) & **operator=** (const [PrivateKey](#) &privateKey)
- void [Deserialize](#) (const std::string &encoded)
- void [Generate](#) ()
- [PublicKey](#) [GetPublicKey](#) () const
- std::string [Serialize](#) () const
- [ByteArray](#) [DecryptMessage](#) (const [ByteArray](#) &ct) const

### 12.38.1 Constructor & Destructor Documentation

#### 12.38.1.1 PrivateKey()

```
tcf::crypto::pkenc::PrivateKey::PrivateKey (
    const std::string & encoded )
```

Constructor from encoded string. Throws RuntimeError, ValueError.

#### 12.38.1.2 ~PrivateKey()

```
tcf::crypto::pkenc::PrivateKey::~~PrivateKey ( )
```

[PrivateKey](#) Destructor.

### 12.38.2 Member Function Documentation

#### 12.38.2.1 DecryptMessage()

```
ByteArray tcf::crypto::pkenc::PrivateKey::DecryptMessage (
    const ByteArray & ciphertext ) const
```

Decrypt message with RSA private key and return plaintext. Throws RuntimeError.

#### Parameters

<i>ciphertext</i>	string contains raw binary ciphertext
-------------------	---------------------------------------

**Returns**

ByteArray containing raw binary plaintext

**12.38.2.2 Deserialize()**

```
void tcf::crypto::pkenc::PrivateKey::Deserialize (
    const std::string & encoded )
```

Deserialize RSA Private Key. Throws RunrimeError, ValueError.

**12.38.2.3 Generate()**

```
void tcf::crypto::pkenc::PrivateKey::Generate ( )
```

Generate RSA private key. Throws RuntimeError.

**12.38.2.4 GetPublicKey()**

```
pcrypto::pkenc::PublicKey tcf::crypto::pkenc::PrivateKey::GetPublicKey ( ) const
```

Get Public encryption from [PrivateKey](#). Throws RuntimeError.

**12.38.2.5 operator=()**

```
pcrypto::pkenc::PrivateKey & tcf::crypto::pkenc::PrivateKey::operator= (
    const PrivateKey & privateKey )
```

Assignment operator = overload. Throws RuntimeError.

**12.38.2.6 Serialize()**

```
std::string tcf::crypto::pkenc::PrivateKey::Serialize ( ) const
```

Serialize Private Key. Throws RunrimeError.

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

- [/home/dano/git/avalon/common/cpp/crypto/pkenc\\_private\\_key.h](#)
- [/home/dano/git/avalon/common/cpp/crypto/pkenc\\_private\\_key.cpp](#)

## 12.39 tcf::crypto::sig::PrivateKey Class Reference

### Public Member Functions

- **PrivateKey** (const [PrivateKey](#) &privateKey)
- **PrivateKey** ([PrivateKey](#) &&privateKey)
- [PrivateKey](#) (const std::string &encoded)
- [~PrivateKey](#) ()
- [PrivateKey](#) & **operator=** (const [PrivateKey](#) &privateKey)
- void [Deserialize](#) (const std::string &encoded)
- void [Generate](#) ()
- [PublicKey](#) [GetPublicKey](#) () const
- std::string [Serialize](#) () const
- [ByteArray](#) [SignMessage](#) (const [ByteArray](#) &hashMessage) const

### 12.39.1 Constructor & Destructor Documentation

#### 12.39.1.1 PrivateKey()

```
tcf::crypto::sig::PrivateKey::PrivateKey (
    const std::string & encoded )
```

Constructor from encoded string. Throws RuntimeError, ValueError.

#### 12.39.1.2 ~PrivateKey()

```
tcf::crypto::sig::PrivateKey::~~PrivateKey ( )
```

[PrivateKey](#) Destructor.

### 12.39.2 Member Function Documentation

#### 12.39.2.1 Deserialize()

```
void tcf::crypto::sig::PrivateKey::Deserialize (
    const std::string & encoded )
```

Deserialize ECDSA Private Key. Throws RuntimeError, ValueError.

#### 12.39.2.2 Generate()

```
void tcf::crypto::sig::PrivateKey::Generate ( )
```

Generate ECDSA private key. Throws RuntimeError.

### 12.39.2.3 GetPublicKey()

```
pccrypto::sig::PublicKey tcf::crypto::sig::PrivateKey::GetPublicKey ( ) const
```

Derive Digital Signature public key from private key. Throws `RuntimeError`.

### 12.39.2.4 operator=()

```
pccrypto::sig::PrivateKey & tcf::crypto::sig::PrivateKey::operator= (
    const PrivateKey & privateKey )
```

Assignment operator = overload. Throws `RuntimeError`.

### 12.39.2.5 Serialize()

```
std::string tcf::crypto::sig::PrivateKey::Serialize ( ) const
```

Serialize ECDSA `PrivateKey`. Throws `RuntimeError`.

### 12.39.2.6 SignMessage()

```
ByteArray tcf::crypto::sig::PrivateKey::SignMessage (
    const ByteArray & hashMessage ) const
```

Signs `hashMessage.data()` with ECDSA privkey. It's expected that caller of this function passes the hash value of the original message to this function for signing. Throws `RuntimeError`.

#### Returns

ByteArray containing raw binary signature data

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

- [/home/dano/git/avalon/common/cpp/crypto/sig\\_private\\_key.h](#)
- [/home/dano/git/avalon/common/cpp/crypto/sig\\_private\\_key.cpp](#)

## 12.40 tcf::crypto::sig::PublicKey Class Reference

### Public Member Functions

- `PublicKey ()`
- `PublicKey (const PublicKey &publicKey)`
- `PublicKey (PublicKey &&publicKey)`
- `PublicKey (const PrivateKey &privateKey)`
- `PublicKey (const std::string &encoded)`
- `~PublicKey ()`
- `PublicKey & operator= (const PublicKey &publicKey)`
- `void Deserialize (const std::string &encoded)`
- `std::string Serialize () const`
- `std::string SerializeXYToHex () const`
- `void DeserializeXYFromHex (const std::string &hexXY)`
- `int VerifySignature (const ByteArray &hashMessage, const ByteArray &signature) const`

## 12.40.1 Constructor & Destructor Documentation

### 12.40.1.1 PublicKey() [1/2]

```
tcf::crypto::sig::PublicKey::PublicKey ( )
```

[PublicKey](#) constructor.

### 12.40.1.2 PublicKey() [2/2]

```
tcf::crypto::sig::PublicKey::PublicKey (
    const std::string & encoded )
```

Constructor from encoded string. Throws RuntimeError, ValueError.

### 12.40.1.3 ~PublicKey()

```
tcf::crypto::sig::PublicKey::~~PublicKey ( )
```

[PublicKey](#) Destructor.

## 12.40.2 Member Function Documentation

### 12.40.2.1 Deserialize()

```
void tcf::crypto::sig::PublicKey::Deserialize (
    const std::string & encoded )
```

Deserialize Digital Signature Public Key. Throws RunTime.

### 12.40.2.2 DeserializeXYFromHex()

```
void tcf::crypto::sig::PublicKey::DeserializeXYFromHex (
    const std::string & hexXY )
```

Deserialize EC point (X,Y) hex string. Throws RuntimeError, ValueError.

### 12.40.2.3 operator=()

```
pcrypto::sig::PublicKey & tcf::crypto::sig::PublicKey::operator= (
    const PublicKey & publicKey )
```

Assignment operator = overload. Throws RuntimeError.

#### 12.40.2.4 Serialize()

```
std::string tcf::crypto::sig::PublicKey::Serialize ( ) const
```

Serialize Digital Signature Public Key. Throws RuntimeError.

#### 12.40.2.5 SerializeXYToHex()

```
std::string tcf::crypto::sig::PublicKey::SerializeXYToHex ( ) const
```

Serialize EC point (X,Y) to a hexadecimal string. Throws RuntimeError.

#### 12.40.2.6 VerifySignature()

```
int tcf::crypto::sig::PublicKey::VerifySignature (
    const ByteArray & hashMessage,
    const ByteArray & signature ) const
```

Verifies ECDSA signature of message. It's expected that the caller of this function passes a hash value of the original message.

##### Parameters

<i>signature</i>	ByteArray contains raw binary signature data
------------------	--

##### Returns

1 if signature is valid, 0 if signature is invalid, and -1 if there is an internal error.

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

- [/home/dano/git/avalon/common/cpp/crypto/sig\\_public\\_key.h](#)
- [/home/dano/git/avalon/common/cpp/crypto/sig\\_public\\_key.cpp](#)

## 12.41 tcf::crypto::pkenc::PublicKey Class Reference

### Public Member Functions

- [PublicKey](#) ()
- **PublicKey** (const [PublicKey](#) &publicKey)
- **PublicKey** ([PublicKey](#) &&publicKey)
- **PublicKey** (const [PrivateKey](#) &privateKey)
- [PublicKey](#) (const std::string &encoded)
- [~PublicKey](#) ()
- [PublicKey](#) & [operator=](#) (const [PublicKey](#) &publicKey)
- void [Deserialize](#) (const std::string &encoded)
- std::string [Serialize](#) () const
- [ByteArray](#) [EncryptMessage](#) (const [ByteArray](#) &message) const



## 12.41.1 Constructor & Destructor Documentation

### 12.41.1.1 PublicKey() [1/2]

```
tcf::crypto::pkenc::PublicKey::PublicKey ( )
```

[PublicKey](#) constructor.

### 12.41.1.2 PublicKey() [2/2]

```
tcf::crypto::pkenc::PublicKey::PublicKey (
    const std::string & encoded )
```

Constructor from encoded string. Throws RuntimeError, ValueError.

### 12.41.1.3 ~PublicKey()

```
tcf::crypto::pkenc::PublicKey::~~PublicKey ( )
```

[PublicKey](#) destructor.

## 12.41.2 Member Function Documentation

### 12.41.2.1 Deserialize()

```
void tcf::crypto::pkenc::PublicKey::Deserialize (
    const std::string & encoded )
```

Deserialize Public Key. Throws RuntimeError, ValueError.

### 12.41.2.2 EncryptMessage()

```
ByteArray tcf::crypto::pkenc::PublicKey::EncryptMessage (
    const ByteArray & message ) const
```

Encrypt message with RSA public key and return ciphertext. Throws RuntimeError.

#### Parameters

<i>message</i>	ByteArray containing raw binary plaintext
----------------	---

**Returns**

ByteArray containing raw binary ciphertext

**12.41.2.3 operator=()**

```
pcrypto::pkenc::PublicKey & tcf::crypto::pkenc::PublicKey::operator= (
    const PublicKey & publicKey )
```

Assignment operator = overload. Throws RuntimeError.

**12.41.2.4 Serialize()**

```
std::string tcf::crypto::pkenc::PublicKey::Serialize ( ) const
```

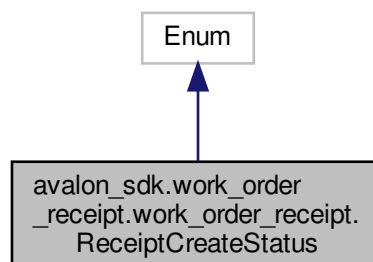
Serialize Public Key. Throws RuntimeError.

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

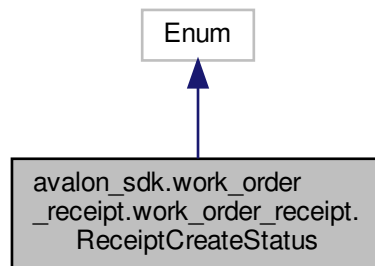
- [/home/dano/git/avalon/common/cpp/crypto/pkenc\\_public\\_key.h](#)
- [/home/dano/git/avalon/common/cpp/crypto/pkenc\\_public\\_key.cpp](#)

## 12.42 avalon\_sdk.work\_order\_receipt.work\_order\_receipt.ReceiptCreateStatus **Class Reference**

Inheritance diagram for avalon\_sdk.work\_order\_receipt.work\_order\_receipt.ReceiptCreateStatus:



Collaboration diagram for avalon\_sdk.work\_order\_receipt.work\_order\_receipt.ReceiptCreateStatus:



### Static Public Attributes

- int **PENDING** = 0
- int **COMPLETED** = 1
- int **PROCESSED** = 2
- int **FAILED** = 3
- int **REJECTED** = 4

#### 12.42.1 Detailed Description

Receipt creation status values:

0 - PENDING. The work order is waiting to be processed by the worker  
1 - COMPLETED. The worker processed the Work Order and no more worker updates are expected  
2 - PROCESSED. The worker processed the Work Order, but additional worker updates are expected, e.g. oracle notifications  
3 - FAILED. The Work Order processing failed, e.g. by the worker service because of an invalid workerId  
4 - REJECTED. The Work Order is rejected by the smart contract, e.g. invalid workerServiceId  
5 to 254 - reserved  
255 - indicates any status  
>255 - application-specific values

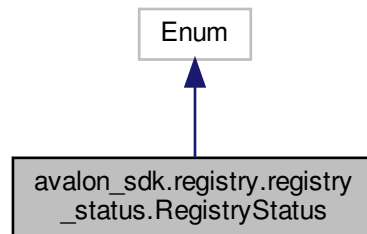
Defined in EEA spec 7.1.

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

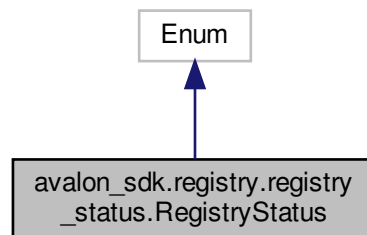
- /home/dano/git/avalon/sdk/avalon\_sdk/work\_order\_receipt/work\_order\_receipt.py

## 12.43 avalon\_sdk.registry.registry\_status.RegistryStatus Class Reference

Inheritance diagram for avalon\_sdk.registry.registry\_status.RegistryStatus:



Collaboration diagram for avalon\_sdk.registry.registry\_status.RegistryStatus:



### Static Public Attributes

- int **ACTIVE** = 1
- int **OFF\_LINE** = 2
- int **DECOMMISSIONED** = 3

### 12.43.1 Detailed Description

Worker registry status values:  
1 - registry is ACTIVE  
2 - registry is temporarily OFF\_LINE  
3 - registry is DECOMMISSIONED

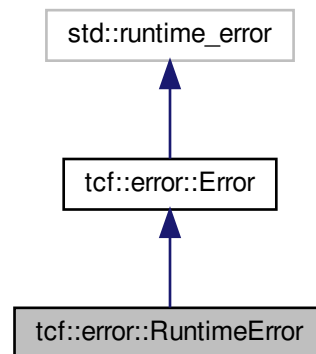
From EEA spec 5.2.

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

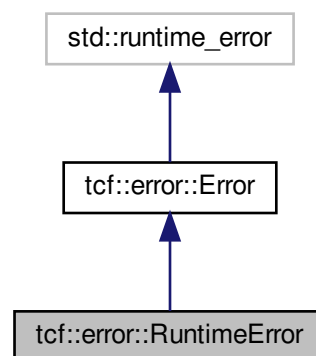
- `/home/dano/git/avalon/sdk/avalon_sdk/registry/registry_status.py`

## 12.44 tcf::error::RuntimeError Class Reference

Inheritance diagram for tcf::error::RuntimeError:



Collaboration diagram for tcf::error::RuntimeError:



### Public Member Functions

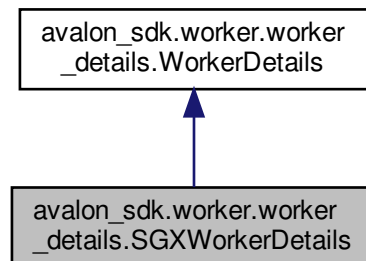
- **RuntimeError** (const std::string &msg)

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

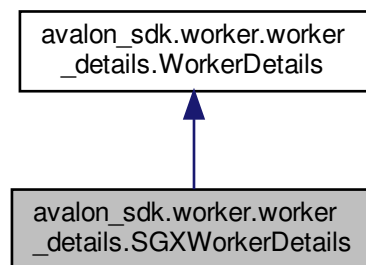
- `/home/dano/git/avalon/common/cpp/error.h`

## 12.45 avalon\_sdk.worker.worker\_details.SGXWorkerDetails Class Reference

Inheritance diagram for avalon\_sdk.worker.worker\_details.SGXWorkerDetails:



Collaboration diagram for avalon\_sdk.worker.worker\_details.SGXWorkerDetails:



### Public Member Functions

- `def __init__ (self)`
- `def load\_worker (self, worker_data)`

### Public Attributes

- `verification_key`
- `extended_measurements`
- `proof_data_type`
- `proof_data`
- `encryption_key`
- `encryption_key_nonce`
- `encryption_key_signature`

- `enclave_certificate`
- `worker_id`
- `hashing_algorithm`
- `signing_algorithm`
- `key_encryption_algorithm`
- `data_encryption_algorithm`

### 12.45.1 Detailed Description

Contains Intel SGX TEE worker type data.

### 12.45.2 Member Function Documentation

#### 12.45.2.1 `load_worker()`

```
def avalon_sdk.worker.worker_details.SGXWorkerDetails.load_worker (
    self,
    worker_data )
```

Load member variables of this class  
based on worker-retrieved details.

Parameters:  
    `worker_data` Worker Data to load into the class

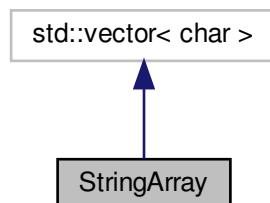
The documentation for this class was generated from the following file:

- `/home/dano/git/avalon/sdk/avalon_sdk/worker/worker_details.py`

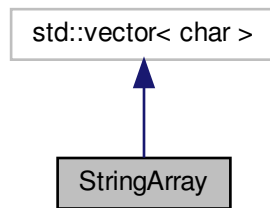
## 12.46 StringArray Class Reference

```
#include <types.h>
```

Inheritance diagram for StringArray:



Collaboration diagram for StringArray:



### Public Member Functions

- **StringArray** (const std::string &value)
- **StringArray** (const size\_t size)
- void **assign** (const std::string &value)
- std::string **str** ()

#### 12.46.1 Detailed Description

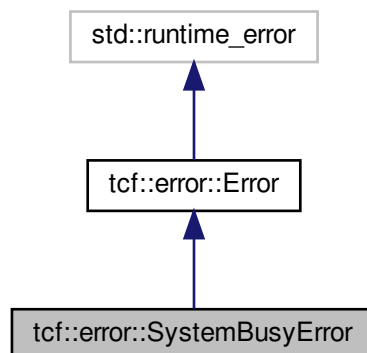
Vector type for containing printable characters.

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

- </home/dano/git/avalon/common/cpp/types.h>

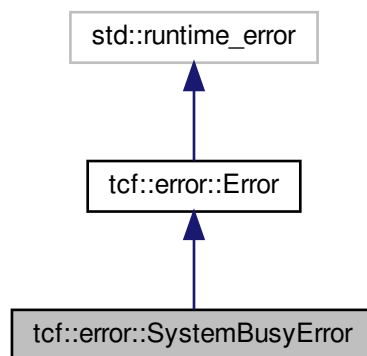
## 12.47 tcf::error::SystemBusyError Class Reference

Inheritance diagram for tcf::error::SystemBusyError:





Collaboration diagram for tcf::error::SystemBusyError:



### Public Member Functions

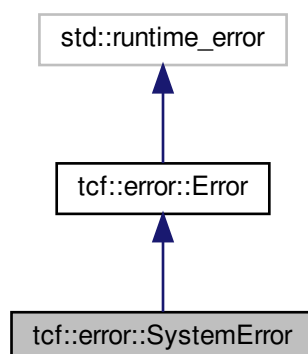
- **SystemBusyError** (const std::string &msg)

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

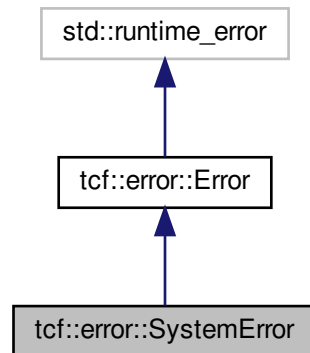
- /home/dano/git/avalon/common/cpp/[error.h](#)

## 12.48 tcf::error::SystemError Class Reference

Inheritance diagram for tcf::error::SystemError:



Collaboration diagram for `tcf::error::SystemError`:



### Public Member Functions

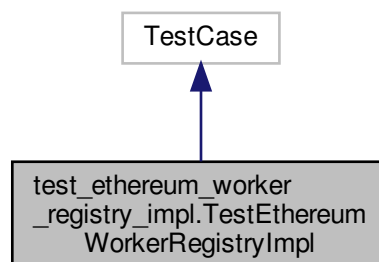
- **SystemError** (const std::string &msg)

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

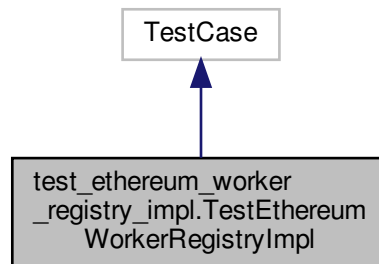
- </home/dano/git/avalon/common/cpp/error.h>

## 12.49 test\_ethereum\_worker\_registry\_impl.TestEthereumWorkerRegistryImpl Class Reference

Inheritance diagram for `test_ethereum_worker_registry_impl.TestEthereumWorkerRegistryImpl`:



Collaboration diagram for test\_ethereum\_worker\_registry\_impl.TestEthereumWorkerRegistryImpl:



### Public Member Functions

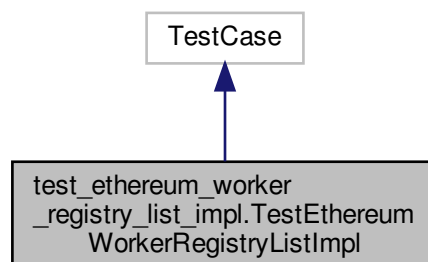
- def **\_\_init\_\_** (self, config\_file)
- def **test\_worker\_register** (self)
- def **test\_worker\_set\_status** (self)
- def **test\_worker\_update** (self)
- def **test\_worker\_lookup** (self)
- def **test\_worker\_retrieve** (self)
- def **test\_worker\_lookup\_next** (self)

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

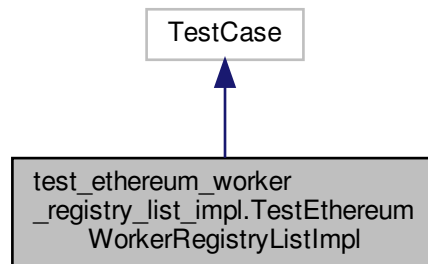
- /home/dano/git/avalon/sdk/avalon\_sdk/connector/blockchains/ethereum/unit\_tests/test\_ethereum\_worker\_registry\_impl.py↔

## 12.50 test\_ethereum\_worker\_registry\_list\_impl.TestEthereumWorkerRegistryListImpl Class Reference

Inheritance diagram for test\_ethereum\_worker\_registry\_list\_impl.TestEthereumWorkerRegistryListImpl:



Collaboration diagram for `test_ethereum_worker_registry_list_impl.TestEthereumWorkerRegistryListImpl`:



### Public Member Functions

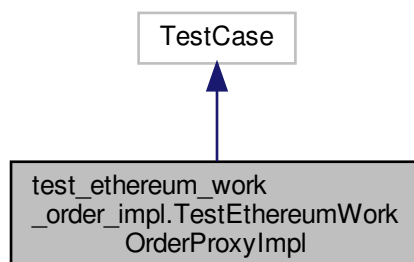
- `def __init__(self, config_file)`
- `def test_registry_add(self)`
- `def test_registry_update(self)`
- `def test_registry_set_status(self)`
- `def test_registry_lookup(self)`
- `def test_registry_retrieve(self)`
- `def test_registry_lookup_next(self)`

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

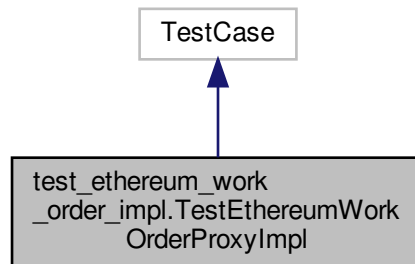
- `/home/dano/git/avalon/sdk/avalon_sdk/connector/blockchains/ethereum/unit_tests/test_ethereum_worker_registry_list_impl.py`

## 12.51 test\_ethereum\_work\_order\_impl.TestEthereumWorkOrderProxyImpl Class Reference

Inheritance diagram for `test_ethereum_work_order_impl.TestEthereumWorkOrderProxyImpl`:



Collaboration diagram for test\_ethereum\_work\_order\_impl.TestEthereumWorkOrderProxyImpl:



## Public Member Functions

- def `__init__` (self, config\_file)
- def `test_work_order_submit_positive` (self)
- def `test_work_order_submit_mismatch` (self)
- def `test_work_order_get_result` (self)
- def `test_work_order_complete` (self)
- def `test_work_order_complete_error` (self)
- def `test_is_wo_id_in_event_positive` (self)
- def `test_is_wo_id_in_event_wo_id_not_matched` (self)
- def `test_is_wo_id_in_event_error_result` (self)
- def `test_is_wo_id_in_event_no_wo_id` (self)
- def `test_is_valid_work_order_json` (self)

### 12.51.1 Member Function Documentation

#### 12.51.1.1 test\_is\_wo\_id\_in\_event\_error\_result()

```
def test_ethereum_work_order_impl.TestEthereumWorkOrderProxyImpl.test_is_wo_id_in_event_error_result (
    self )
```

This case mocks an event and verifies the `wo_id_in_event` function for a positive result. The event has an error response from work order execution.

#### 12.51.1.2 test\_is\_wo\_id\_in\_event\_positive()

```
def test_ethereum_work_order_impl.TestEthereumWorkOrderProxyImpl.test_is_wo_id_in_event_↵  
positive (   
    self )
```

This case mocks an event and verifies the wo\_id\_in\_event function for a positive result.

#### 12.51.1.3 test\_is\_wo\_id\_in\_event\_wo\_id\_not\_matched()

```
def test_ethereum_work_order_impl.TestEthereumWorkOrderProxyImpl.test_is_wo_id_in_event_↵  
id_not_matched (   
    self )
```

This case mocks an event and verifies the wo\_id\_in\_event function for a negative result. The wo\_id does not match.

#### 12.51.1.4 test\_work\_order\_complete()

```
def test_ethereum_work_order_impl.TestEthereumWorkOrderProxyImpl.test_work_order_complete (   
    self )
```

This function verifies if work order complete function succeeds when the in work order execution is done.

#### 12.51.1.5 test\_work\_order\_complete\_error()

```
def test_ethereum_work_order_impl.TestEthereumWorkOrderProxyImpl.test_work_order_complete_↵  
error (   
    self )
```

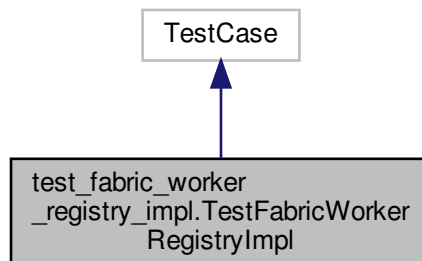
This function verifies if work order complete function succeeds when there is an error in work order execution.

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

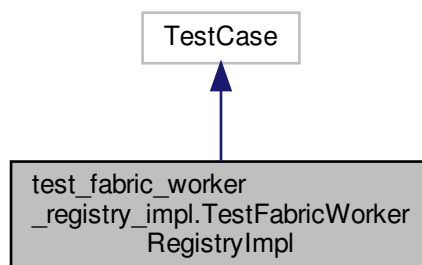
- /home/dano/git/avalon/sdk/avalon\_sdk/connector/blockchains/ethereum/unit\_tests/test\_ethereum\_work\_↵  
order\_impl.py

## 12.52 test\_fabric\_worker\_registry\_impl.TestFabricWorkerRegistryImpl Class Reference

Inheritance diagram for test\_fabric\_worker\_registry\_impl.TestFabricWorkerRegistryImpl:



Collaboration diagram for test\_fabric\_worker\_registry\_impl.TestFabricWorkerRegistryImpl:



### Public Member Functions

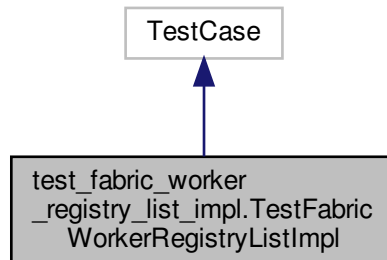
- def **\_\_init\_\_** (self, config\_file)
- def **test\_worker\_register** (self)
- def **test\_worker\_set\_status** (self)
- def **test\_worker\_update** (self)
- def **test\_worker\_lookup** (self)
- def **test\_worker\_retrieve** (self)
- def **test\_worker\_lookup\_next** (self)

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

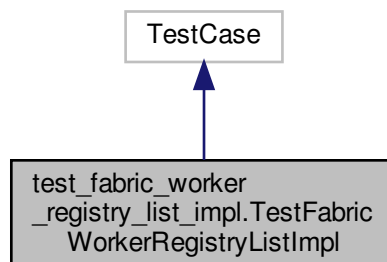
- /home/dano/git/avalon/sdk/avalon\_sdk/connector/blockchains/fabric/unit\_tests/test\_fabric\_worker\_registry\_↔\_impl.py

## 12.53 test\_fabric\_worker\_registry\_list\_impl.TestFabricWorkerRegistryListImpl Class Reference

Inheritance diagram for test\_fabric\_worker\_registry\_list\_impl.TestFabricWorkerRegistryListImpl:



Collaboration diagram for test\_fabric\_worker\_registry\_list\_impl.TestFabricWorkerRegistryListImpl:



### Public Member Functions

- def **\_\_init\_\_** (self, config\_file)
- def **test\_registry\_add** (self)
- def **test\_registry\_update** (self)
- def **test\_registry\_set\_status** (self)
- def **test\_registry\_lookup** (self)
- def **test\_registry\_retrieve** (self)
- def **test\_registry\_lookup\_next** (self)

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

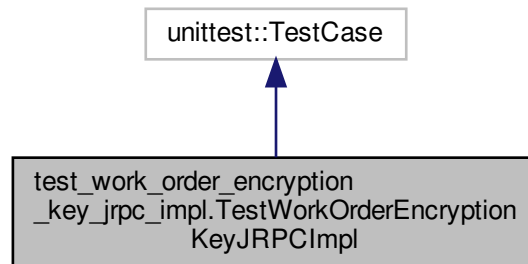
- /home/dano/git/avalon/sdk/avalon\_sdk/connector/blockchains/fabric/unit\_tests/test\_fabric\_worker\_registry\_list\_impl.py↔



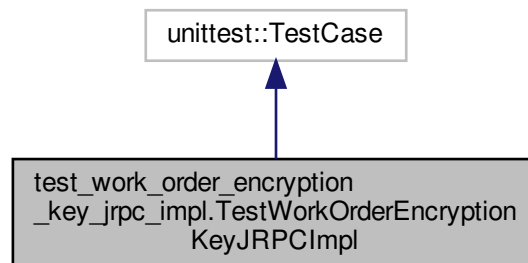


## 12.55 test\_work\_order\_encryption\_key\_jrpc\_impl.TestWorkOrderEncryptionKeyJRPCImpl Class Reference

Inheritance diagram for test\_work\_order\_encryption\_key\_jrpc\_impl.TestWorkOrderEncryptionKeyJRPCImpl:



Collaboration diagram for test\_work\_order\_encryption\_key\_jrpc\_impl.TestWorkOrderEncryptionKeyJRPCImpl:



### Public Member Functions

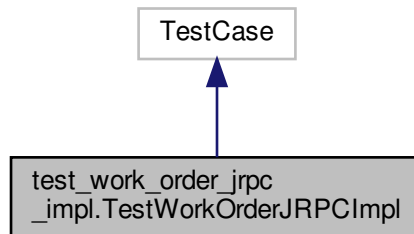
- def **\_\_init\_\_** (self, config\_file)
- def **test\_encryption\_key\_get** (self)
- def **test\_encryption\_key\_set** (self)

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

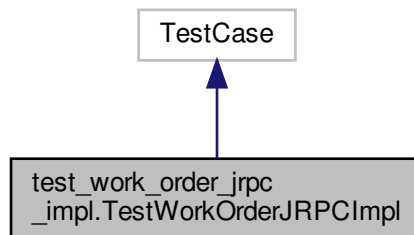
- /home/dano/git/avalon/sdk/avalon\_sdk/connector/direct/jrpc/unit\_tests/test\_work\_order\_encryption\_key\_jrpc\_impl.py

## 12.56 test\_work\_order\_jrpc\_impl.TestWorkOrderJRPCImpl Class Reference

Inheritance diagram for test\_work\_order\_jrpc\_impl.TestWorkOrderJRPCImpl:



Collaboration diagram for test\_work\_order\_jrpc\_impl.TestWorkOrderJRPCImpl:



### Public Member Functions

- def `__init__` (self, config\_file)
- def `test_work_order_submit` (self)
- def `test_work_order_get_result` (self)

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

- `/home/dano/git/avalon/sdk/avalon_sdk/connector/direct/jrpc/unit_tests/test_work_order_jrpc_impl.py`

## 12.57 tcf::utility::Timer Class Reference

### Public Member Functions

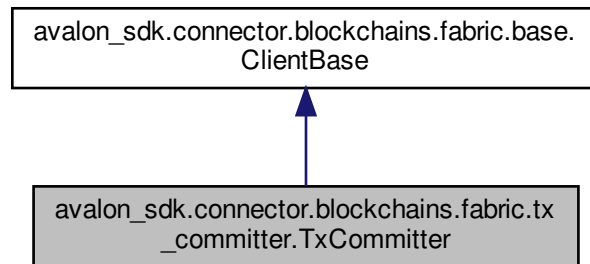
- **Timer** (const std::string &key)

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

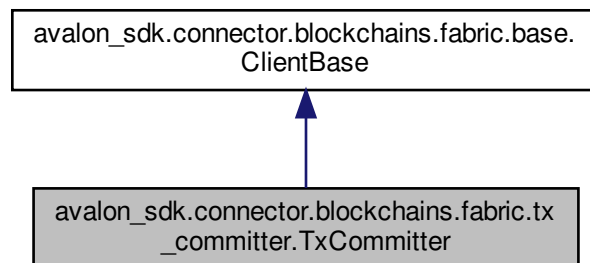
- `/home/dano/git/avalon/common/cpp/timer.h`

## 12.58 `avalon_sdk.connector.blockchains.fabric.tx_committer.TxCommitter` Class Reference

Inheritance diagram for `avalon_sdk.connector.blockchains.fabric.tx_committer.TxCommitter`:



Collaboration diagram for `avalon_sdk.connector.blockchains.fabric.tx_committer.TxCommitter`:



### Public Member Functions

- def `cc_invoke` (self, args, cc\_name, fcn, cc\_version, queryonly=False)
- def `cc_query` (self, args, cc\_name, fcn)

### Additional Inherited Members

#### 12.58.1 Detailed Description

Utility class to invoke Fabric chain code and query chain code.

## 12.58.2 Member Function Documentation

### 12.58.2.1 cc\_invoke()

```
def avalon_sdk.connector.blockchains.fabric.tx_committer.TxCommitter.cc_invoke (
    self,
    args,
    cc_name,
    fcn,
    cc_version,
    queryonly = False )
```

Invoke a chaincode method.

Parameters:

args        JSON RPC serialized data used as the  
             sole parameter to invoke the chaincode  
cc\_name     chaincode name  
fcn         chaincode function name to be invoked  
cc\_version   chaincode version to be used  
queryonly   If the invocation does not result in ledger change,  
             queryonly should be set to True.  
             If the invocation does result in ledger change, it should  
             be set to False.

### 12.58.2.2 cc\_query()

```
def avalon_sdk.connector.blockchains.fabric.tx_committer.TxCommitter.cc_query (
    self,
    args,
    cc_name,
    fcn )
```

Invoke a chaincode query method. If there is no query method from the chaincode, then this will fail.

Parameters:

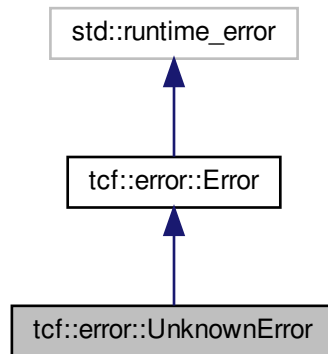
args        Array of the strings used as the parameters to query method  
cc\_name     Chaincode name  
fcn         Chaincode function name

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

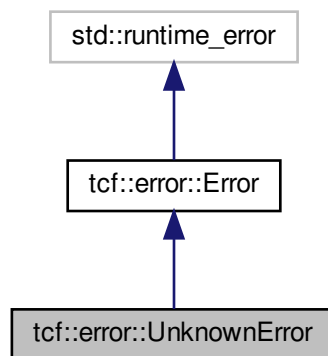
- /home/dano/git/avalon/sdk/avalon\_sdk/connector/blockchains/fabric/tx\_committer.py

## 12.59 tcf::error::UnknownError Class Reference

Inheritance diagram for tcf::error::UnknownError:



Collaboration diagram for tcf::error::UnknownError:



### Public Member Functions

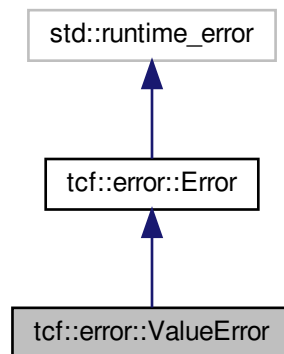
- **UnknownError** (const std::string &msg)

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

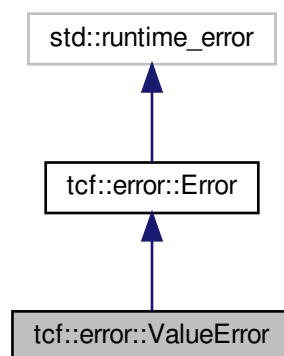
- `/home/dano/git/avalon/common/cpp/error.h`

## 12.60 tcf::error::ValueError Class Reference

Inheritance diagram for tcf::error::ValueError:



Collaboration diagram for tcf::error::ValueError:



### Public Member Functions

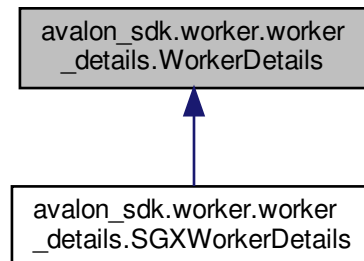
- **ValueError** (const std::string &msg)

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

- /home/dano/git/avalon/common/cpp/[error.h](#)

## 12.61 `avalon_sdk.worker.worker_details.WorkerDetails` Class Reference

Inheritance diagram for `avalon_sdk.worker.worker_details.WorkerDetails`:



### Public Member Functions

- `def __init__(self)`
- `def validate\_worker\_details(self, details)`

### Public Attributes

- `work_order_sync_uri`
- `work_order_async_uri`
- `work_order_pull_uri`
- `work_order_notify_uri`
- `receipt_invocation_uri`
- `work_order_invocation_address`
- `receipt_invocation_address`
- `from_address`
- `hashing_algorithm`
- `signing_algorithm`
- `key_encryption_algorithm`
- `data_encryption_algorithm`
- `work_order_payload_formats`

### 12.61.1 Detailed Description

Class to store the worker details

### 12.61.2 Constructor & Destructor Documentation



### 12.61.2.1 `__init__()`

```
def avalon_sdk.worker.worker_details.WorkerDetails.__init__ (
    self )
```

Set the member variables of this class with default values as per the EEA Spec.

## 12.61.3 Member Function Documentation

### 12.61.3.1 `validate_worker_details()`

```
def avalon_sdk.worker.worker_details.WorkerDetails.validate_worker_details (
    self,
    details )
```

Validate the details field of a worker.

Parameters:

Details is json formatted string

Returns:

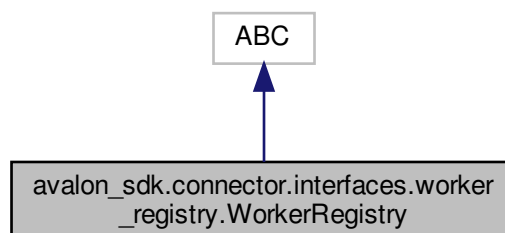
None on success and error string on failure

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

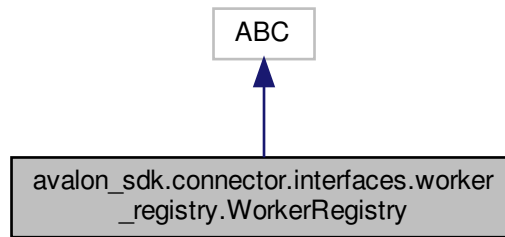
- `/home/dano/git/avalon/sdk/avalon_sdk/worker/worker_details.py`

## 12.62 `avalon_sdk.connector.interfaces.worker_registry.WorkerRegistry` Class Reference

Inheritance diagram for `avalon_sdk.connector.interfaces.worker_registry.WorkerRegistry`:



Collaboration diagram for `avalon_sdk.connector.interfaces.worker_registry.WorkerRegistry`:



## Public Member Functions

- `def __init__(self)`
- `def worker_retrieve(self, worker_id, id=None)`
- `def worker_lookup(self, worker_type, organization_id, application_type_id, id=None)`
- `def worker_lookup_next(self, worker_type, organization_id, application_type_id, lookup_tag, id=None)`
- `def worker_register(self, worker_id, worker_type, organization_id, application_type_ids, details, id=None)`
- `def worker_update(self, worker_id, details, id=None)`
- `def worker_set_status(self, worker_id, status, id=None)`

### 12.62.1 Detailed Description

This class is an abstract base class containing abstract APIs which can be called from client to manage workers.

### 12.62.2 Member Function Documentation

#### 12.62.2.1 `worker_lookup()`

```
def avalon_sdk.connector.interfaces.worker_registry.WorkerRegistry.worker_lookup (
    self,
    worker_type,
    organization_id,
    application_type_id,
    id = None )
```

Lookup a worker identified worker\_type, organization, and application\_id.

All fields are optional and, if present, condition should match for all fields. If none are passed it should return all workers.

If the list is too large to fit into a single response (the maximum number of entries in a single response is implementation specific), the smart contract should return the first batch of the results and provide a lookupTag that can be used by the caller to retrieve the next batch by calling worker\_lookup\_next.

Parameters:

worker\_type Optional characteristic of workers for which you may wish to search  
 organization\_id Optional organization ID that can be used to search for one or more workers that belong to this organization  
 application\_id Optional application type ID that is supported by the worker  
 id Optional JSON RPC request ID

Returns:

Tuple containing workers count, lookup tag, and list of worker IDs:

total\_count Total number of entries matching a specified lookup criteria. If this number is larger than the size of the IDs array, the caller should use lookupTag to call worker\_lookup\_next to retrieve the rest of the IDs

lookup\_tag Optional parameter. If it is returned, it means that there are more matching worker IDs, which can then be retrieved by calling function worker\_lookup\_next with this tag as an input parameter

ids Array of the worker IDs that match the input parameters

On error returns None.

### 12.62.2.2 worker\_lookup\_next()

```
def avalon_sdk.connector.interfaces.worker_registry.WorkerRegistry.worker_lookup_next (
    self,
    worker_type,
    organization_id,
    application_type_id,
    lookup_tag,
    id = None )
```

Retrieve additional worker lookup results after calling worker\_lookup.

Parameters:

worker\_type Characteristic of Workers for which you may wish to search.  
 organization\_id Organization ID to which a Worker belongs  
 application\_id Optional application type ID that is supported by the worker  
 lookup\_tag is returned by a previous call to either this function or to worker\_lookup  
 id Optional Optional JSON RPC request ID

Returns:

Tuple containing the following:

total\_count Total number of entries matching this lookup criteria. If this number is larger than the number

of IDs returned so far, the caller should use `lookupTag` to call `worker_lookup_next` to retrieve the rest of the IDs

`new_lookup_tag` Optional parameter. If it is returned, it means that there are more matching worker IDs that can be retrieved by calling this function again with this tag as an input parameter

`ids` Array of the worker IDs that match the input parameters

On error returns `None`.

### 12.62.2.3 worker\_register()

```
def avalon_sdk.connector.interfaces.worker_registry.WorkerRegistry.worker_register (
    self,
    worker_id,
    worker_type,
    organization_id,
    application_type_ids,
    details,
    id = None )
```

Register a new worker with details of the worker.

Parameters:

`worker_id` Worker ID value. E.g., a Fabric address or Ethereum DID

`worker_type` Type of Worker. Currently defined types are:

- \* "TEE-SGX": an Intel SGX Trusted Execution Environment
- \* "MPC": Multi-Party Compute
- \* "ZK": Zero-Knowledge

`organization_id` Optional parameter representing the organization that hosts the Worker, e.g. a bank in the consortium or anonymous entity

`application_ids` Optional parameter that defines application types supported by the Worker

`details` Detailed information about the worker in JSON RPC format as defined in <https://entethalliance.github.io/trusted-computing/spec.html#common-data-for-all-worker-types>

`id` Optional Optional JSON RPC request ID

Returns:

`ContractResponse.SUCCESS` on success or `ContractResponse.ERROR` on error.

### 12.62.2.4 worker\_retrieve()

```
def avalon_sdk.connector.interfaces.worker_registry.WorkerRegistry.worker_retrieve (
    self,
    worker_id,
    id = None )
```

Retrieve worker identified by worker ID.

Parameters:

`worker_id` Worker ID of the registry whose details are requested  
`id` Optional JSON RPC request ID

Returns:

Tuple containing worker status (defined in `worker_set_status`),  
 worker type, organization ID, list of application IDs, and worker  
 details (JSON RPC string).

On error returns None.

### 12.62.2.5 worker\_set\_status()

```
def avalon_sdk.connector.interfaces.worker_registry.WorkerRegistry.worker_set_status (
    self,
    worker_id,
    status,
    id = None )
```

Set the registry status identified by worker ID

Parameters:

`worker_id` Worker ID value. E.g., a Fabric address  
 or Ethereum DID  
`status` Worker status. The currently defined values are:  
 1 - worker is active  
 2 - worker is temporarily "off-line"  
 3 - worker is decommissioned  
 4 - worker is compromised  
`id` Optional Optional JSON RPC request ID

Returns:

`ContractResponse.SUCCESS` on success  
 or `ContractResponse.ERROR` on error.

### 12.62.2.6 worker\_update()

```
def avalon_sdk.connector.interfaces.worker_registry.WorkerRegistry.worker_update (
    self,
    worker_id,
    details,
    id = None )
```

Update a worker with details data.

Parameters:

`worker_id` Worker ID  
`details` Detailed information about the worker in JSON format  
`id` Optional Optional JSON RPC request ID

Returns:

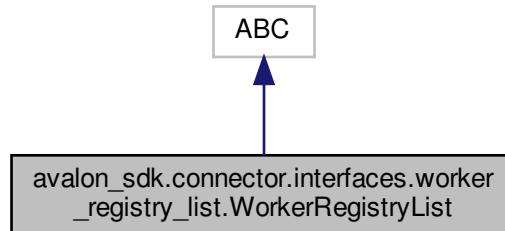
`ContractResponse.SUCCESS` on success  
 or `ContractResponse.ERROR` on error.

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

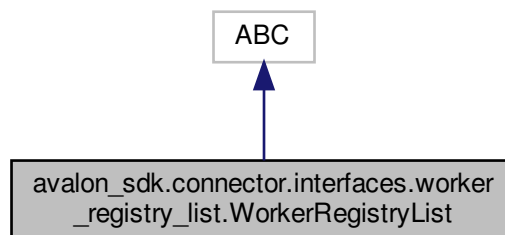
- `/home/dano/git/avalon/sdk/avalon_sdk/connector/interfaces/worker_registry.py`

## 12.63 `avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList` Class Reference

Inheritance diagram for `avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList`:



Collaboration diagram for `avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList`:



### Public Member Functions

- `def __init__(self)`
- `def registry_retrieve(self, organization_id)`
- `def registry_lookup(self, application_type_id)`
- `def registry_lookup_next(self, application_type_id, lookup_tag)`
- `def registry_add(self, organization_id, uri, sc_addr, application_type_ids)`
- `def registry_update(self, organization_id, uri, sc_addr, application_type_ids)`
- `def registry_set_status(self, organization_id, status)`

### 12.63.1 Detailed Description

This is an abstract base class to read/write the worker registries, which can be called by client.

## 12.63.2 Member Function Documentation

### 12.63.2.1 registry\_add()

```
def avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList.registry_add (
    self,
    organization_id,
    uri,
    sc_addr,
    application_type_ids )
```

Add a new registry.

Parameters:

`organization_id` bytes[] identifies organization that hosts the registry, e.g. a bank in the consortium or an anonymous entity

`uri` String defines a URI for this registry that supports the Off-Chain Worker Registry JSON RPC API.

`sc_addr` bytes[] defines an smart contract address that runs the Worker Registry Smart Contract API smart contract for this registry

`application_type_ids` []bytes[] is an optional parameter that defines application types supported by the worker managed by the registry

Returns:

Transaction receipt on success or None on error.

### 12.63.2.2 registry\_lookup()

```
def avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList.registry_lookup (
    self,
    application_type_id )
```

Registry Lookup identified by application type ID

Parameters:

`application_type_id` Application type ID to lookup in the registry

Returns:

Tuple containing `totalCount`, `lookupTag`, and `ids` on success:

`totalCount` Total number of entries matching a specified lookup criteria. If this number is larger than the size of the `ids` array, the caller should use the `lookupTag` to call `registry_lookup_next` to retrieve the rest of the IDs

`lookupTag` Optional parameter. If it is returned, it means that there are more matching registry IDs that can be retrieved by calling the function `registry_lookup_next` with this tag as an input parameter.

`ids` Array of the registry organization ids that match the input parameters.

Returns None on error.

### 12.63.2.3 registry\_lookup\_next()

```
def avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList.registry_lookup_next (
    self,
    application_type_id,
    lookup_tag )
```

This function is called to retrieve additional results of the Registry lookup initiated by the registry\_lookup call.

**Parameters:**

application\_type\_id      Application type that has to be supported by the workers retrieved  
lookup\_tag                Returned by a previous call to either this function or to registry\_lookup

**Returns:**

Outputs tuple on success containing the following:

total\_count      Total number of entries matching the lookup criteria. If this number is larger than the number of IDs returned so far, the caller should use lookup\_tag to call registry\_lookup\_next to retrieve the rest of the IDs  
new\_lookup\_tag    Optional parameter. If it is returned, it means that there are more matching registry IDs that can be retrieved by calling this function again with this tag as an input parameter  
ids                Array of the registry IDs that match the input parameters

Returns None on error.

### 12.63.2.4 registry\_retrieve()

```
def avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList.registry_retrieve (
    self,
    organization_id )
```

Retrieve registry information identified by the organization ID.

**Parameters:**

organization\_id      Organization ID to lookup

**Returns:**

Tuple containing following on success:

uri                      String defines a URI for this registry that supports the Off-Chain Worker Registry JSON RPC API. It will be None for the proxy model  
sc\_addr                 smart contract address for worker registry  
application\_type\_ids    List of application ids (array of byte[])  
status                   Status of the registry

Returns None on error.



## 12.63.2.5 registry\_set\_status()

```
def avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList.registry_set_status (
    self,
    organization_id,
    status )
```

Set registry status.

Parameters:

organization\_id bytes[] identifies organization that hosts the registry, e.g. a bank in the consortium or an anonymous entity

status Defines the registry status to set.

The currently defined values are:

- 1 - the registry is active
- 2 - the registry is temporarily "off-line"
- 3 - the registry is decommissioned

Returns:

Transaction receipt on success or None on error.

## 12.63.2.6 registry\_update()

```
def avalon_sdk.connector.interfaces.worker_registry_list.WorkerRegistryList.registry_update (
    self,
    organization_id,
    uri,
    sc_addr,
    application_type_ids )
```

Update a registry.

Parameters:

organization\_id bytes[] identifies organization that hosts the registry, e.g. a bank in the consortium or an anonymous entity

uri string defines a URI for this registry that supports the Off-Chain Worker Registry JSON RPC API

sc\_addr bytes[] defines an smart contract address that runs a Worker Registry Smart Contract API smart contract for this registry

application\_type\_ids []bytes[] is an optional parameter that defines application types supported by the worker managed by the registry

Returns:

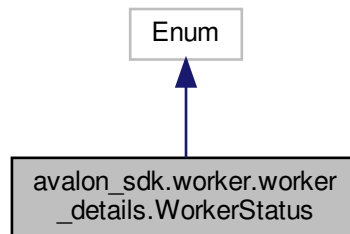
Transaction receipt on success or None on error.

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

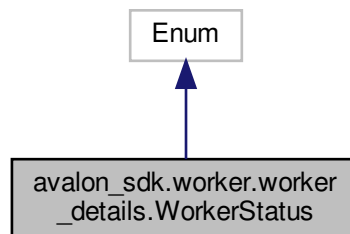
- /home/dano/git/avalon/sdk/avalon\_sdk/connector/interfaces/worker\_registry\_list.py

## 12.64 avalon\_sdk.worker.worker\_details.WorkerStatus Class Reference

Inheritance diagram for avalon\_sdk.worker.worker\_details.WorkerStatus:



Collaboration diagram for avalon\_sdk.worker.worker\_details.WorkerStatus:



### Static Public Attributes

- int **ACTIVE** = 1
- int **OFF\_LINE** = 2
- int **DECOMMISSIONED** = 3
- int **COMPROMISED** = 4

### 12.64.1 Detailed Description

Worker status values:  
1 - worker is ACTIVE  
2 - worker is temporarily OFF\_LINE  
3 - worker is DECOMMISSIONED  
4 - worker is COMPROMISED

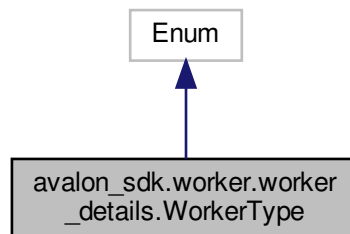
From EEA spec 5.2.

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

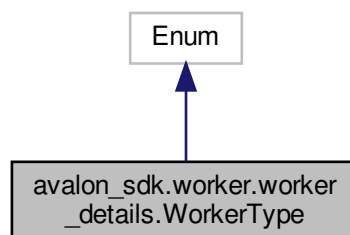
- `/home/dano/git/avalon/sdk/avalon_sdk/worker/worker_details.py`

## 12.65 avalon\_sdk.worker.worker\_details.WorkerType Class Reference

Inheritance diagram for avalon\_sdk.worker.worker\_details.WorkerType:



Collaboration diagram for avalon\_sdk.worker.worker\_details.WorkerType:



### Static Public Attributes

- int **TEE\_SGX** = 1
- int **MPC** = 2
- int **ZK** = 3

### 12.65.1 Detailed Description

Worker types are:

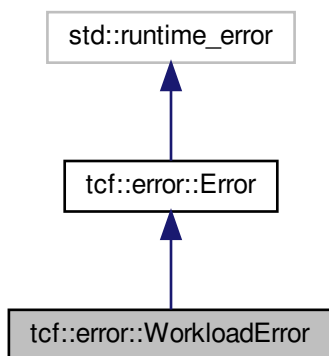
1 = TEE-SGX: Intel SGX Trusted Execution Environment (hardware based)  
2 = MPC: Trusted Multi-Party Compute (software/hardware based)  
3 = ZK: Zero-knowledge proofs (software based)

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

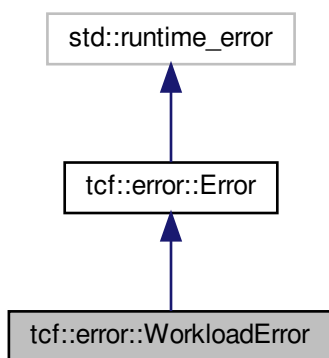
- /home/dano/git/avalon/sdk/avalon\_sdk/worker/worker\_details.py

## 12.66 tcf::error::WorkloadError Class Reference

Inheritance diagram for tcf::error::WorkloadError:



Collaboration diagram for tcf::error::WorkloadError:



### Public Member Functions

- **WorkloadError** (const std::string &msg)

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

- `/home/dano/git/avalon/common/cpp/error.h`

## 12.67 WorkloadProcessor Class Reference

```
#include <workload_processor.h>
```

### Public Member Functions

- virtual [WorkloadProcessor](#) \* [Clone](#) () const =0
- virtual void [ProcessWorkOrder](#) (std::string workload\_id, const [ByteArray](#) &requester\_id, const [ByteArray](#) &worker\_id, const [ByteArray](#) &work\_order\_id, const std::vector< [tcf::WorkOrderData](#) > &in\_work\_order\_data, std::vector< [tcf::WorkOrderData](#) > &out\_work\_order\_data)=0

### Static Public Member Functions

- static [WorkloadProcessor](#) \* [CreateWorkloadProcessor](#) (std::string workload\_id)
- static [WorkloadProcessor](#) \* [RegisterWorkloadProcessor](#) (std::string workload\_id, [WorkloadProcessor](#) \*processor)

### Static Public Attributes

- static std::map< std::string, [WorkloadProcessor](#) \* > [workload\\_processor\\_table](#)

#### 12.67.1 Detailed Description

Class to register, create, and process a workload.

#### 12.67.2 Member Function Documentation

##### 12.67.2.1 Clone()

```
virtual WorkloadProcessor* WorkloadProcessor::Clone ( ) const [pure virtual]
```

Clone a [WorkloadProcessor](#)

##### 12.67.2.2 CreateWorkloadProcessor()

```
WorkloadProcessor * WorkloadProcessor::CreateWorkloadProcessor (
    std::string workload_id ) [static]
```

Create a [WorkloadProcessor](#)

## Parameters

<i>workload</i> ↔ _id	Workload identifier
--------------------------	---------------------

## Returns

Pointer to [WorkloadProcessor](#)

## 12.67.2.3 ProcessWorkOrder()

```
virtual void WorkloadProcessor::ProcessWorkOrder (
    std::string workload_id,
    const ByteArray & requester_id,
    const ByteArray & worker_id,
    const ByteArray & work_order_id,
    const std::vector< tcf::WorkOrderData > & in_work_order_data,
    std::vector< tcf::WorkOrderData > & out_work_order_data ) [pure virtual]
```

Process the workload.

## Parameters

<i>workload_id</i>	Workload identifier string
<i>requester_id</i>	Requester ID to identify who submitted work order
<i>worker_id</i>	Worker ID, a unique string identifying this type of work order processor
<i>work_order_id</i>	Unique work order ID for this type of work order processor
<i>in_work_order_data</i>	Work order data input submitted to the work order processor
<i>out_work_order_data</i>	Work order data returned by the work order processor

## 12.67.2.4 RegisterWorkloadProcessor()

```
WorkloadProcessor * WorkloadProcessor::RegisterWorkloadProcessor (
    std::string workload_id,
    WorkloadProcessor * processor ) [static]
```

Register a [WorkloadProcessor](#). Used by the workloads to register themselves

## Parameters

<i>workload</i> ↔ _id	Workload identifier
--------------------------	---------------------

## Returns

Pointer to [WorkloadProcessor](#)

## 12.67.3 Member Data Documentation

## 12.67.3.1 workload\_processor\_table

```
std::map< std::string, WorkloadProcessor * > WorkloadProcessor::workload_processor_table [static]
```

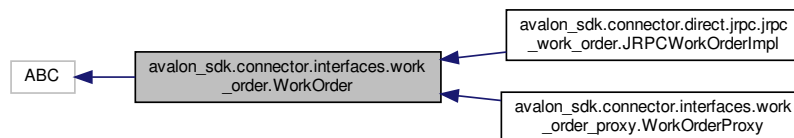
Mapping between workload id and [WorkloadProcessor](#).

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

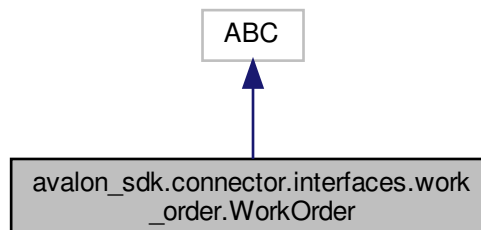
- /home/dano/git/avalon/common/sgx\_workload/workload/[workload\\_processor.h](#)
- /home/dano/git/avalon/common/sgx\_workload/workload/[workload\\_processor.cpp](#)

## 12.68 avalon\_sdk.connector.interfaces.work\_order.WorkOrder Class Reference

Inheritance diagram for `avalon_sdk.connector.interfaces.work_order.WorkOrder`:



Collaboration diagram for `avalon_sdk.connector.interfaces.work_order.WorkOrder`:



## Public Member Functions

- `def __init__(self)`
- `def work_order_submit(self, work_order_id, worker_id, requester_id, work_order_request, id=None)`
- `def work_order_get_result(self, work_order_id, id=None)`
- `def encryption_key_get(self, worker_id, requester_id, last_used_key_nonce=None, tag=None, signature_nonce=None, signature=None, id=None)`
- `def encryption_key_set(self, worker_id, encryption_key, encryption_nonce, tag, signature, id=None)`

### 12.68.1 Detailed Description

This class is an abstract base class that contains abstract APIs to manage work orders.

### 12.68.2 Member Function Documentation

#### 12.68.2.1 encryption\_key\_get()

```
def avalon_sdk.connector.interfaces.work_order.WorkOrder.encryption_key_get (
    self,
    worker_id,
    requester_id,
    last_used_key_nonce = None,
    tag = None,
    signature_nonce = None,
    signature = None,
    id = None )
```

Get Encryption Key Request Payload.

#### Inputs

1. worker\_id is an id of the worker to retrieve an encryption key for.
2. last\_used\_key\_nonce is an optional nonce associated with the last key retrieved. If it is provided, the key retrieved should be newer than this one. Otherwise any key can be retrieved.
3. tag is tag that should be associated with the returned key, e.g. requester id. This is an optional parameter. If it is not provided, requesterId below is used as a key.
4. requester\_id is an id of the requester that plans to use the returned key to submit one or more work orders using this key.
5. signature\_nonce is an optional parameter and is used only if signature below is also provided.
6. signature is an optional signature of workerId, lastUsedKeyNonce, tag, and signatureNonce.
7. id is json rpc request id, it is optional

#### Output

1. errorCode is the result of the operation.
  - 1 - generic error
  - 2 - operation is not supported
  - 3 - invalid parameter
  - 4 - access denied
  - 5 - not ready, retry later. This is a recoverable error that may happen if the requester makes its first request for keys, or the requester retrieves keys, faster than the worker produces them.



- The requester should retry later.
2. worker\_id is an id of the worker that created the encryption key.
  3. encryptionKey is an encryption key.
  4. encryptionKeyNonce is a nonce associated with the key.
  5. tag is tag associated with the key.
  6. signature is a signature generated by the worker.

### 12.68.2.2 encryption\_key\_set()

```
def avalon_sdk.connector.interfaces.work_order.WorkOrder.encryption_key_set (
    self,
    worker_id,
    encryption_key,
    encryption_nonce,
    tag,
    signature,
    id = None )
```

Set encryption key request payload

Parameters:

worker\_id ID of the worker to retrieve an encryption key  
 encryption\_key Encryption key  
 encryption\_nonce Nonce associated with the key  
 tag Tag that should be associated with the returned key,  
     e.g. requester id. This is an optional parameter.  
     If it is not provided, requesterId below is used  
     as a key  
 signature Signature generated by the worker on the worker\_id,  
     tag, and encryption\_nonce  
 id Optional Optional JSON RPC request ID

Returns:

Error code is the result of the operation.

### 12.68.2.3 work\_order\_get\_result()

```
def avalon_sdk.connector.interfaces.work_order.WorkOrder.work_order_get_result (
    self,
    work_order_id,
    id = None )
```

Query blockchain to get a work order result.

If a Requester receives a response stating that its work order state is "scheduled" or "processing", it should poll the Worker Service later to get the result:

1. Poll the Worker Service periodically until the Work Order is completed successfully or in error
2. Wait for the Work Order Receipt complete event and retrieve a final result.

Parameters:

work\_order\_id Work Order ID that was sent in the  
     corresponding work\_order\_submit request  
 id Optional JSON RPC request ID

Returns:

Tuple containing work order status, worker id, work order request, work order response, and error code.  
 None on error.

### 12.68.2.4 work\_order\_submit()

```
def avalon_sdk.connector.interfaces.work_order.WorkOrder.work_order_submit (
    self,
    work_order_id,
    worker_id,
    requester_id,
    work_order_request,
    id = None )
```

Submit a work order request.

**Parameters:**

work_order_id	Unique ID of the work order request
worker_id	Identifier for the worker
requester_id	Unique id to identify the requester
work_order_request	JSON RPC string work order request. Defined in EEA specification 6.1.1.
id	Optional JSON RPC request ID

**Returns:**

errorCode	0 on success and non-zero on error.
-----------	-------------------------------------

work\_order\_request is a JSON string containing following parameters:

```
{
    "responseTimeoutMSecs": <integer>,
    "payloadFormat": <string>
    "resultUri": <string>,
    "notifyUri": <string>,
    "workOrderId": <hex string>,
    "workerId": <hex string or DID>,
    "workloadId": <hex string>,
    "requesterId": <hex string>,
    "workerEncryptionKey": <hex string>,
    "dataEncryptionAlgorithm": <string>,
    "encryptedSessionKey": <hex string>,
    "sessionKeyIv": <hex string>,
    "requesterNonce": <hex string>,
    "encryptedRequestHash": <hex string>,
    "requesterSignature": <BASE64 string>,
},
```

1. responseTimeoutMSecs - is a maximum timeout in milliseconds that the caller will wait for the response. Setting this timeout to zero means that the work order is submitted in the asynchronous (resultUri is present), notify (notifyUri is present), or poll mode (neither resultUri nor notifyUri is present). In this case, the TCS should schedule the request for execution and immediately return an error response with error code set to "scheduled". If the timeout is not zero, the work order is in synchronous mode. The TCS should wait for the work order completion before returning the response to the participant. If the request cannot be completed within the allocated interval, the work order should be cancelled and a corresponding error should be returned to the participant.
2. payloadFormat defines how signatures and data items are formatted in this work order request and corresponding response.
3. resultUri is an optional parameter. If it is specified, the WorkerService should submit the Work Order result to this URI. See section Work Order Asynchronous Result.
4. notifyUri is an optional parameter. If it is specified, the WorkerService should send an event to this URI upon the Work Order completion.
5. workOrderId is an id assigned to the Work Order by the Requester and can be registered using the Work Order Receipts API.
6. workerId is a worker id to process the work order, e.g. an Ethereum address or its DID.
7. workloadId is an id of the workload to be executed by the worker. It is an optional value if the worker includes a single workload.
8. requesterId is either the Requester's Ethereum address or its DID.
9. workerEncryptionKey is an optional parameter containing the worker

encryption key used for this Work Order. It is useful if a Worker frequently updates its encryption key in the registry and allows some time overlap in utilizing multiple keys.

We assume here that the 'details' submitted during the registration of a worker contain one or more public keys associated with the worker.

10. `dataEncryptionAlgorithm` is an optional parameter that defines an algorithm for encrypting the data in this work order. The default is the first value in the corresponding parameter for the worker

(defined by `workerId`). See section Common Data for All Worker Types.

11. `encryptedSessionKey` is a one-time encryption key generated by the participant submitting the work order. It is sent encrypted with the worker's public encryption key. It is used to encrypt

`encryptedRequestHash` and data item specific data encryption keys.

For the latter see Work Order Data Formats.

12. `sessionKeyIv` is an initialization vector if required by the data encryption algorithm (`encryptedSessionKey`). The default is all zeros.

13. `requesterNonce` is a random string generated by the participant.

It is used to calculate a hash of this work order request.

14. `encryptedRequestHash` is a hash of the work order request encrypted with the key provided in `encryptedSessionKey`.

15. `requesterSignature` is an optional parameter. See section

Work Order Signing for the details.

16. `inData` contains either a JWT of the specified data or an array of one or more Work Order inputs, e.g. state, message containing input parameters.

```
{
  "index": <number>,
  "dataHash": <hex string>,
  "data": <BASE64 string>,
  "encryptedDataEncryptionKey": <hex string>,
  "iv": <hex string>
}
```

i. `index` is an index that determines order of the data items for the hash generation. It also can be used by the worker to identify different inputs and outputs.

ii. `dataHash` is an optional hash value of the data. It is only applicable to `inData` in the work order request and `outData` in the response.

iii. `data` contains either data inline within the JSON document or a reference (e.g. URI) to the data. It is up to the worker to determine how to interpret the data content. This parameter is applicable to

-> `inData` in the work order request

-> `outData` in the request if it contains a reference for the output

-> `outData` in the response

iv. `encryptedDataEncryptionKey` defines if data are encrypted and what key to use. It is included only in the work order request as one of the options below.

If this key is not provided or set to "null" or to "", the data is encrypted using `encryptedSessionKey` from the work order request.

If the key value is set to "-", the data item is not encrypted, a.k.a. sent as clear text.

Otherwise, the data item is sent encrypted with a one-time encryption key generated by a 3rd party that owns this data item (it may be different from the work order requester).

`encryptedDataEncryptionKey` contains this encryption key in double encrypted format.

First, it is encrypted with the worker's public encryption key (e.g. by a 3rd party that owns the data so the requester cannot see the data).

Then the result of the previous encryption above is encrypted with the key from `encryptedSessionKey` (by the requester to enforce the work order integrity).

v. `iv` is an initialization vector if required by the data encryption algorithm. The default is all zeros. If the same encryption key is used to encrypt more than one data item or the hash value of the work order request, the `iv` must be a unique random number for every encryption operation. It is included only in the work order request.

17. `outData` contains information about what and how the work order execution results should be delivered. Same as `inData`

18. `id` is used for json rpc request

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

- `/home/dano/git/avalon/sdk/avalon_sdk/connector/interfaces/work_order.py`

## 12.69 tcf::WorkOrderData Class Reference

```
#include <work_order_data.h>
```

### Public Member Functions

- **WorkOrderData** (int in\_index, [ByteArray](#) data)

### Public Attributes

- int **index**
- [ByteArray](#) **decrypted\_data** = {}

### 12.69.1 Detailed Description

Wrapper class for work order data submitted to workload processors.

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

- /home/dano/git/avalon/common/sgx\_workload/workload/[work\\_order\\_data.h](#)
- /home/dano/git/avalon/common/sgx\_workload/workload/[work\\_order\\_data.cpp](#)

## 12.70 avalon\_sdk.work\_order.work\_order\_params.WorkOrderParams Class Reference

### Public Member Functions

- def **\_\_init\_\_** (self, work\_order\_id, worker\_id, workload\_id, requester\_id, session\_key, session\_iv, requester\_nonce, verifying\_key=None, payload\_format="JSON-RPC", response\_timeout\_msecs=6000, result\_uri=None, notify\_uri=None, worker\_encryption\_key=None, data\_encryption\_algorithm=None)
- def [set\\_response\\_timeout\\_msecs](#) (self, response\_timeout\_msecs)
- def [set\\_payload\\_format](#) (self, payload\_format)
- def [set\\_result\\_uri](#) (self, result\_uri)
- def [set\\_notify\\_uri](#) (self, notify\_uri)
- def [set\\_worker\\_id](#) (self, worker\_id)
- def [set\\_work\\_order\\_id](#) (self, work\_order\_id)
- def [set\\_workload\\_id](#) (self, workload\_id)
- def [set\\_requester\\_id](#) (self, requester\_id)
- def [set\\_worker\\_encryption\\_key](#) (self, worker\_encryption\_key)
- def [set\\_data\\_encryption\\_algorithm](#) (self, data\_encryption\_algorithm)
- def [set\\_encrypted\\_session\\_key](#) (self, encrypted\_session\_key)
- def [set\\_session\\_key\\_iv](#) (self, session\_iv)
- def [set\\_requester\\_nonce](#) (self, requester\_nonce)
- def [add\\_encrypted\\_request\\_hash](#) (self)
- def [add\\_requester\\_signature](#) (self, private\_key)
- def [set\\_verifying\\_key](#) (self, verifying\_key)
- def [add\\_in\\_data](#) (self, data, data\_hash=None, encrypted\_data\_encryption\_key=None, data\_iv=None)
- def [add\\_out\\_data](#) (self, data, data\_hash=None, encrypted\_data\_encryption\_key=None, data\_iv=None)
- def [get\\_params](#) (self)

- def [get\\_in\\_data](#) (self)
- def [get\\_out\\_data](#) (self)
- def [get\\_requester\\_nonce](#) (self)
- def [get\\_worker\\_id](#) (self)
- def [get\\_workload\\_id](#) (self)
- def [get\\_requester\\_id](#) (self)
- def [get\\_session\\_key\\_iv](#) (self)
- def [get\\_work\\_order\\_id](#) (self)
- def [to\\_jrpc\\_string](#) (self, id)
- def [to\\_string](#) (self)

## Public Attributes

- **params\_obj**
- **session\_iv**
- **session\_key**
- **final\_hash**

## 12.70.1 Member Function Documentation

### 12.70.1.1 add\_encrypted\_request\_hash()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.add_encrypted_request_hash (
    self )
```

Calculates request hash based on EEA trusted-computing spec 6.1.8.1 and set encryptedRequestHash parameter in the request.

### 12.70.1.2 add\_in\_data()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.add_in_data (
    self,
    data,
    data_hash = None,
    encrypted_data_encryption_key = None,
    data_iv = None )
```

Add inData work order parameter.

### 12.70.1.3 add\_out\_data()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.add_out_data (
    self,
    data,
    data_hash = None,
    encrypted_data_encryption_key = None,
    data_iv = None )
```

Add outData work order parameter.

### 12.70.1.4 add\_requester\_signature()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.add_requester_signature (
    self,
    private_key )
```

Calculate the signature of the request  
as defined in Off-Chain Trusted Compute EEA spec 6.1.8.3  
and set the requesterSignature parameter in the request.

### 12.70.1.5 get\_in\_data()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.get_in_data (
    self )
```

Return inData work order parameter.

### 12.70.1.6 get\_out\_data()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.get_out_data (
    self )
```

Return outData work order parameter.

#### 12.70.1.7 get\_params()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.get_params (
    self )
```

Return a copy of work order parameters.

#### 12.70.1.8 get\_requester\_id()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.get_requester_id (
    self )
```

Return requesterId work order parameter.

#### 12.70.1.9 get\_requester\_nonce()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.get_requester_nonce (
    self )
```

Return requesterNonce work order parameter.

#### 12.70.1.10 get\_session\_key\_iv()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.get_session_key_iv (
    self )
```

Return sessionKeyIv work order parameter.

#### 12.70.1.11 get\_work\_order\_id()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.get_work_order_id (
    self )
```

Return workOrderId work order parameter.

**12.70.1.12 get\_worker\_id()**

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.get_worker_id (
    self )
```

Return workerId work order parameter.

**12.70.1.13 get\_workload\_id()**

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.get_workload_id (
    self )
```

Return workloadId work order parameter.

**12.70.1.14 set\_data\_encryption\_algorithm()**

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_data_encryption_algorithm (
    self,
    data_encryption_algorithm )
```

Set dataEncryptionAlgorithm work order parameter.

**12.70.1.15 set\_encrypted\_session\_key()**

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_encrypted_session_key (
    self,
    encrypted_session_key )
```

Set encryptedSessionKey work order parameter.

**12.70.1.16 set\_notify\_uri()**

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_notify_uri (
    self,
    notify_uri )
```

Set notifyUri work order parameter.



**12.70.1.17 set\_payload\_format()**

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_payload_format (
    self,
    payload_format )
```

Set payloadFormat work order parameter.

**12.70.1.18 set\_requester\_id()**

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_requester_id (
    self,
    requester_id )
```

Set requesterId work order parameter.

**12.70.1.19 set\_requester\_nonce()**

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_requester_nonce (
    self,
    requester_nonce )
```

Set requesterNonce work order parameter.

**12.70.1.20 set\_response\_timeout\_msecs()**

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_response_timeout_msecs (
    self,
    response_timeout_msecs )
```

Set responseTimeoutMsecs work order parameter.

**12.70.1.21 set\_result\_uri()**

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_result_uri (
    self,
    result_uri )
```

Set resultUri work order parameter.

**12.70.1.22 set\_session\_key\_iv()**

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_session_key_iv (
    self,
    session_iv )
```

Set sessionKeyIv work order parameter.

**12.70.1.23 set\_verifying\_key()**

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_verifying_key (
    self,
    verifying_key )
```

Set verifyingKey work order parameter.

**12.70.1.24 set\_work\_order\_id()**

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_work_order_id (
    self,
    work_order_id )
```

Set workOrderId work order parameter.

**12.70.1.25 set\_worker\_encryption\_key()**

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_worker_encryption_key (
    self,
    worker_encryption_key )
```

Set workerEncryptionKey work order parameter.

**12.70.1.26 set\_worker\_id()**

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_worker_id (
    self,
    worker_id )
```

Set workerId work order parameter.

### 12.70.1.27 set\_workload\_id()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.set_workload_id (
    self,
    workload_id )
```

Set workloadId work order parameter.

### 12.70.1.28 to\_jrpc\_string()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.to_jrpc_string (
    self,
    id )
```

Create a JRPC request in string format using the work order params\_obj.

Parameters:

id JRPC request ID

Returns:

Work order JRPC request as a string.

### 12.70.1.29 to\_string()

```
def avalon_sdk.work_order.work_order_params.WorkOrderParams.to_string (
    self )
```

Create work order request string. It is used to submit a work order.

Returns:

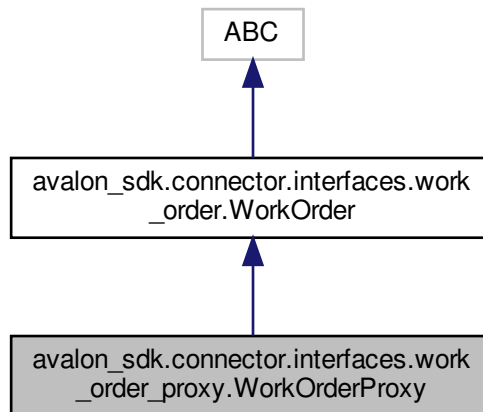
Work order request as a string

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

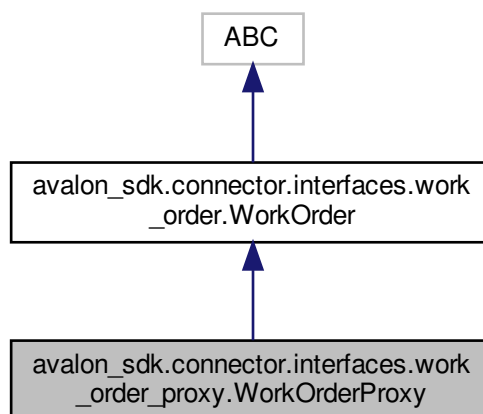
- /home/dano/git/avalon/sdk/avalon\_sdk/work\_order/work\_order\_params.py

## 12.71 `avalon_sdk.connector.interfaces.work_order_proxy.WorkOrderProxy` Class Reference

Inheritance diagram for `avalon_sdk.connector.interfaces.work_order_proxy.WorkOrderProxy`:



Collaboration diagram for `avalon_sdk.connector.interfaces.work_order_proxy.WorkOrderProxy`:



### Public Member Functions

- `def __init__(self)`
- `def encryption_key_start(self, tag, id=None)`
- `def work_order_complete(self, work_order_id, work_order_response)`

### 12.71.1 Detailed Description

This class is an abstract base class that contains abstract APIs to manage work orders.  
This interface is going to be used by proxy model.

### 12.71.2 Member Function Documentation

#### 12.71.2.1 encryption\_key\_start()

```
def avalon_sdk.connector.interfaces.work_order_proxy.WorkOrderProxy.encryption_key_start (
    self,
    tag,
    id = None )
```

Inform the Worker that it should start encryption key generation for this requester.  
This API is for the proxy model.

Parameters:

tag is an optional parameter.  
If it is zero, the transaction sender's address is used as a tag  
id Optional JSON RPC request ID

Returns:

0 on success, otherwise an error code.

#### 12.71.2.2 work\_order\_complete()

```
def avalon_sdk.connector.interfaces.work_order_proxy.WorkOrderProxy.work_order_complete (
    self,
    work_order_id,
    work_order_response )
```

This function is called by the Worker Service to complete a Work Order successfully or in error.  
This API is for the proxy model.

Parameters:

work\_order\_id Unique ID to identify the work order request  
work\_order\_response Work order response data in a string

Returns:

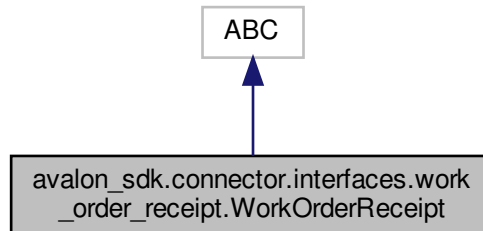
errorCode 0 on success or non-zero on error.

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

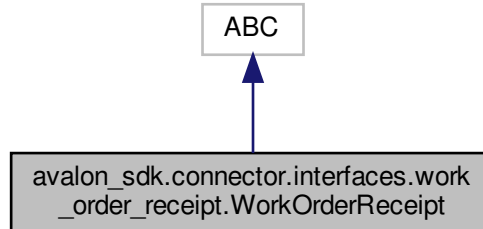
- /home/dano/git/avalon/sdk/avalon\_sdk/connector/interfaces/work\_order\_proxy.py

## 12.72 avalon\_sdk.connector.interfaces.work\_order\_receipt.WorkOrderReceipt Class Reference

Inheritance diagram for avalon\_sdk.connector.interfaces.work\_order\_receipt.WorkOrderReceipt:



Collaboration diagram for avalon\_sdk.connector.interfaces.work\_order\_receipt.WorkOrderReceipt:



### Public Member Functions

- `def __init__(self)`
- `def work_order_receipt_create(self, work_order_id, worker_id, worker_service_id, requester_id, receipt_create_status, work_order_request_hash, id=None)`
- `def work_order_receipt_update(self, work_order_id, updater_id, update_type, update_data, update_signature, signature_rules, id=None)`
- `def work_order_receipt_retrieve(self, work_order_id, id=None)`
- `def work_order_receipt_update_retrieve(self, work_order_id, updater_id, update_index, id=None)`
- `def work_order_receipt_lookup(self, worker_service_id, worker_id, requester_id, receipt_status, id=None)`
- `def work_order_receipt_lookup_next(self, worker_service_id, worker_id, requester_id, receipt_status, last_lookup_tag, id=None)`
- `def work_order_receipt_update(self, work_order_id, updater_id, update_type, update_data, update_signature, signature_rules, id=None)`

### 12.72.1 Detailed Description

This class is an abstract base class that contains abstract APIs to manage work order receipts.

### 12.72.2 Member Function Documentation

#### 12.72.2.1 work\_order\_receipt\_create()

```
def avalon_sdk.connector.interfaces.work_order_receipt.WorkOrderReceipt.work_order_receipt_↔
create (
    self,
    work_order_id,
    worker_id,
    worker_service_id,
    requester_id,
    receipt_create_status,
    work_order_request_hash,
    id = None )
```

Create a work order receipt.

Parameters:

work_order_id	ID of the Work Order
worker_id	Worker id that should execute the Work Order
worker_service_id	ID of the Worker Service that
	hosts the Worker
requester_id	ID of the requester
receipt_create_status	Initial receipt status defined
	in EEA spec 7.1.1
work_order_request_hash	Hash value of the work order request as
	defined in EEA spec 6.7.
id	Optional JSON RPC request ID

Returns:

0 on success, otherwise an error code.

receipt\_create\_status values are:

- 0 - "pending". The work order is waiting to be processed by the worker
- 1 - "completed". The worker processed the Work Order and no more worker updates are expected
- 2 - "processed". The worker processed the Work Order, but additional worker updates are expected, e.g. oracle notifications
- 3 - "failed". The Work Order processing failed, e.g. by the worker service because of invalid workerId
- 4 - "rejected". The Work Order is rejected by the smart contract, e.g. invalid workerServiceId
- 5 to 254 - are reserved
- 255 - indicates any status
- >255 - application-specific values

**12.72.2.2 work\_order\_receipt\_lookup()**

```
def avalon_sdk.connector.interfaces.work_order_receipt.WorkOrderReceipt.work_order_receipt_lookup (
    self,
    worker_service_id,
    worker_id,
    requester_id,
    receipt_status,
    id = None )
```

Lookup a work order receipt.

**Parameters:**

worker\_service\_id Worker Service ID whose receipts will be retrieved  
 worker\_id Worker Id whose receipts are requested  
 requester\_id ID of the entity requesting receipts  
 receipt\_status Defines the status of the receipts retrieved  
 id Optional JSON RPC request ID

**Returns:**

On success, return tuple containing matching count, lookup tag, and list of work order receipt ids:

total\_count Total number of receipts matching the lookup criteria. If this number is bigger than the size of the ids array, the caller should use a lookup\_tag to call work\_order\_receipt\_lookup\_next() to retrieve the remainder of the receipt IDs  
 lookup\_tag Optional parameter. If returned, it means that there are more matching receipts. They can be retrieved by calling work\_order\_receipt\_lookup\_next() with this tag as input  
 ids Array of the Work Order receipt IDs that match the input

**12.72.2.3 work\_order\_receipt\_lookup\_next()**

```
def avalon_sdk.connector.interfaces.work_order_receipt.WorkOrderReceipt.work_order_receipt_lookup_next (
    self,
    worker_service_id,
    worker_id,
    requester_id,
    receipt_status,
    last_lookup_tag,
    id = None )
```

Retrieve subsequent work order receipts after calling work\_order\_receipt\_lookup().

**Parameters:**

worker\_service\_id Worker Service ID  
 worker\_id Worker ID value derived from the worker's DID  
 requester\_id Requester ID  
 last\_lookup\_tag One of the output parameters for function work\_order\_receipt\_lookup()  
 id Optional JSON RPC request ID

**Returns:**



On success, return a tuple containing total count, look up tag, and list of work order IDs:

`total_count` Total number of receipts matching the lookup criteria  
`lookup_tag` Optional parameter. If it is returned, it means that there are more matching receipts that can be retrieved by calling this function again and with this tag as an input parameter  
`ids` Array of the Work Order receipt IDs that match the input criteria from the corresponding call to `work_order_receipt_lookup()`.

#### 12.72.2.4 work\_order\_receipt\_retrieve()

```
def avalon_sdk.connector.interfaces.work_order_receipt.WorkOrderReceipt.work_order_receipt_retrieve (
    self,
    work_order_id,
    id = None )
```

Retrieve a work order receipt.

Parameters:

`work_order_id` ID of the Work Order to be retrieved  
`id` Optional JSON RPC request ID

Outputs:

On success, return `worker_service_id`, `requester_id`, `work_order_id`, `receipt_create_status`, and `work_order_request_hash`, as defined in `work_order_receipt_create()`.

`receipt_create_status` matches the status at the time of the receipt creation if there has not been any receipt updates changing its status. Otherwise it matches the status set by the latest receipt update.

#### 12.72.2.5 work\_order\_receipt\_update() [1/2]

```
def avalon_sdk.connector.interfaces.work_order_receipt.WorkOrderReceipt.work_order_receipt_update (
    self,
    work_order_id,
    updater_id,
    update_type,
    update_data,
    update_signature,
    signature_rules,
    id = None )
```

Update a Work Order Receipt.

This API is implemented by a work order receipts smart contract and it can be called by one of the following participants:

- By or on the behalf of the Worker identified during the receipt creation, e.g. to notify about the work order completion
- By or on the behalf of other Workers, e.g. to submit an oracle notification
- By the Work Order Receipt creator (requester)
- By other participants, e.g. to acknowledge the Work Order results in case of multi-party Work Order processing

Parameters::

`work_order_id` Work Order ID that was sent in the corresponding `work_order_submit` request

`updater_id` ID of the updating entity. It is optional if it is the same as the transaction sender address

`update_type` Type of the Work Order update that defines how the update should be handled.  
If `update_type` is from 0 to 255, the update sets the receipt status to `update_type` value. Refer to [Creating a Work Order Receipt](#). For other values, the processing is application-specific

`update_data` Update-specific data that depends on the updater type defined in EEA spec 7.1.2.  
If the update sets the Work Order Receipt status to completed or processed, it is a hash value of the Work Order Response. In all other cases, `update_data` are application-specific

`update_signature` Optional signature of concatenated `work_order_id`, `update_type`, and `update_data`.  
It is required only if the `updater_id` is not the same as the transaction sender address. Hashing and signing algorithms are defined by `signature_rules`

`signature_rules` Defines hashing and signing algorithms, that are separated by forward slash `'/'`.  
E.g. `"SHA-256/RSA-OAEP-4096"`. Optional parameter but required if signing algorithms are different from the algorithms defined for the Worker defined during receipt creation

`id` Optional JSON RPC request ID

Returns:

Zero on success, otherwise an error code.

### 12.72.2.6 `work_order_receipt_update()` [2/2]

```
def avalon_sdk.connector.interfaces.work_order_receipt.WorkOrderReceipt.work_order_receipt_↔
update (
    self,
    work_order_id,
    updater_id,
    update_type,
    update_data,
    update_signature,
    signature_rules,
    id = None )
```

Update a Work Order Receipt.

This API is implemented by a Work Order Receipts smart contract and it can be called by one of the following participants:

- By or on the behalf of the Worker identified during the receipt creation, e.g. to notify about the work order completion
- By or on the behalf of other Workers, e.g. to submit an oracle

notification

- By the Work Order Receipt creator (requester)
- By other participants, e.g. to acknowledge the Work Order results in case of multi-party Work Order processing

Parameters:

`work_order_id` Work Order ID that was sent in the corresponding `work_order_submit` request

`updater_id` ID of the updating entity. It is optional if it is the same as the transaction sender address

`update_type` Type of the Work Order update that defines how the update should be handled.  
If `update_type` is from 0 to 255, the update sets the receipt status to `update_type` value. Refer to [Creating a Work Order Receipt](#). For other values, the processing is application-specific

`update_data` Update-specific data that depends on the updater type defined in EEA spec 7.1.2.  
If the update sets the Work Order Receipt status to completed or processed, it is a hash value of the Work Order Response. In all other cases, `update_data` are application-specific

`update_signature` Optional signature of concatenated `work_order_id`, `update_type`, and `update_data`.  
It is required only if the `updater_id` is not the same as the transaction sender address. Hashing and signing algorithms are defined by `signature_rules`

`signature_rules` Defines hashing and signing algorithms, that are separated by forward slash `'/'`.  
E.g. "SHA-256/RSA-OAEP-4096". Optional parameter but required if signing algorithms are different from the algorithms defined for the Worker defined during receipt creation

`id` Optional JSON RPC request ID

### 12.72.2.7 work\_order\_receipt\_update\_retrieve()

```
def avalon_sdk.connector.interfaces.work_order_receipt.WorkOrderReceipt.work_order_receipt_update_retrieve (
    self,
    work_order_id,
    updater_id,
    update_index,
    id = None )
```

Retrieving an update to a work order receipt.

Parameters:

`work_order_id` Work Order ID that was sent in the corresponding `work_order_submit` request

`updater_id` ID of the updating entity. Ignored if null

`update_index` Index of the update to retrieve  
Value "0xFFFFFFFF" is reserved to retrieve the last received update

`id` Optional JSON RPC request ID

Returns:

On success, return `updater_id`, `update_type`, `update_data`, `update_signature`, `signature_rules` as defined `work_order_receipt_update()`, and `update_count`.

If `updater_id` is null, `update_count` is the total number of updates for this receipt, otherwise it is the total number of updates made by `updater_id`.

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

- `/home/dano/git/avalon/sdk/avalon_sdk/connector/interfaces/work_order_receipt.py`

## 12.73 avalon\_sdk.work\_order\_receipt.work\_order\_receipt.WorkOrderReceiptRequest

### Class Reference

#### Public Member Functions

- `def __init__ (self)`
- `def create_receipt (self, wo_request, receipt_create_status, signing_key, nonce=None)`
- `def update_receipt (self, work_order_id, update_type, update_data, signing_key)`

#### Public Attributes

- `sig_obj`
- `SIGNING_ALGORITHM`
- `HASHING_ALGORITHM`

#### 12.73.1 Detailed Description

Class to create work order receipt APIs such as create, update, retrieve, and lookup.

#### 12.73.2 Member Function Documentation

##### 12.73.2.1 create\_receipt()

```
def avalon_sdk.work_order_receipt.work_order_receipt.WorkOrderReceiptRequest.create_receipt (
    self,
    wo_request,
    receipt_create_status,
    signing_key,
    nonce = None )
```

Create a work order receipt corresponding to a workorder ID.

Parameters:

<code>wo_request</code>	JSON RPC work order request used to create the work order request as defined in EEA spec 6.1.1
<code>receipt_create_status</code>	Receipt creation status
<code>signing_key</code>	Private key of the signer
<code>nonce</code>	Optional random number or monotonic counter

Returns:

JSON RPC request of type dictionary

## 12.73.2.2 update\_receipt()

```
def avalon_sdk.work_order_receipt.work_order_receipt.WorkOrderReceiptRequest.update_receipt (
    self,
    work_order_id,
    update_type,
    update_data,
    signing_key )
```

Update the existing work order receipt with  
update\_type and update\_data.

Parameters:

work\_order\_id Work order ID whose receipt  
needs to be updated  
update\_type Update type. These values correspond to  
receipt status as defined in EEA Spec 7.1.1  
update\_data Update-specific data that depends on  
the workOrderStatus

Returns:

JSON RPC work order update receipt request of type dictionary

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

- /home/dano/git/avalon/sdk/avalon\_sdk/work\_order\_receipt/work\_order\_receipt.py

## 12.74 avalon\_sdk.work\_order.work\_order\_request\_validator.WorkOrderRequestValidator Class Reference

### Public Member Functions

- def [\\_\\_init\\_\\_](#) (self)
- def [validate\\_parameters](#) (self, params)
- def [validate\\_data\\_format](#) (self, data)

### 12.74.1 Detailed Description

WorkOrderRequestValidator validates work order requests  
for proper parameter fields and valid data formats.

### 12.74.2 Constructor & Destructor Documentation

#### 12.74.2.1 \_\_init\_\_()

```
def avalon_sdk.work_order.work_order_request_validator.WorkOrderRequestValidator.__init__ (
    self )
```

Initialize \_\_param\_key\_map and \_\_value\_key\_map,  
which are key-value pairs.  
The key is the field name and value is a boolean that  
indicates whether a field is mandatory (True) or optional (False).

### 12.74.3 Member Function Documentation

#### 12.74.3.1 `validate_data_format()`

```
def avalon_sdk.work_order.work_order_request_validator.WorkOrderRequestValidator.validate_data_format (
    self,
    data )
```

Validate data format of the params data field (inData or outData).

Parameters:  
data     Data (inData or OutData)

Returns:  
True and empty string on success and  
False and string with error message on failure.

#### 12.74.3.2 `validate_parameters()`

```
def avalon_sdk.work_order.work_order_request_validator.WorkOrderRequestValidator.validate_parameters (
    self,
    params )
```

Validate params dictionary for existence of fields and mandatory fields

Parameters:  
params     Parameter dictionary to validate

Returns:  
True and empty string on success and  
False and string with error message on failure.

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

- `/home/dano/git/avalon/sdk/avalon_sdk/work_order/work_order_request_validator.py`

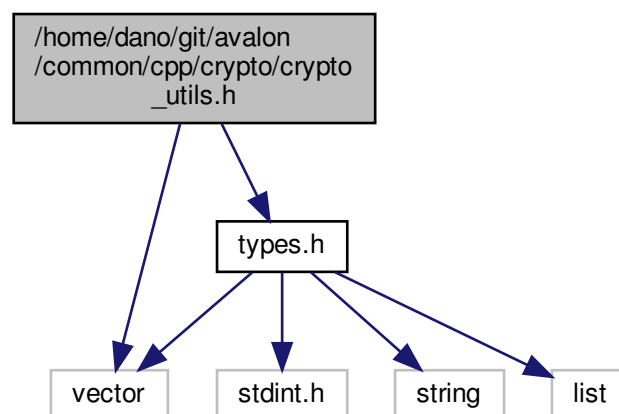
## Chapter 13

# File Documentation

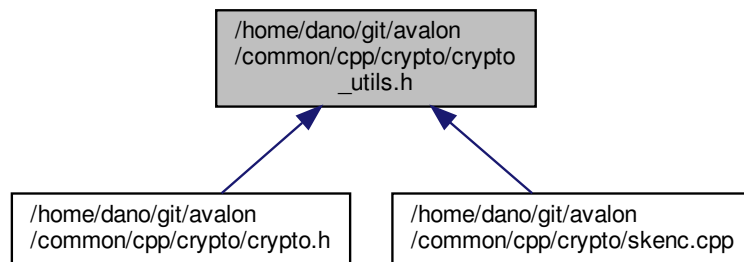
### 13.1 /home/dano/git/avalon/common/cpp/crypto/crypto\_utils.h File Reference

```
#include <vector>
#include "types.h"
```

Include dependency graph for crypto\_utils.h:



This graph shows which files directly or indirectly include this file:



## Functions

- `ByteArray tcf::crypto::ComputeMessageHash` (const `ByteArray` &message)
- `ByteArray tcf::crypto::RandomBitString` (size\_t length)
- `std::string tcf::crypto::CreateHexEncodedEncryptionKey` ()
- `std::string tcf::crypto::DecryptData` (std::string cipher, std::string key)
- `std::string tcf::crypto::EncryptData` (std::string msg, std::string key)

### 13.1.1 Detailed Description

Avalon Crypto Utilities: hashing, base 64 conversion, random number generation, key generation, encrypt, and decrypt.

### 13.1.2 Function Documentation

#### 13.1.2.1 ComputeMessageHash()

```

ByteArray tcf::crypto::ComputeMessageHash (
    const ByteArray & message )
  
```

SHA256 hashing.

Compute SHA256 hash of message.data(). Returns ByteArray containing raw binary data.

#### 13.1.2.2 CreateHexEncodedEncryptionKey()

```

std::string tcf::crypto::CreateHexEncodedEncryptionKey ( )
  
```

Create symmetric encryption key and return hex encoded key string.

Create symmetric encryption key and return hex encoded key string. Uses AES-GCM 256, which also includes authentication. Key generated is 256 bits represented as a 64 hex digit printable string. Key for use with [EncryptData\(\)](#) and [DecryptData\(\)](#).



## 13.1.2.3 DecryptData()

```
std::string tcf::crypto::DecryptData (
    std::string cipher,
    std::string key )
```

Decrypt cipher using given encryption key and return message.

Decrypt cipher using given encryption key and return message. Uses AES-GCM 256, which also includes authentication. Use symmetric encryption key generated by [CreateHexEncodedEncryptionKey\(\)](#). Implemented using [pcrypto::skenc::DecryptMessage\(\)](#).

## 13.1.2.4 EncryptData()

```
std::string tcf::crypto::EncryptData (
    std::string msg,
    std::string key )
```

Encrypt the message using given encryption key and return cipher.

Encrypt the message using given encryption key and return cipher. Uses AES-GCM 256, which also includes authentication. Use symmetric encryption key generated by [CreateHexEncodedEncryptionKey\(\)](#). Implemented using [pcrypto::skenc::EncryptMessage\(\)](#).

## 13.1.2.5 RandomBitString()

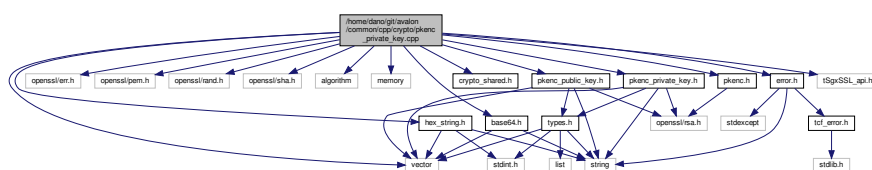
```
ByteArray tcf::crypto::RandomBitString (
    size_t length )
```

Generate a cryptographically strong random bitstring.

Generate a cryptographically strong random bit string. Throws: `RuntimeError`.

## 13.2 /home/dano/git/avalon/common/cpp/crypto/pkenc\_private\_key.cpp File Reference

```
#include "pkenc_private_key.h"
#include <openssl/err.h>
#include <openssl/pem.h>
#include <openssl/rand.h>
#include <openssl/sha.h>
#include <algorithm>
#include <memory>
#include <vector>
#include "base64.h"
#include "crypto_shared.h"
#include "error.h"
#include "hex_string.h"
#include "pkenc.h"
#include "pkenc_public_key.h"
#include "tSgxSSL_api.h"
Include dependency graph for pkenc_private_key.cpp:
```



## Typedefs

- `typedef std::unique_ptr< BIO, void(*)(BIO *)> BIO_ptr`
- `typedef std::unique_ptr< EVP_CIPHER_CTX, void(*)(EVP_CIPHER_CTX *)> CTX_ptr`
- `typedef std::unique_ptr< BN_CTX, void(*)(BN_CTX *)> BN_CTX_ptr`
- `typedef std::unique_ptr< BIGNUM, void(*)(BIGNUM *)> BIGNUM_ptr`
- `typedef std::unique_ptr< RSA, void(*)(RSA *)> RSA_ptr`

## Functions

- `RSA * deserializeRSAPrivateKey (const std::string &encoded)`

### 13.2.1 Detailed Description

Avalon RSA public key generation, serialization, and decryption functions.

### 13.2.2 Function Documentation

#### 13.2.2.1 `deserializeRSAPrivateKey()`

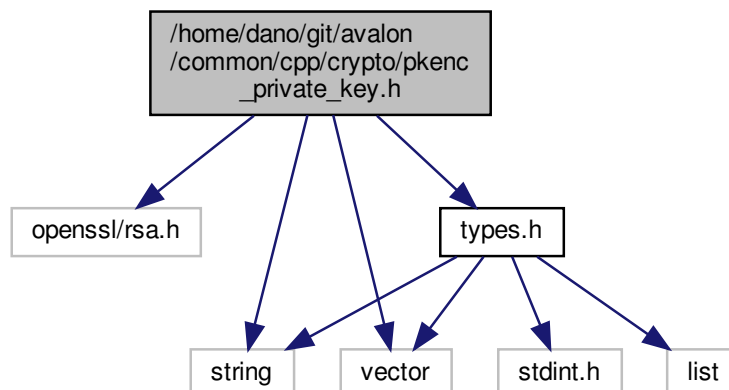
```
RSA* deserializeRSAPrivateKey (
    const std::string & encoded )
```

Utility function: deserialize RSA Private Key. Throws `RuntimeError`, `ValueError`.

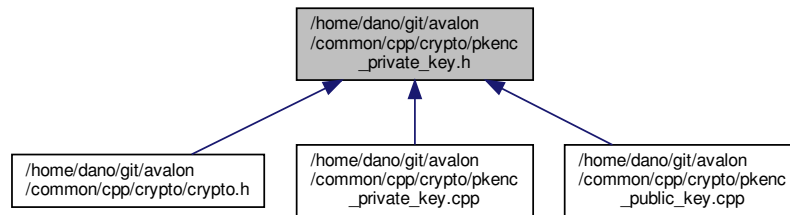
## 13.3 `/home/dano/git/avalon/common/cpp/crypto/pkenc_private_key.h` File Reference

```
#include <openssl/rsa.h>
#include <string>
#include <vector>
#include "types.h"
```

Include dependency graph for `pkenc_private_key.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [tcf::crypto::pkenc::PrivateKey](#)

### 13.3.1 Detailed Description

Avalon RSA public key generation, serialization, and decryption functions.

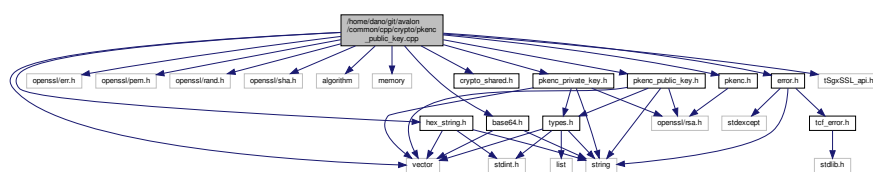
## 13.4 /home/dano/git/avalon/common/cpp/crypto/pkenc\_public\_key.cpp File Reference

```

#include "pkenc_public_key.h"
#include <openssl/err.h>
#include <openssl/pem.h>
#include <openssl/rand.h>
#include <openssl/sha.h>
#include <algorithm>
#include <memory>
#include <vector>
#include "base64.h"
#include "crypto_shared.h"
#include "error.h"
#include "hex_string.h"
#include "pkenc.h"
#include "pkenc_private_key.h"
#include "pkenc_public_key.h"
#include "tSgxSSL_api.h"

```

Include dependency graph for pkenc\_public\_key.cpp:



## Typedefs

- `typedef std::unique_ptr< BIO, void(*)(BIO *)> BIO_ptr`
- `typedef std::unique_ptr< EVP_CIPHER_CTX, void(*)(EVP_CIPHER_CTX *)> CTX_ptr`
- `typedef std::unique_ptr< BN_CTX, void(*)(BN_CTX *)> BN_CTX_ptr`
- `typedef std::unique_ptr< BIGNUM, void(*)(BIGNUM *)> BIGNUM_ptr`
- `typedef std::unique_ptr< RSA, void(*)(RSA *)> RSA_ptr`

## Functions

- `RSA * deserializeRSAPublicKey (const std::string &encoded)`

### 13.4.1 Detailed Description

Avalon RSA public key generation, serialization, and encryption functions.

### 13.4.2 Function Documentation

#### 13.4.2.1 `deserializeRSAPublicKey()`

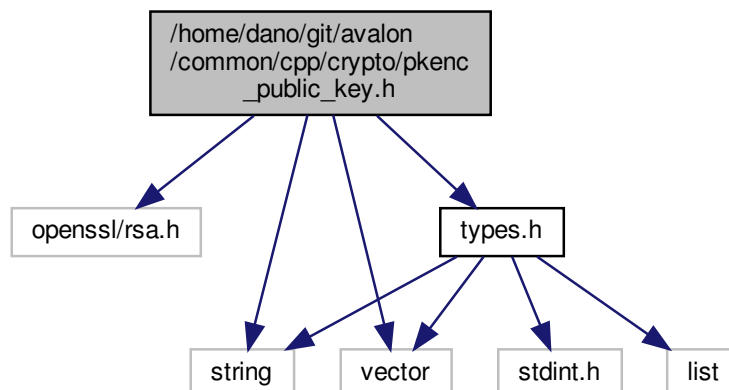
```
RSA* deserializeRSAPublicKey (
    const std::string & encoded )
```

Utility function: deserialize RSA Public Key. Throws `RuntimeError`, `ValueError`.

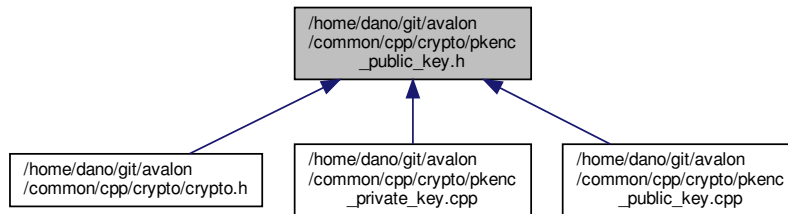
## 13.5 `/home/dano/git/avalon/common/cpp/crypto/pkenc_public_key.h` File Reference

```
#include <openssl/rsa.h>
#include <string>
#include <vector>
#include "types.h"
```

Include dependency graph for `pkenc_public_key.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `tcf::crypto::pkenc::PublicKey`

### 13.5.1 Detailed Description

Avalon RSA public key generation, serialization, and encryption functions.

## 13.6 /home/dano/git/avalon/common/cpp/crypto/SAVE/verify\_certificate.h File Reference

### Functions

- bool `verify_certificate_chain` (const char \*cert\_pem, const char \*ca\_cert\_pem)

### 13.6.1 Detailed Description

Avalon CA certification verification.

### 13.6.2 Function Documentation

#### 13.6.2.1 `verify_certificate_chain()`

```
bool verify_certificate_chain (
    const char * cert_pem,
    const char * ca_cert_pem )
```

Verifies certificate against CA certificate.

Verify that cert\_pem is signed by CA, using CA certificate ca\_cert\_pem as a root of trust.

## Parameters

<i>cert_pem</i>	X.509 Certificate to verify with BEGIN end BLOCKS and new lines
<i>ca_cert_pem</i>	CA Certificate (usually the IAS CA cert) with BEGIN end BLOCKS and new lines

## Returns

true on success and false on failure.

## 13.7 /home/dano/git/avalon/common/cpp/crypto/verify\_certificate.h File Reference

## Functions

- bool [verify\\_certificate\\_chain](#) (const char \*cert\_pem, const char \*ca\_cert\_pem)

### 13.7.1 Detailed Description

Avalon CA certification verification.

### 13.7.2 Function Documentation

#### 13.7.2.1 [verify\\_certificate\\_chain\(\)](#)

```
bool verify_certificate_chain (
    const char * cert_pem,
    const char * ca_cert_pem )
```

Verifies certificate against CA certificate.

Verify that cert\_pem is signed by CA, using CA certificate ca\_cert\_pem as a root of trust.

## Parameters

<i>cert_pem</i>	X.509 Certificate to verify with BEGIN end BLOCKS and new lines
<i>ca_cert_pem</i>	CA Certificate (usually the IAS CA cert) with BEGIN end BLOCKS and new lines

## Returns

true on success and false on failure.

## 13.8 /home/dano/git/avalon/common/cpp/crypto/SAVE/verify\_signature.h File Reference

## Functions

- bool [verify\\_signature](#) (const char \*cert\_pem, const char \*msg, unsigned int msg\_len, const char \*signature, unsigned int signature\_len)

### 13.8.1 Detailed Description

Avalon signature verification.

### 13.8.2 Function Documentation

#### 13.8.2.1 verify\_signature()

```
bool verify_signature (
    const char * cert_pem,
    const char * msg,
    unsigned int msg_len,
    const char * signature,
    unsigned int signature_len )
```

Verifies signature of the message by extracting public key from certificate.

Verify a signature given a message, signature, and cert. Use SHA-256 to hash the message.

#### Parameters

<i>cert_pem</i>	X.509 Certificate to verify with BEGIN end BLOCKS and new lines
<i>msg</i>	Message to verify
<i>msg_len</i>	Length of msg
<i>signature</i>	Signature to verify message, base64 encoded
<i>signature_len</i>	Length of signature

#### Returns

true on success and false on failure.

## 13.9 /home/dano/git/avalon/common/cpp/crypto/verify\_signature.h File Reference

## Functions

- bool [verify\\_signature](#) (const char \*cert\_pem, const char \*msg, unsigned int msg\_len, const char \*signature, unsigned int signature\_len)

### 13.9.1 Detailed Description

Avalon signature verification.

### 13.9.2 Function Documentation

#### 13.9.2.1 `verify_signature()`

```
bool verify_signature (
    const char * cert_pem,
    const char * msg,
    unsigned int msg_len,
    const char * signature,
    unsigned int signature_len )
```

Verifies signature of the message by extracting public key from certificate.

Verify a signature given a message, signature, and cert. Use SHA-256 to hash the message.

#### Parameters

<i>cert_pem</i>	X.509 Certificate to verify with BEGIN end BLOCKS and new lines
<i>msg</i>	Message to verify
<i>msg_len</i>	Length of msg
<i>signature</i>	Signature to verify message, base64 encoded
<i>signature_len</i>	Length of signature

#### Returns

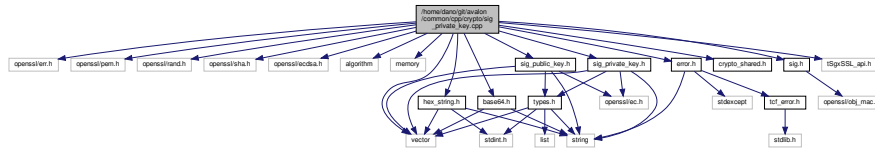
true on success and false on failure.

## 13.10 `/home/dano/git/avalon/common/cpp/crypto/sig_private_key.cpp` File Reference

```
#include <openssl/err.h>
#include <openssl/pem.h>
#include <openssl/rand.h>
#include <openssl/sha.h>
#include <openssl/ecdsa.h>
#include <algorithm>
#include <memory>
#include <vector>
#include "base64.h"
#include "crypto_shared.h"
#include "error.h"
#include "hex_string.h"
#include "sig.h"
```



```
#include "sig_public_key.h"
#include "sig_private_key.h"
#include "tSgxSSL_api.h"
Include dependency graph for sig_private_key.cpp:
```



## Typedefs

- typedef std::unique\_ptr< BIO, void(\*) (BIO \*)> **BIO\_ptr**
- typedef std::unique\_ptr< EVP\_CIPHER\_CTX, void(\*) (EVP\_CIPHER\_CTX \*)> **CTX\_ptr**
- typedef std::unique\_ptr< BN\_CTX, void(\*) (BN\_CTX \*)> **BN\_CTX\_ptr**
- typedef std::unique\_ptr< BIGNUM, void(\*) (BIGNUM \*)> **BIGNUM\_ptr**
- typedef std::unique\_ptr< EC\_GROUP, void(\*) (EC\_GROUP \*)> **EC\_GROUP\_ptr**
- typedef std::unique\_ptr< EC\_POINT, void(\*) (EC\_POINT \*)> **EC\_POINT\_ptr**
- typedef std::unique\_ptr< EC\_KEY, void(\*) (EC\_KEY \*)> **EC\_KEY\_ptr**
- typedef std::unique\_ptr< ECDSA\_SIG, void(\*) (ECDSA\_SIG \*)> **ECDSA\_SIG\_ptr**

## Functions

- void **ECDSA\_SIG\_get0** (const ECDSA\_SIG \*sig, const BIGNUM \*\*ptr\_r, const BIGNUM \*\*ptr\_s)
- int **ECDSA\_SIG\_set0** (ECDSA\_SIG \*sig, BIGNUM \*r, BIGNUM \*s)
- EC\_KEY \* **deserializeECDSAPrivateKey** (const std::string &encoded)

### 13.10.1 Detailed Description

Avalon ECDSA private key functions: generation, serialization, and signing. Used for Secp256k1.

### 13.10.2 Function Documentation

#### 13.10.2.1 deserializeECDSAPrivateKey()

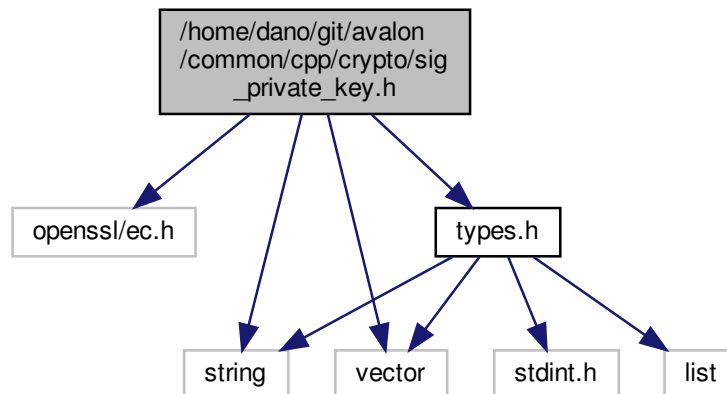
```
EC_KEY* deserializeECDSAPrivateKey (
    const std::string & encoded )
```

Utility function: Deserialize ECDSA Private Key. Throws RuntimeError, ValueError.

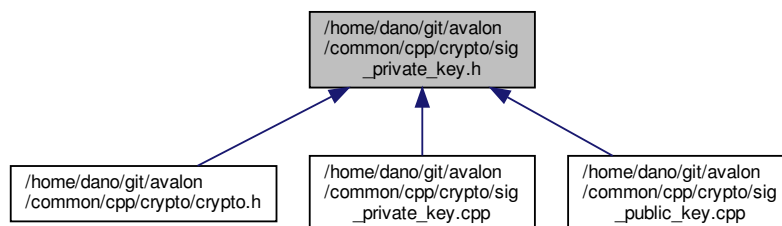
### 13.11 /home/dano/git/avalon/common/cpp/crypto/sig\_private\_key.h File Reference

```
#include <openssl/ec.h>
#include <string>
#include <vector>
#include "types.h"
```

Include dependency graph for sig\_private\_key.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class `tcf::crypto::sig::PrivateKey`

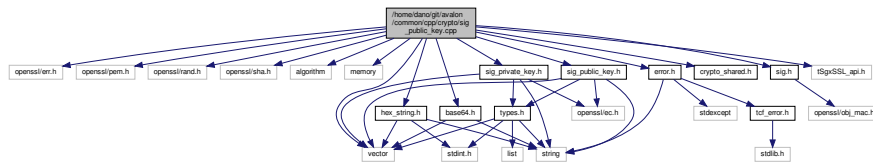
#### 13.11.1 Detailed Description

Avalon ECDSA private key functions: generation, serialization, and signing. ECDSA used for Secp256k1 elliptical curves.

## 13.12 /home/dano/git/avalon/common/cpp/crypto/sig\_public\_key.cpp File Reference

```
#include <openssl/err.h>
#include <openssl/pem.h>
#include <openssl/rand.h>
#include <openssl/sha.h>
#include <algorithm>
#include <memory>
#include <vector>
#include "base64.h"
#include "crypto_shared.h"
#include "error.h"
#include "hex_string.h"
#include "sig.h"
#include "sig_private_key.h"
#include "sig_public_key.h"
#include "tSgxSSL_api.h"
```

Include dependency graph for sig\_public\_key.cpp:



### Typedefs

- typedef std::unique\_ptr< BIO, void(\*) (BIO \*)> **BIO\_ptr**
- typedef std::unique\_ptr< EVP\_CIPHER\_CTX, void(\*) (EVP\_CIPHER\_CTX \*)> **CTX\_ptr**
- typedef std::unique\_ptr< BN\_CTX, void(\*) (BN\_CTX \*)> **BN\_CTX\_ptr**
- typedef std::unique\_ptr< BIGNUM, void(\*) (BIGNUM \*)> **BIGNUM\_ptr**
- typedef std::unique\_ptr< EC\_KEY, void(\*) (EC\_KEY \*)> **EC\_KEY\_ptr**
- typedef std::unique\_ptr< EC\_GROUP, void(\*) (EC\_GROUP \*)> **EC\_GROUP\_ptr**
- typedef std::unique\_ptr< EC\_POINT, void(\*) (EC\_POINT \*)> **EC\_POINT\_ptr**
- typedef std::unique\_ptr< ECDSA\_SIG, void(\*) (ECDSA\_SIG \*)> **ECDSA\_SIG\_ptr**

### Functions

- EC\_KEY \* [deserializeECDSAPublicKey](#) (const std::string &encoded)

#### 13.12.1 Detailed Description

Avalon ECDSA signature public key serialization and verification functions. Used for Secp256k1.

#### 13.12.2 Function Documentation

### 13.12.2.1 deserializeECDSAPublicKey()

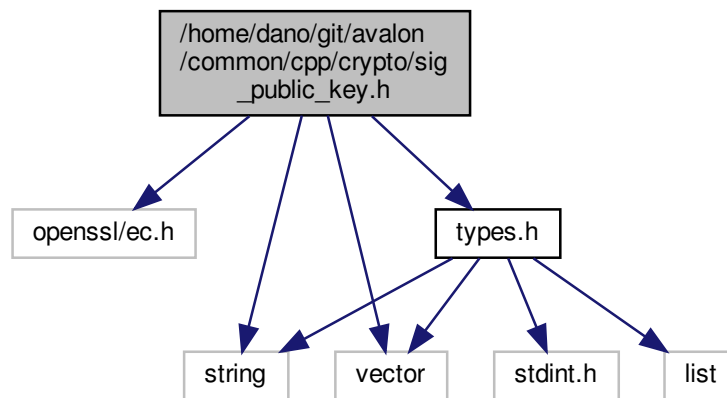
```
EC_KEY* deserializeECDSAPublicKey (
    const std::string & encoded )
```

Utility function: deserialize ECDSA Public Key. Throws RuntimeError, ValueError.

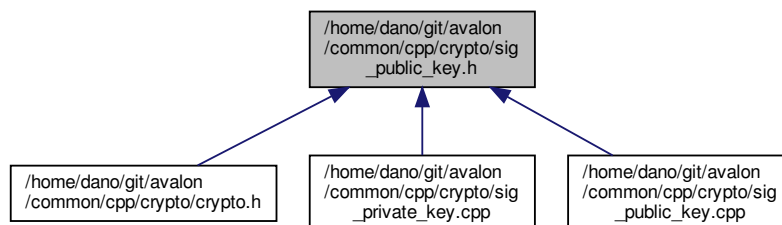
## 13.13 /home/dano/git/avalon/common/cpp/crypto/sig\_public\_key.h File Reference

```
#include <openssl/ec.h>
#include <string>
#include <vector>
#include "types.h"
```

Include dependency graph for sig\_public\_key.h:



This graph shows which files directly or indirectly include this file:



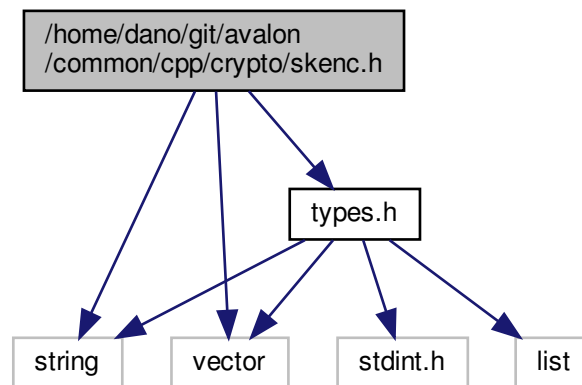
## Classes

- class [tcf::crypto::sig::PublicKey](#)

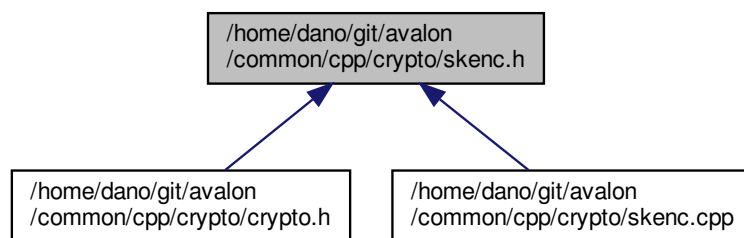


### 13.15 /home/dano/git/avalon/common/cpp/crypto/skenc.h File Reference

```
#include <string>
#include <vector>
#include "types.h"
Include dependency graph for skenc.h:
```



This graph shows which files directly or indirectly include this file:



#### Namespaces

- [tcf::crypto::constants](#)
- [tcf::crypto::skenc](#)

#### Functions

- [ByteArray tcf::crypto::skenc::GenerateKey\(\)](#)
- [ByteArray tcf::crypto::skenc::GenerateIV\(const std::string &IVstring=std::string\(""\)\)](#)

- `ByteArray tcf::crypto::skenc::EncryptMessage` (const `ByteArray` &key, const `ByteArray` &iv, const `ByteArray` &message)
- `ByteArray tcf::crypto::skenc::EncryptMessage` (const `ByteArray` &key, const `ByteArray` &message)
- `ByteArray tcf::crypto::skenc::DecryptMessage` (const `ByteArray` &key, const `ByteArray` &iv, const `ByteArray` &message)
- `ByteArray tcf::crypto::skenc::DecryptMessage` (const `ByteArray` &key, const `ByteArray` &message)

## Variables

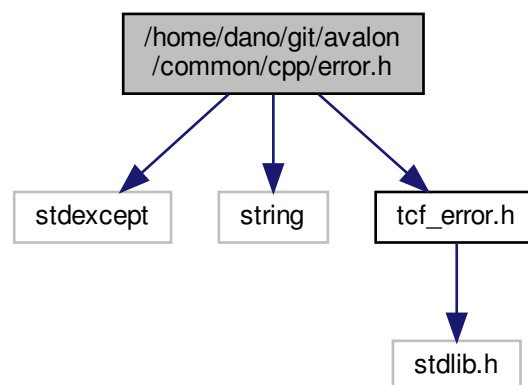
- const int `tcf::crypto::constants::IV_LEN` = 12
- const int `tcf::crypto::constants::SYM_KEY_LEN` = 32
- const int `tcf::crypto::constants::TAG_LEN` = 16

### 13.15.1 Detailed Description

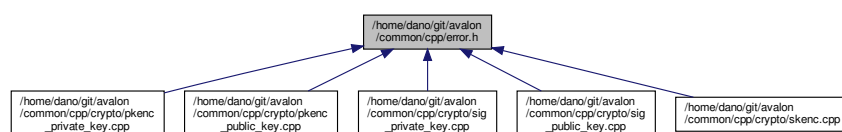
Avalon secret key encryption. Uses AES-GCM 256, which also includes authentication.

## 13.16 /home/dano/git/avalon/common/cpp/error.h File Reference

```
#include <stdexcept>
#include <string>
#include "tcf_error.h"
Include dependency graph for error.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [tcf::error::Error](#)
- class [tcf::error::CryptoError](#)
- class [tcf::error::MemoryError](#)
- class [tcf::error::IOError](#)
- class [tcf::error::RuntimeError](#)
- class [tcf::error::IndexError](#)
- class [tcf::error::DivisionByZero](#)
- class [tcf::error::OverflowError](#)
- class [tcf::error::ValueError](#)
- class [tcf::error::SystemError](#)
- class [tcf::error::SystemBusyError](#)
- class [tcf::error::WorkloadError](#)
- class [tcf::error::UnknownError](#)

## Functions

- `template<typename PointerType >`  
`void tcf::error::ThrowIfNull (const PointerType ptr, const char *msg=nullptr)`
- `template<typename except >`  
`void tcf::error::ThrowIf (bool condition, const char *msg)`

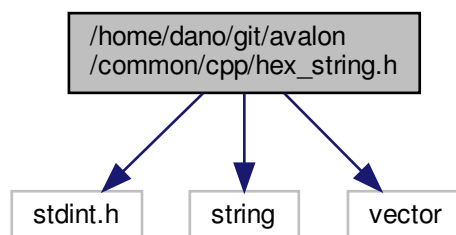
### 13.16.1 Detailed Description

Avalon error and exception handling functions. Namespace `tcf::error`

## 13.17 `/home/dano/git/avalon/common/cpp/hex_string.h` File Reference

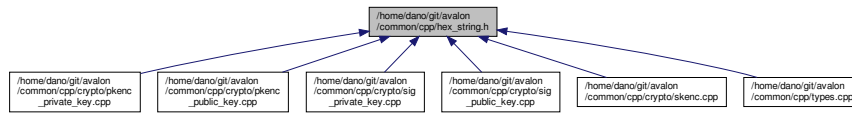
```
#include <stdint.h>
#include <string>
#include <vector>
```

Include dependency graph for `hex_string.h`:





This graph shows which files directly or indirectly include this file:



## Macros

- `#define HEX_STRING_SIZE(x) (static_cast<size_t>(((x) * 2)))`

## Functions

- `std::vector< uint8_t > tcf::HexStringToBinary (const std::string &inHexString)`
- `void tcf::HexStringToBinary (uint8_t *outBinaryData, size_t inBinaryDataLength, const std::string &inHexString)`
- `std::string tcf::BinaryToHexString (const std::vector< uint8_t > &inBinaryData)`
- `std::string tcf::BinaryToHexString (const uint8_t *inBinaryData, size_t inBinaryDataLength)`

### 13.17.1 Detailed Description

Avalon hexadecimal string conversion functions.

### 13.17.2 Macro Definition Documentation

#### 13.17.2.1 HEX\_STRING\_SIZE

```
#define HEX_STRING_SIZE(
    x ) (static_cast<size_t>(((x) * 2)))
```

This macro calculates the length of the actual data portion of the hex-string encoding of a buffer with x bytes PLUS the additional byte needed for the string terminator.

### 13.17.3 Function Documentation

#### 13.17.3.1 BinaryToHexString()

```
std::string tcf::BinaryToHexString (
    const std::vector< uint8_t > & inBinaryData )
```

Convert an array of bytes (represented as either a `std::vector` of bytes or a raw array) to a hex string.

### 13.17.3.2 HexStringToBinary()

```
std::vector< uint8_t > tcf::HexStringToBinary (
    const std::string & inHexString )
```

Convert a hex string (i.e., a string of characters with values between '0'-'9', 'A'-'F') to an array of bytes.

## 13.18 /home/dano/git/avalon/common/cpp/json\_utils.h File Reference

### Functions

- const char \* **GetJsonStr** (const [JSON\\_Object](#) \*json\_object, const char \*name, const char \*err\_msg=NULL)
- void **JsonSetStr** ([JSON\\_Object](#) \*json, const char \*name, const char \*value, const char \*err)
- void **GetByteArray** (const [JSON\\_Object](#) \*object, const char \*name, const char \*err\_msg, [ByteArray](#) &dst)
- double **GetJsonNumber** (const [JSON\\_Object](#) \*object, const char \*name)
- void **JsonSetNumber** ([JSON\\_Object](#) \*json, const char \*name, double value, const char \*err)

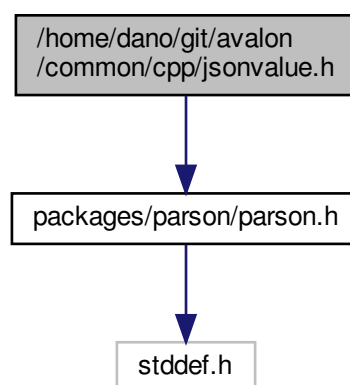
### 13.18.1 Detailed Description

Avalon JSON utilities.

## 13.19 /home/dano/git/avalon/common/cpp/jsonvalue.h File Reference

```
#include "packages/parson/parson.h"
```

Include dependency graph for jsonvalue.h:



### Classes

- class [JsonValue](#)

### 13.19.1 Detailed Description

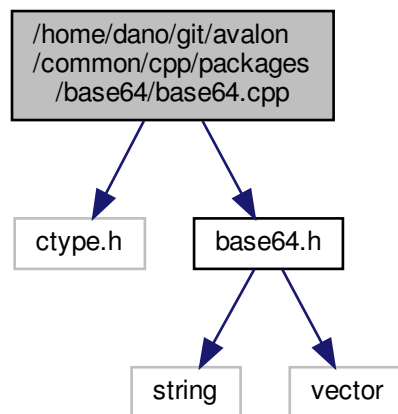
Avalon JSON object extraction utilities.

## 13.20 /home/dano/git/avalon/common/cpp/packages/base64/base64.cpp File Reference

```
#include <ctype.h>
```

```
#include "base64.h"
```

Include dependency graph for base64.cpp:



### Functions

- `std::string base64_encode (const std::vector< uint8_t > &buf)`
- `std::vector< uint8_t > base64_decode (const std::string &encoded_string)`

### 13.20.1 Detailed Description

Base64 encode and decode functions. Used to encode/decode between binary data into a printable character format.

### 13.20.2 Function Documentation

#### 13.20.2.1 `base64_decode()`

```
std::vector<uint8_t> base64_decode (  
    const std::string & encoded_string )
```

Decode a base64 encoded printable string into a vector of binary data. 0 to 2 '=' padding characters may be appended. Decoding stops at first non-base64 character.

## Parameters

<i>encoded_string</i>	Printable string containing base64 encoded data. No embedded whitespace characters are present.
-----------------------	---

## Returns

Vector containing decoded binary data

13.20.2.2 `base64_encode()`

```
std::string base64_encode (  
    const std::vector< uint8_t > & buf )
```

Encode a vector of binary data to a printable base64 string. 0 to 2 '=' padding characters may be appended. No headers or whitespace is generated.

## Parameters

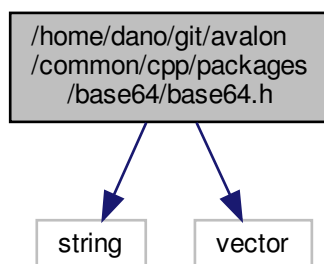
<i>buf</i>	Buffer containing binary data to encode
------------	---

## Returns

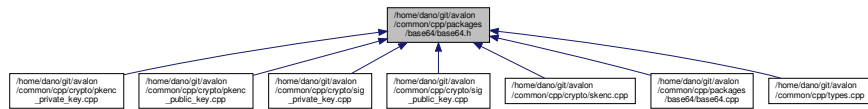
String containing base64 encoded data

13.21 `/home/dano/git/avalon/common/cpp/packages/base64/base64.h` File Reference

```
#include <string>  
#include <vector>  
Include dependency graph for base64.h:
```



This graph shows which files directly or indirectly include this file:



## Functions

- `std::string base64_encode` (`const std::vector< uint8_t > &raw_buffer`)
- `std::vector< uint8_t > base64_decode` (`const std::string &encoded_string`)

### 13.21.1 Detailed Description

Base64 encode and decode functions. Used to encode/decode between binary data into a printable character format.

### 13.21.2 Function Documentation

#### 13.21.2.1 base64\_decode()

```
std::vector<uint8_t> base64_decode (
    const std::string & encoded_string )
```

Decode a base64 encoded printable string into a vector of binary data. 0 to 2 '=' padding characters may be appended. Decoding stops at first non-base64 character.

#### Parameters

<i>encoded_string</i>	Printable string containing base64 encoded data. No embedded whitespace characters are present.
-----------------------	---

#### Returns

Vector containing decoded binary data

#### 13.21.2.2 base64\_encode()

```
std::string base64_encode (
    const std::vector< uint8_t > & buf )
```

Encode a vector of binary data to a printable base64 string. 0 to 2 '=' padding characters may be appended. No headers or whitespace is generated.

## Parameters

<i>buf</i>	Buffer containing binary data to encode
------------	---

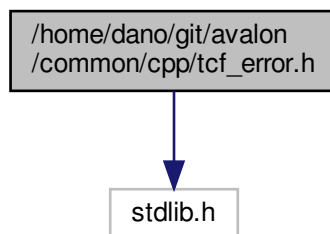
## Returns

String containing base64 encoded data

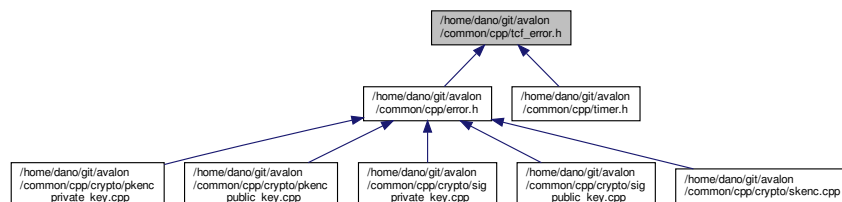
## 13.22 /home/dano/git/avalon/common/cpp/tcf\_error.h File Reference

```
#include <stdlib.h>
```

Include dependency graph for tcf\_error.h:



This graph shows which files directly or indirectly include this file:



## Typedefs

- typedef void(\* **tcf\_log\_t**) (tcf\_log\_level\_t, const char \*message)

## Enumerations

- enum `tcf_err_t` {  
`TCF_SUCCESS` = 0, `TCF_ERR_UNKNOWN` = -1, `TCF_ERR_MEMORY` = -2, `TCF_ERR_IO` = -3,  
`TCF_ERR_RUNTIME` = -4, `TCF_ERR_INDEX` = -5, `TCF_ERR_DIVIDE_BY_ZERO` = -6, `TCF_ERR_OVERFLOW` = -7,  
`TCF_ERR_VALUE` = -8, `TCF_ERR_SYSTEM` = -9, `TCF_ERR_SYSTEM_BUSY` = -10, `TCF_ERR_CRYPTOP`  
= -11,  
`TCF_ERR_INVALID_WORKLOAD` = -12 }
- enum `tcf_log_level_t` {  
`TCF_LOG_DEBUG` = 0, `TCF_LOG_INFO` = 1, `TCF_LOG_WARNING` = 2, `TCF_LOG_ERROR` = 3,  
`TCF_LOG_CRITICAL` = 4 }

### 13.22.1 Detailed Description

Avalon logging levels and error codes.

### 13.22.2 Enumeration Type Documentation

#### 13.22.2.1 `tcf_err_t`

enum `tcf_err_t`

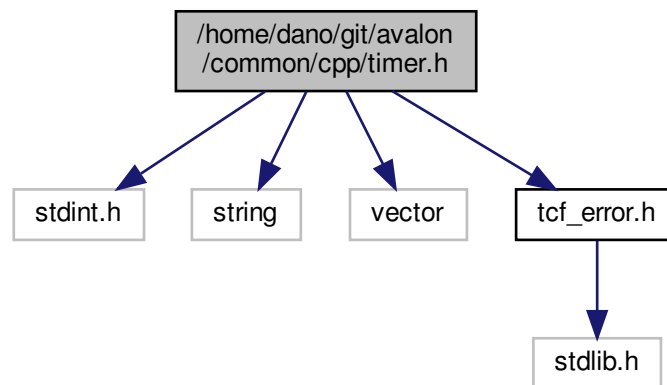
##### Enumerator

<code>TCF_ERR_SYSTEM_BUSY</code>	Indicates that the system is busy and the operation may be retried again. If retries fail this should be converted to a <code>TCF_ERR_SYSTEM</code> for reporting.
<code>TCF_ERR_INVALID_WORKLOAD</code>	Invalid workload ID

## 13.23 /home/dano/git/avalon/common/cpp/timer.h File Reference

```
#include <stdint.h>
#include <string>
#include <vector>
#include "tcf_error.h"
```

Include dependency graph for timer.h:



## Classes

- class [tcf::utility::Timer](#)

## Macros

- `#define __TIMEIT__() {}`

## Functions

- `uint64_t GetTimer (void)`
- `void Log (int level, const char *fmt,...)`

### 13.23.1 Detailed Description

Avalon timer utilities.

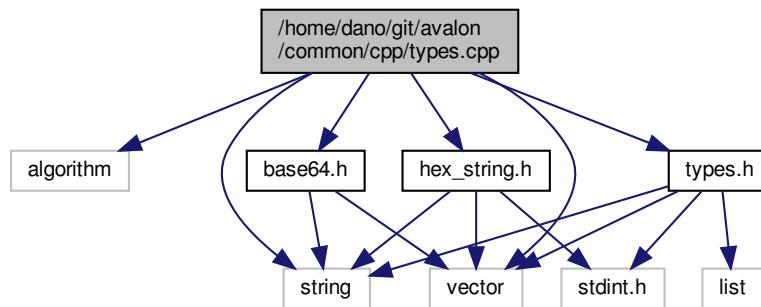
## 13.24 /home/dano/git/avalon/common/cpp/types.cpp File Reference

```
#include <algorithm>
#include <string>
#include <vector>
#include "types.h"
#include "base64.h"
```



```
#include "hex_string.h"
```

Include dependency graph for types.cpp:



## Functions

- `std::string ByteArrayToString` (const `ByteArray` &inArray)
- `StringArray ByteArrayToStringArray` (const `ByteArray` &inArray)
- `Base64EncodedString ByteArrayToBase64EncodedString` (const `ByteArray` &buf)
- `ByteArray Base64EncodedStringToByteArray` (const `Base64EncodedString` &encoded)
- `HexEncodedString ByteArrayToHexEncodedString` (const `ByteArray` &buf)
- `ByteArray HexEncodedStringToByteArray` (const `HexEncodedString` &encoded)

### 13.24.1 Detailed Description

Avalon string utilities, including base 64, hex, byte array, and string array conversion.

### 13.24.2 Function Documentation

#### 13.24.2.1 Base64EncodedStringToByteArray()

```
ByteArray Base64EncodedStringToByteArray (
    const Base64EncodedString & encoded )
```

Simple conversion from Base64EncodedString to ByteArray.

#### 13.24.2.2 ByteArrayToBase64EncodedString()

```
Base64EncodedString ByteArrayToBase64EncodedString (
    const ByteArray & buf )
```

Simple conversion from ByteArray to Base64EncodedString.

### 13.24.2.3 ByteArrayToHexEncodedString()

```
HexEncodedString ByteArrayToHexEncodedString (
    const ByteArray & buf )
```

Simple conversion from ByteArray to HexEncodedString.

### 13.24.2.4 ByteArrayToString()

```
std::string ByteArrayToString (
    const ByteArray & inArray )
```

Simple conversion from ByteArray to std::string

### 13.24.2.5 ByteArrayToStringArray()

```
StringArray ByteArrayToStringArray (
    const ByteArray & inArray )
```

Conversion from ByteArray to [StringArray](#).

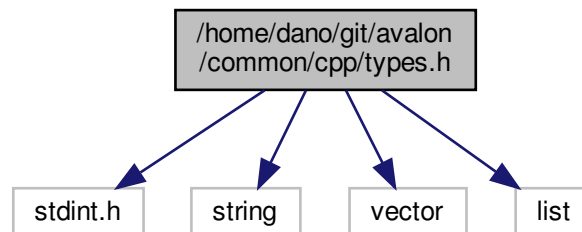
### 13.24.2.6 HexEncodedStringToByteArray()

```
ByteArray HexEncodedStringToByteArray (
    const HexEncodedString & encoded )
```

Simple conversion from HexEncodedString to ByteArray. Throws ValueError.

## 13.25 /home/dano/git/avalon/common/cpp/types.h File Reference

```
#include <stdint.h>
#include <string>
#include <vector>
#include <list>
Include dependency graph for types.h:
```



- class **StringArray**

- typedef std::vector< uint8\_t > ByteArray
- typedef std::string Base64EncodedString
- typedef std::string HexEncodedString

- enum **EnclaveType** { **SINGLETON\_ENCLAVE** = 1, **KME\_ENCLAVE** = 2, **WPE\_ENCLAVE** = 3 }

- `std::string ByteArrayToString` (const `ByteArray` &inArray)
- `StringArray ByteArrayToStringArray` (const `ByteArray` &inArray)
- `Base64EncodedString ByteArrayToBase64EncodedString` (const `ByteArray` &buf)
- `ByteArray Base64EncodedStringToByteArray` (const `Base64EncodedString` &encoded)
- `HexEncodedString ByteArrayToHexEncodedString` (const `ByteArray` &buf)
- `ByteArray HexEncodedStringToByteArray` (const `HexEncodedString` &encoded)

Basic storage types used by Avalon. Avalon string utilities, including base 64, hex, and byte array conversion.

### 13.25.2.1 Base64EncodedString

Type for printable base64 encoded string. May include '=' padding characters. No whitespace, header line, or footer line. For example, SHlwZXJsZW50ZXI=

### 13.25.2.2 ByteArray

```
typedef std::vector<uint8_t> ByteArray
```

Vector type for binary unformatted data.

### 13.25.2.3 HexEncodedString

```
typedef std::string HexEncodedString
```

Type for printable hex encoded string. For example, 2D81454D9C59D73867D65C0FCC98143D4B6F1B0BDB7↵EB04EFED72697F462309C

## 13.25.3 Enumeration Type Documentation

### 13.25.3.1 EnclaveType

```
enum EnclaveType
```

Avalon worker enclave Type (singleton, KME, WPE)

## 13.25.4 Function Documentation

### 13.25.4.1 Base64EncodedStringToByteArray()

```
ByteArray Base64EncodedStringToByteArray (
    const Base64EncodedString & encoded )
```

Simple conversion from Base64EncodedString to ByteArray.

### 13.25.4.2 ByteArrayToBase64EncodedString()

```
Base64EncodedString ByteArrayToBase64EncodedString (
    const ByteArray & buf )
```

Simple conversion from ByteArray to Base64EncodedString.

### 13.25.4.3 ByteArrayToHexEncodedString()

```
HexEncodedString ByteArrayToHexEncodedString (
    const ByteArray & buf )
```

Simple conversion from ByteArray to HexEncodedString.

#### 13.25.4.4 ByteArrayToString()

```
std::string ByteArrayToString (
    const ByteArray & inArray )
```

Simple conversion from ByteArray to std::string

#### 13.25.4.5 ByteArrayToStringArray()

```
StringArray ByteArrayToStringArray (
    const ByteArray & inArray )
```

Conversion from ByteArray to [StringArray](#).

#### 13.25.4.6 HexEncodedStringToByteArray()

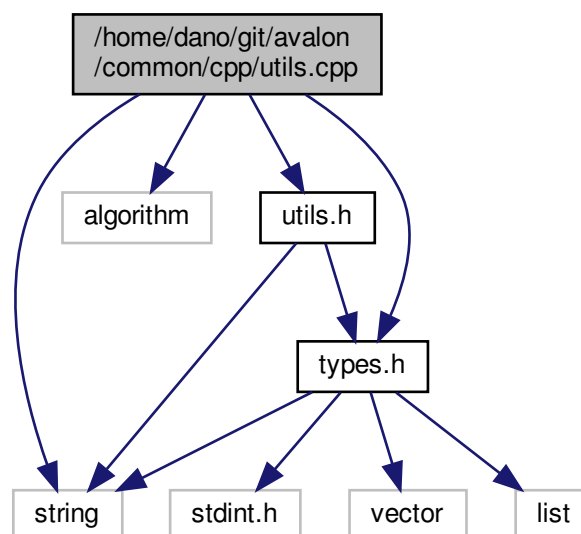
```
ByteArray HexEncodedStringToByteArray (
    const HexEncodedString & encoded )
```

Simple conversion from HexEncodedString to ByteArray. Throws ValueError.

## 13.26 /home/dano/git/avalon/common/cpp/utils.cpp File Reference

```
#include <string>
#include <algorithm>
#include "types.h"
#include "utils.h"
```

Include dependency graph for utils.cpp:



## Functions

- [ByteArray StrToByteArray](#) (std::string str)
- std::string [ByteArrayToStr](#) (ByteArray ba)

### 13.26.1 Detailed Description

Avalon ByteArray and String conversion utilities.

### 13.26.2 Function Documentation

#### 13.26.2.1 ByteArrayToStr()

```
std::string ByteArrayToStr (
    ByteArray ba )
```

Convert a ByteArray vector to a C++ string.

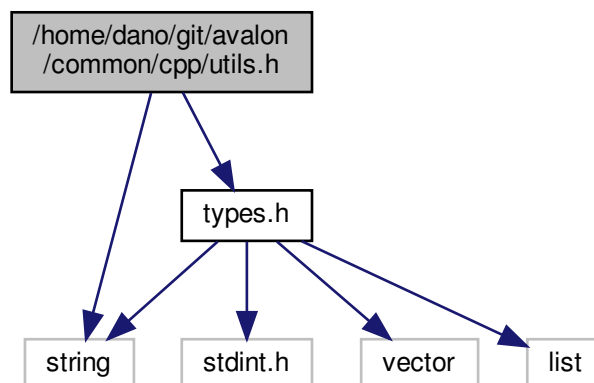
#### 13.26.2.2 StrToByteArray()

```
ByteArray StrToByteArray (
    std::string str )
```

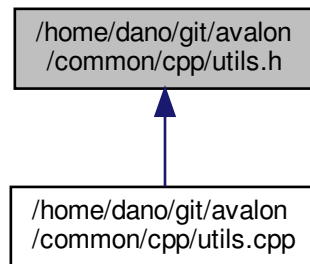
Convert a C++ string to a ByteArray vector.

## 13.27 /home/dano/git/avalon/common/cpp/utils.h File Reference

```
#include <string>
#include "types.h"
Include dependency graph for utils.h:
```



This graph shows which files directly or indirectly include this file:



## Functions

- [ByteArray StrToByteArray](#) (`std::string str`)
- `std::string` [ByteArrayToStr](#) ([ByteArray ba](#))

### 13.27.1 Detailed Description

Avalon ByteArray and String conversion utilities.

### 13.27.2 Function Documentation

#### 13.27.2.1 ByteArrayToStr()

```
std::string ByteArrayToStr (  
    ByteArray ba )
```

Convert a ByteArray vector to a C++ string.

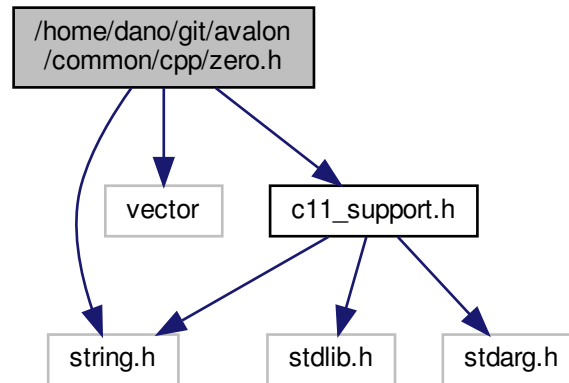
#### 13.27.2.2 StrToByteArray()

```
ByteArray StrToByteArray (  
    std::string str )
```

Convert a C++ string to a ByteArray vector.

## 13.28 /home/dano/git/avalon/common/cpp/zero.h File Reference

```
#include <string.h>
#include <vector>
#include "c11_support.h"
Include dependency graph for zero.h:
```



### Functions

- `template<typename T >`  
`void Zero (T &v)`
- `template<typename T >`  
`void ZeroV (std::vector< T > &v)`
- `void Zero (void *v, size_t length)`

### 13.28.1 Detailed Description

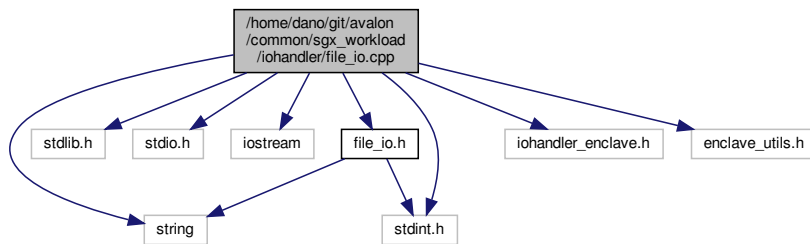
Avalon zero initialization utilities.

## 13.29 /home/dano/git/avalon/common/sgx\_workload/iohandler/file\_io.cpp File Reference

```
#include <string>
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include <stdint.h>
#include "file_io.h"
#include "iohandler_enclave.h"
```



```
#include "enclave_utils.h"
Include dependency graph for file_io.cpp:
```



## Macros

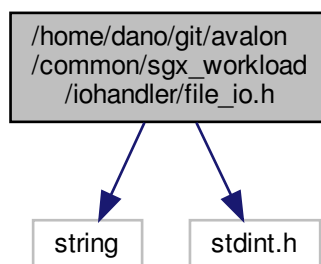
- `#define MAX_FILE_SIZE 1024`
- `#define MAX_IO_RESULT_SIZE 128`

### 13.29.1 Detailed Description

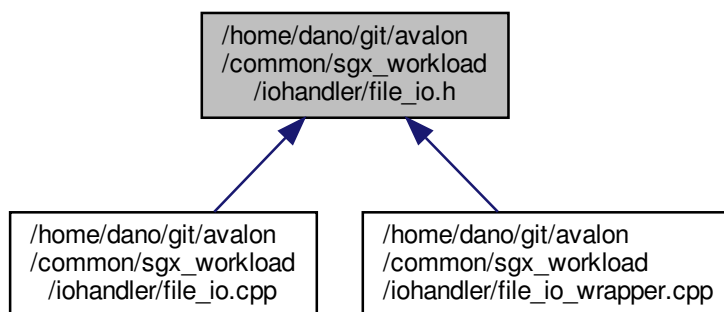
[FileIoExecutor](#) C++ class implementation for Avalon Inside-Out File I/O. To use, `#include "file_io.h"`

## 13.30 /home/dano/git/avalon/common/sgx\_workload/iohandler/file\_io.h File Reference

```
#include <string>
#include <stdint.h>
Include dependency graph for file_io.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [FileIoExecutor](#)

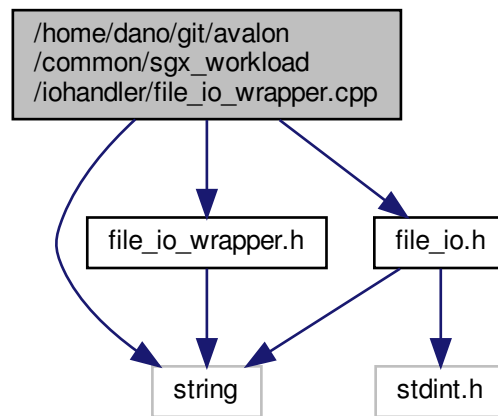
### 13.30.1 Detailed Description

[FileIoExecutor](#) C++ class definitions for Avalon Inside-Out File I/O. To use, #include "file\_io.h"

### 13.31 /home/dano/git/avalon/common/sgx\_workload/iohandler/file\_io\_wrapper.cpp File Reference

```
#include <string>
#include "file_io.h"
#include "file_io_wrapper.h"
```

Include dependency graph for file\_io\_wrapper.cpp:



## Functions

- `std::string` [Read](#) (`std::string` file\_name)
- `uint32_t` [Write](#) (`std::string` file\_name, `std::string` data)
- `uint32_t` [Delete](#) (`std::string` file\_name)

### 13.31.1 Detailed Description

C++ non-class wrapper implementation for Avalon Inside-Out File I/O. To use, `#include "file_io_wrapper.h"`

### 13.31.2 Function Documentation

#### 13.31.2.1 Delete()

```
uint32_t Delete (
    std::string file_name )
```

Delete a file named file\_name. Return the integer status (0 is success, non-0 is failure).

#### Parameters

<i>file_name</i>	Name of the file to delete
------------------	----------------------------

**Returns**

Status of operation (0 on success, non-0 on failure)

**13.31.2.2 Read()**

```
std::string Read (
    std::string file_name )
```

Read a file named file\_name and return the contents in a string. Return an empty string ("") on failure.

**Parameters**

<i>file_name</i>	Name of the file to be read
------------------	-----------------------------

**Returns**

String containing file contents

**13.31.2.3 Write()**

```
uint32_t Write (
    std::string file_name,
    std::string data )
```

Write the contents of string data to a file named file\_name. Return the integer status (0 is success, non-0 is failure).

**Parameters**

<i>file_name</i>	Name of the file to write
<i>data</i>	Contents of file to write

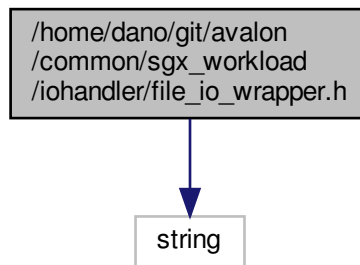
**Returns**

Status of operation (0 on success, non-0 on failure)

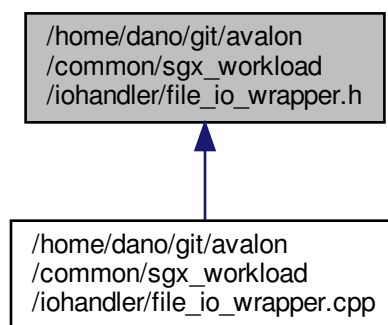
## 13.32 /home/dano/git/avalon/common/sgx\_workload/iohandler/file\_io\_wrapper.h File Reference

```
#include <string>
```

Include dependency graph for file\_io\_wrapper.h:



This graph shows which files directly or indirectly include this file:



## Functions

- `std::string` [Read](#) (`std::string` file\_name)
- `uint32_t` [Write](#) (`std::string` file\_name, `std::string` data)
- `uint32_t` [Delete](#) (`std::string` file\_name)

### 13.32.1 Detailed Description

C++ non-class wrapper definitions for Avalon Inside-Out File I/O. To use, `#include "file_io_wrapper.h"`

### 13.32.2 Function Documentation

### 13.32.2.1 Delete()

```
uint32_t Delete (
    std::string file_name )
```

Delete a file named file\_name. Return the integer status (0 is success, non-0 is failure).

#### Parameters

<i>file_name</i>	Name of the file to delete
------------------	----------------------------

#### Returns

Status of operation (0 on success, non-0 on failure)

### 13.32.2.2 Read()

```
std::string Read (
    std::string file_name )
```

Read a file named file\_name and return the contents in a string. Return an empty string ("") on failure.

#### Parameters

<i>file_name</i>	Name of the file to be read
------------------	-----------------------------

#### Returns

String containing file contents

### 13.32.2.3 Write()

```
uint32_t Write (
    std::string file_name,
    std::string data )
```

Write the contents of string data to a file named file\_name. Return the integer status (0 is success, non-0 is failure).

#### Parameters

<i>file_name</i>	Name of the file to write
<i>data</i>	Contents of file to write

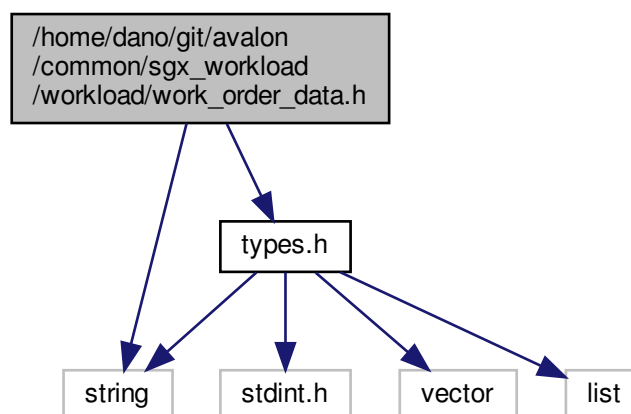
#### Returns

Status of operation (0 on success, non-0 on failure)

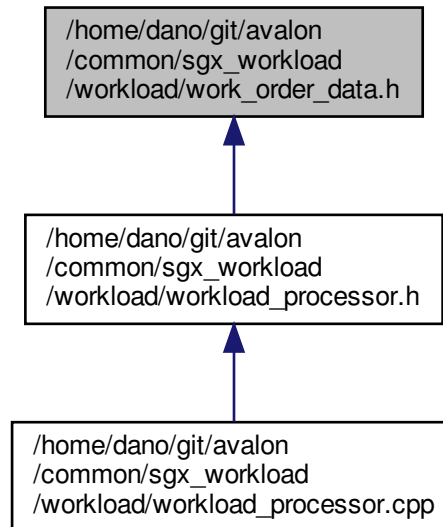
### 13.33 /home/dano/git/avalon/common/sgx\_workload/workload/work\_order\_data.h File Reference

```
#include <string>
#include "types.h"
```

Include dependency graph for work\_order\_data.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [tcf::WorkOrderData](#)

### 13.33.1 Detailed Description

Defines class [tcf::WorkOrderData](#) for work order data submitted to workload processors. To use, `#include "work_order_data.h"`

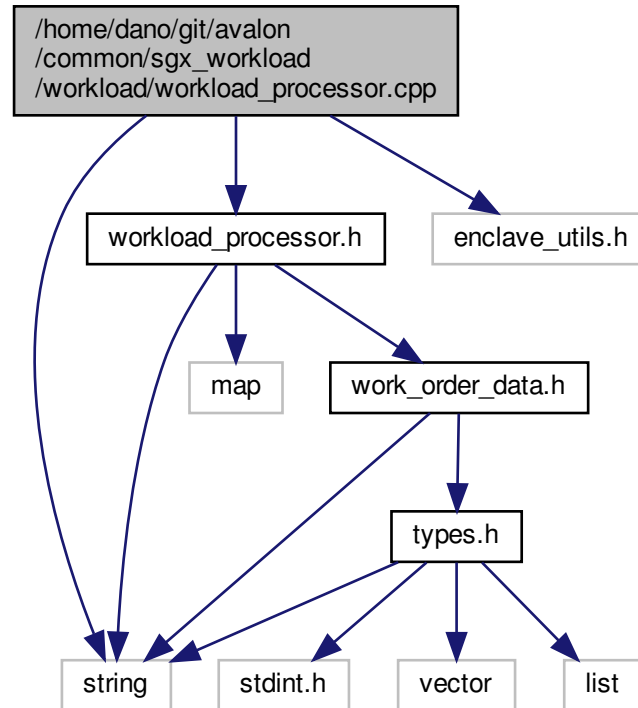
## 13.34 /home/dano/git/avalon/common/sgx\_workload/workload/workload\_processor.cpp File Reference

```
#include <string>
#include "workload_processor.h"
```



```
#include "enclave_utils.h"
```

Include dependency graph for workload\_processor.cpp:



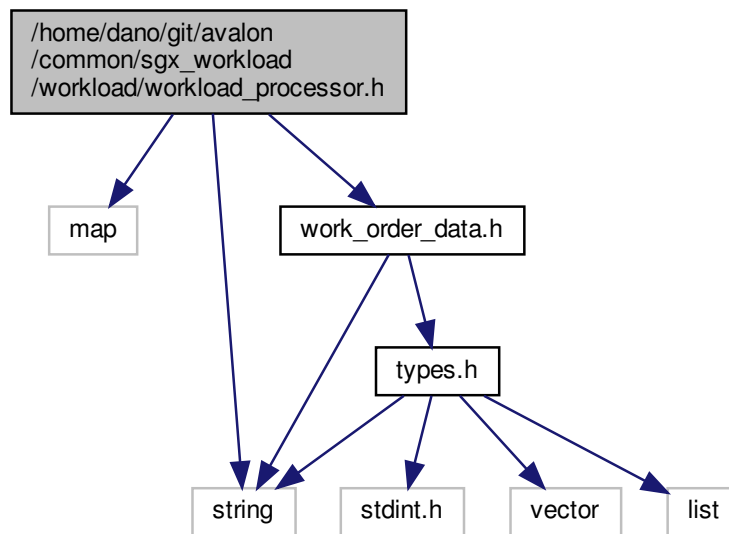
### 13.34.1 Detailed Description

Implements base class [WorkloadProcessor](#) to create an Avalon workload processor.

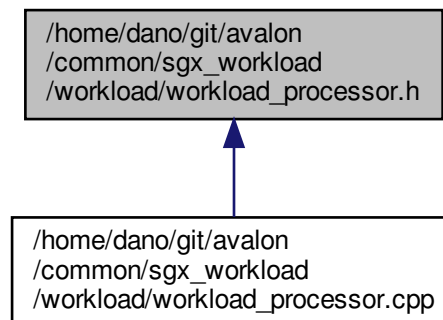
## 13.35 /home/dano/git/avalon/common/sgx\_workload/workload/workload\_processor.h File Reference

```
#include <map>
#include <string>
#include "work_order_data.h"
```

Include dependency graph for workload\_processor.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [WorkloadProcessor](#)

## Macros

- `#define` [IMPL\\_WORKLOAD\\_PROCESSOR\\_CLONE](#)(TYPE) [WorkloadProcessor](#)\* Clone() const { return new TYPE(\*this); }
- `#define` [REGISTER\\_WORKLOAD\\_PROCESSOR](#)(WORKLOADID\_STR, TYPE)

### 13.35.1 Detailed Description

Defines base class [WorkloadProcessor](#) and other definitions to create an Avalon workload processor. To use, #include "workload\_processor.h"

### 13.35.2 Macro Definition Documentation

#### 13.35.2.1 IMPL\_WORKLOAD\_PROCESSOR\_CLONE

```
#define IMPL_WORKLOAD_PROCESSOR_CLONE(  
    TYPE ) WorkloadProcessor* Clone() const { return new TYPE(*this); }
```

This macro clones an instance of class [WorkloadProcessor](#) for an Avalon worker. Example usage in a .h header file: [IMPL\\_WORKLOAD\\_PROCESSOR\\_CLONE\(Workload\)](#)

##### Parameters

<i>TYPE</i>	Name of the Workload class
-------------	----------------------------

#### 13.35.2.2 REGISTER\_WORKLOAD\_PROCESSOR

```
#define REGISTER_WORKLOAD_PROCESSOR(  
    WORKLOADID_STR,  
    TYPE )
```

##### Value:

```
WorkloadProcessor* TYPE##_myProcessor = \  
    WorkloadProcessor::RegisterWorkloadProcessor(  
        WORKLOADID_STR, new TYPE());
```

This macro registers a workload processor for a specific application. It associates a string with a workload. This is the same string that is passed in the work order request JSON payload. Example usage in a .cpp source file: [REGISTER\\_WORKLOAD\\_PROCESSOR\(workload\\_id\\_string, Workload\)](#)

##### Parameters

<i>WORKLOADID_STR</i>	A string literal or variable identifying the workload type
<i>TYPE</i>	Name of the Workload class



# Index

/home/dano/git/avalon/common/cpp/crypto/SAV↔  
E/verify\_certificate.h, 183

/home/dano/git/avalon/common/cpp/crypto/SAV↔  
E/verify\_signature.h, 184

/home/dano/git/avalon/common/cpp/crypto/crypto↔  
utils.h, 177

/home/dano/git/avalon/common/cpp/crypto/pkenc↔  
private\_key.cpp, 179

/home/dano/git/avalon/common/cpp/crypto/pkenc↔  
private\_key.h, 180

/home/dano/git/avalon/common/cpp/crypto/pkenc↔  
public\_key.cpp, 181

/home/dano/git/avalon/common/cpp/crypto/pkenc↔  
public\_key.h, 182

/home/dano/git/avalon/common/cpp/crypto/sig↔  
private\_key.cpp, 186

/home/dano/git/avalon/common/cpp/crypto/sig↔  
private\_key.h, 188

/home/dano/git/avalon/common/cpp/crypto/sig\_public↔  
\_key.cpp, 189

/home/dano/git/avalon/common/cpp/crypto/sig\_public↔  
\_key.h, 190

/home/dano/git/avalon/common/cpp/crypto/skenc.cpp,  
191

/home/dano/git/avalon/common/cpp/crypto/skenc.h, 192

/home/dano/git/avalon/common/cpp/crypto/verify↔  
certificate.h, 184

/home/dano/git/avalon/common/cpp/crypto/verify↔  
signature.h, 185

/home/dano/git/avalon/common/cpp/error.h, 193

/home/dano/git/avalon/common/cpp/hex\_string.h, 194

/home/dano/git/avalon/common/cpp/json\_utils.h, 196

/home/dano/git/avalon/common/cpp/jsonvalue.h, 196

/home/dano/git/avalon/common/cpp/packages/base64/base64↔  
cpp, 197

/home/dano/git/avalon/common/cpp/packages/base64/base64↔  
h, 198

/home/dano/git/avalon/common/cpp/tcf\_error.h, 200

/home/dano/git/avalon/common/cpp/timer.h, 201

/home/dano/git/avalon/common/cpp/types.cpp, 202

/home/dano/git/avalon/common/cpp/types.h, 204

/home/dano/git/avalon/common/cpp/utils.cpp, 207

/home/dano/git/avalon/common/cpp/utils.h, 208

/home/dano/git/avalon/common/cpp/zero.h, 210

/home/dano/git/avalon/common/sgx\_workload/iohandler/file↔  
\_io.cpp, 210

/home/dano/git/avalon/common/sgx\_workload/iohandler/file↔  
\_io.h, 211

/home/dano/git/avalon/common/sgx\_workload/iohandler/file↔  
\_io\_wrapper.cpp, 212

/home/dano/git/avalon/common/sgx\_workload/iohandler/file↔  
\_io\_wrapper.h, 214

/home/dano/git/avalon/common/sgx\_workload/workload/work↔  
\_order\_data.h, 217

/home/dano/git/avalon/common/sgx\_workload/workload/workload↔  
\_processor.cpp, 218

/home/dano/git/avalon/common/sgx\_workload/workload/workload↔  
\_processor.h, 219

\_\_init\_\_

avalon\_crypto\_utils::keys::EnclaveKeys, 36

avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_worker\_registry::Ethereum↔  
WorkerRegistryImpl, 39

avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_work\_order::FabricWorkOrderImpl,  
70

avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_work\_order\_receipt::FabricWork↔  
OrderReceiptImpl, 75

avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_worker\_registry::FabricWorker↔  
RegistryImpl, 61

avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_worker\_registry\_list::FabricWorker↔  
RegistryListImpl, 66

avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_wrapper::FabricWrapper, 79

avalon\_sdk::connector::direct::avalon\_direct↔  
client::AvalonDirectClient, 26

avalon\_sdk::work\_order::work\_order\_request↔  
validator::WorkOrderRequestValidator, 175

avalon\_sdk::worker::worker\_details::Worker↔  
Details, 138

\_sgx\_errlist\_t, 25

PrivateKey

tcf::crypto::pkenc::PrivateKey, 109

tcf::crypto::sig::PrivateKey, 111

~PublicKey

tcf::crypto::pkenc::PublicKey, 115

tcf::crypto::sig::PublicKey, 113

add\_encrypted\_request\_hash

avalon\_sdk::work\_order::work\_order\_params::↔  
WorkOrderParams, 159

add\_in\_data

avalon\_sdk::work\_order::work\_order\_params::↔  
WorkOrderParams, 159

add\_out\_data

- avalon\_sdk::work\_order::work\_order\_params::↵  
    WorkOrderParams, 159
- add\_requester\_signature
  - avalon\_sdk::work\_order::work\_order\_params::↵  
        WorkOrderParams, 160
- avalon\_crypto\_utils.keys.EnclaveKeys, 35
- avalon\_crypto\_utils.signature.ClientSignature, 29
- avalon\_crypto\_utils::keys::EnclaveKeys
  - \_\_init\_\_, 36
- avalon\_crypto\_utils::signature::ClientSignature
  - calculate\_datahash, 30
  - calculate\_request\_hash, 30
  - generate\_client\_signature, 30
  - generate\_signature, 31
  - verify\_create\_receipt\_signature, 31
  - verify\_signature, 31
  - verify\_update\_receipt\_signature, 32
- avalon\_sdk.connector.blockchains.common.contract\_↵  
    response.ContractResponse, 32
- avalon\_sdk.connector.blockchains.ethereum.ethereum\_↵  
    \_listener.BlockchainInterface, 27
- avalon\_sdk.connector.blockchains.ethereum.ethereum\_↵  
    \_listener.EventProcessor, 57
- avalon\_sdk.connector.blockchains.ethereum.ethereum\_↵  
    \_work\_order.EthereumWorkOrderProxyImpl, 47
- avalon\_sdk.connector.blockchains.ethereum.ethereum\_↵  
    \_worker\_registry.EthereumWorkerRegistry\_↵  
        Impl, 38
- avalon\_sdk.connector.blockchains.ethereum.ethereum\_↵  
    \_worker\_registry\_list.EthereumWorker\_↵  
        RegistryListImpl, 43
- avalon\_sdk.connector.blockchains.ethereum.ethereum\_↵  
    \_wrapper.EthereumWrapper, 51
- avalon\_sdk.connector.blockchains.fabric.base.Client\_↵  
    Base, 28
- avalon\_sdk.connector.blockchains.fabric.event\_↵  
    listener.EventListener, 55
- avalon\_sdk.connector.blockchains.fabric.fabric\_work\_↵  
    order.FabricWorkOrderImpl, 69
- avalon\_sdk.connector.blockchains.fabric.fabric\_work\_↵  
    \_order\_receipt.FabricWorkOrderReceiptImpl, 74
- avalon\_sdk.connector.blockchains.fabric.fabric\_↵  
    worker\_registry.FabricWorkerRegistryImpl, 60
- avalon\_sdk.connector.blockchains.fabric.fabric\_↵  
    worker\_registry\_list.FabricWorkerRegistry\_↵  
        ListImpl, 65
- avalon\_sdk.connector.blockchains.fabric.fabric\_↵  
    wrapper.FabricWrapper, 79
- avalon\_sdk.connector.blockchains.fabric.tx\_committer.↵  
    TxCommitter, 134
- avalon\_sdk.connector.direct.avalon\_direct\_client.↵  
    AvalonDirectClient, 25
- avalon\_sdk.connector.direct.jrpc.jrpc\_util.JsonRpc\_↵  
    ErrorCode, 103
- avalon\_sdk.connector.direct.jrpc.jrpc\_work\_order.JR\_↵  
    PCWorkOrderImpl, 92
- avalon\_sdk.connector.direct.jrpc.jrpc\_work\_order\_↵  
    receipt.JRPCWorkOrderReceiptImpl, 95
- avalon\_sdk.connector.direct.jrpc.jrpc\_worker\_registry.↵  
    JRPCWorkerRegistryImpl, 88
- avalon\_sdk.connector.interfaces.work\_order.Work\_↵  
    Order, 153
- avalon\_sdk.connector.interfaces.work\_order\_proxy.↵  
    WorkOrderProxy, 166
- avalon\_sdk.connector.interfaces.work\_order\_receipt.↵  
    WorkOrderReceipt, 168
- avalon\_sdk.connector.interfaces.worker\_registry.↵  
    WorkerRegistry, 139
- avalon\_sdk.connector.interfaces.worker\_registry\_list.↵  
    WorkerRegistryList, 144
- avalon\_sdk.http\_client.http\_jrpc\_client.HttpJrpcClient, 85
- avalon\_sdk.http\_client.http\_jrpc\_client.Message\_↵  
    Exception, 107
- avalon\_sdk.registry.registry\_status.RegistryStatus, 118
- avalon\_sdk.work\_order.work\_order\_params.Work\_↵  
    OrderParams, 158
- avalon\_sdk.work\_order.work\_order\_request\_validator.↵  
    WorkOrderRequestValidator, 175
- avalon\_sdk.work\_order\_receipt.work\_order\_receipt.↵  
    ReceiptCreateStatus, 116
- avalon\_sdk.work\_order\_receipt.work\_order\_receipt.↵  
    WorkOrderReceiptRequest, 174
- avalon\_sdk.worker.worker\_details.SGXWorkerDetails, 120
- avalon\_sdk.worker.worker\_details.WorkerDetails, 138
- avalon\_sdk.worker.worker\_details.WorkerStatus, 148
- avalon\_sdk.worker.worker\_details.WorkerType, 149
- avalon\_sdk::connector::blockchains::ethereum\_↵  
    ::ethereum\_listener::BlockchainInterface  
    newListener, 27
- avalon\_sdk::connector::blockchains::ethereum\_↵  
    ::ethereum\_listener::EventProcessor  
    get\_event\_synchronously, 58  
    handler, 58  
    listener, 58  
    start, 59  
    stop, 59  
    sync\_handler, 59
- avalon\_sdk::connector::blockchains::ethereum\_↵  
    ::ethereum\_work\_order::EthereumWork\_↵  
        OrderProxyImpl  
    encryption\_key\_get, 48  
    encryption\_key\_retrieve, 48  
    encryption\_key\_set, 48  
    encryption\_key\_start, 49  
    work\_order\_complete, 49  
    work\_order\_get\_result, 49  
    work\_order\_submit, 50
- avalon\_sdk::connector::blockchains::ethereum\_↵  
    ::ethereum\_worker\_registry::Ethereum\_↵  
        WorkerRegistryImpl  
    \_\_init\_\_, 39

- worker\_lookup, 39
- worker\_lookup\_next, 40
- worker\_register, 41
- worker\_retrieve, 41
- worker\_set\_status, 42
- worker\_update, 42
- avalon\_sdk::connector::blockchains::ethereum↔
  - ::ethereum\_worker\_registry\_list::Ethereum↔
    - WorkerRegistryListImpl
  - registry\_add, 44
  - registry\_lookup, 44
  - registry\_lookup\_next, 44
  - registry\_retrieve, 45
  - registry\_set\_status, 45
  - registry\_update, 46
- avalon\_sdk::connector::blockchains::ethereum↔
  - ::ethereum\_wrapper::EthereumWrapper
  - compile\_source\_file, 51
  - deploy\_contract, 51
  - execute\_transaction, 52
  - execute\_unsigned\_transaction, 52
  - get\_account\_address, 52
  - get\_bytes\_from\_hex, 53
  - get\_chain\_id, 53
  - get\_contract\_instance, 53
  - get\_contract\_instance\_from\_json, 53
  - get\_gas\_limit, 54
  - get\_gas\_price, 54
  - get\_transaction\_params, 54
  - get\_txn\_nonce, 54
  - sign\_execute\_raw\_transaction, 55
- avalon\_sdk::connector::blockchains::fabric::event↔
  - listener::EventListener
  - config, 56
  - get\_single\_event, 56
  - start\_event\_handling, 57
  - stop\_event\_handling, 57
- avalon\_sdk::connector::blockchains::fabric::fabric↔
  - work\_order::FabricWorkOrderImpl
  - \_\_init\_\_, 70
  - encryption\_key\_get, 71
  - encryption\_key\_set, 71
  - encryption\_key\_start, 71
  - get\_work\_order\_completed\_event\_handler, 72
  - get\_work\_order\_submitted\_event\_handler, 72
  - work\_order\_complete, 72
  - work\_order\_get\_result, 73
  - work\_order\_submit, 73
- avalon\_sdk::connector::blockchains::fabric::fabric↔
  - \_work\_order\_receipt::FabricWorkOrder↔
    - ReceiptImpl
  - \_\_init\_\_, 75
  - work\_order\_receipt\_create, 75
  - work\_order\_receipt\_lookup, 76
  - work\_order\_receipt\_lookup\_next, 76
  - work\_order\_receipt\_retrieve, 77
  - work\_order\_receipt\_update, 77
  - work\_order\_receipt\_update\_retrieve, 78
- avalon\_sdk::connector::blockchains::fabric::fabric↔
  - worker\_registry::FabricWorkerRegistryImpl
  - \_\_init\_\_, 61
  - worker\_lookup, 61
  - worker\_lookup\_next, 62
  - worker\_register, 62
  - worker\_retrieve, 63
  - worker\_set\_status, 63
  - worker\_update, 64
- avalon\_sdk::connector::blockchains::fabric::fabric↔
  - worker\_registry\_list::FabricWorkerRegistry↔
    - ListImpl
  - \_\_init\_\_, 66
  - registry\_add, 66
  - registry\_lookup, 66
  - registry\_lookup\_next, 67
  - registry\_retrieve, 67
  - registry\_set\_status, 68
  - registry\_update, 68
- avalon\_sdk::connector::blockchains::fabric::fabric↔
  - wrapper::FabricWrapper
  - \_\_init\_\_, 79
  - get\_event\_handler, 79
  - invoke\_chaincode, 79
- avalon\_sdk::connector::blockchains::fabric::tx↔
  - committer::TxCommitter
  - cc\_invoke, 135
  - cc\_query, 135
- avalon\_sdk::connector::direct::avalon\_direct\_client::↔
  - AvalonDirectClient
  - \_\_init\_\_, 26
  - get\_work\_order\_instance, 26
  - get\_work\_order\_receipt\_instance, 26
  - get\_worker\_registry\_instance, 26
  - get\_worker\_registry\_list\_instance, 27
- avalon\_sdk::connector::direct::jrpc::jrpc\_work\_order::↔
  - JRPCWorkOrderImpl
  - encryption\_key\_get, 93
  - encryption\_key\_set, 93
  - work\_order\_get\_result, 94
  - work\_order\_get\_result\_nonblocking, 94
  - work\_order\_submit, 95
- avalon\_sdk::connector::direct::jrpc::jrpc\_work\_order↔
  - receipt::JRPCWorkOrderReceiptImpl
  - work\_order\_receipt\_create, 96
  - work\_order\_receipt\_lookup, 97
  - work\_order\_receipt\_lookup\_next, 97
  - work\_order\_receipt\_retrieve, 98
  - work\_order\_receipt\_update, 98
  - work\_order\_receipt\_update\_retrieve, 99
- avalon\_sdk::connector::direct::jrpc::jrpc\_worker↔
  - registry::JRPCWorkerRegistryImpl
  - worker\_lookup, 89
  - worker\_lookup\_next, 89
  - worker\_register, 90
  - worker\_retrieve, 90
  - worker\_set\_status, 91
  - worker\_update, 91

avalon\_sdk::connector::interfaces::work\_order::WorkOrder  
     encryption\_key\_get, 154  
     encryption\_key\_set, 155  
     work\_order\_get\_result, 155  
     work\_order\_submit, 155  
 avalon\_sdk::connector::interfaces::work\_order\_proxy::WorkOrderProxy  
     encryption\_key\_start, 167  
     work\_order\_complete, 167  
 avalon\_sdk::connector::interfaces::work\_order\_receipt::WorkOrderReceipt  
     work\_order\_receipt\_create, 169  
     work\_order\_receipt\_lookup, 169  
     work\_order\_receipt\_lookup\_next, 170  
     work\_order\_receipt\_retrieve, 171  
     work\_order\_receipt\_update, 171, 172  
     work\_order\_receipt\_update\_retrieve, 173  
 avalon\_sdk::connector::interfaces::worker\_registry::WorkerRegistry  
     worker\_lookup, 140  
     worker\_lookup\_next, 141  
     worker\_register, 142  
     worker\_retrieve, 142  
     worker\_set\_status, 143  
     worker\_update, 143  
 avalon\_sdk::connector::interfaces::worker\_registry\_list::WorkerRegistryList  
     registry\_add, 145  
     registry\_lookup, 145  
     registry\_lookup\_next, 145  
     registry\_retrieve, 146  
     registry\_set\_status, 146  
     registry\_update, 147  
 avalon\_sdk::work\_order::work\_order\_params::WorkOrderParams  
     add\_encrypted\_request\_hash, 159  
     add\_in\_data, 159  
     add\_out\_data, 159  
     add\_requester\_signature, 160  
     get\_in\_data, 160  
     get\_out\_data, 160  
     get\_params, 160  
     get\_requester\_id, 161  
     get\_requester\_nonce, 161  
     get\_session\_key\_iv, 161  
     get\_work\_order\_id, 161  
     get\_worker\_id, 161  
     get\_workload\_id, 162  
     set\_data\_encryption\_algorithm, 162  
     set\_encrypted\_session\_key, 162  
     set\_notify\_uri, 162  
     set\_payload\_format, 162  
     set\_requester\_id, 163  
     set\_requester\_nonce, 163  
     set\_response\_timeout\_msecs, 163  
     set\_result\_uri, 163  
     set\_session\_key\_iv, 163  
     set\_verifying\_key, 164  
     set\_work\_order\_id, 164  
     set\_worker\_encryption\_key, 164  
     set\_worker\_id, 164  
     set\_workload\_id, 164  
     to\_jrpc\_string, 165  
     to\_string, 165  
 avalon\_sdk::work\_order::work\_order\_request\_validator::WorkOrderRequestValidator  
     \_\_init\_\_, 175  
     validate\_data\_format, 176  
     validate\_parameters, 176  
 avalon\_sdk::work\_order\_receipt::work\_order\_receipt\_request::WorkOrderReceiptRequest  
     create\_receipt, 174  
     update\_receipt, 174  
 avalon\_sdk::worker::worker\_details::SGXWorkerDetails  
     load\_worker, 121  
 avalon\_sdk::worker::worker\_details::WorkerDetails  
     \_\_init\_\_, 138  
     validate\_worker\_details, 139  
  
 base64.cpp  
     base64\_decode, 197  
     base64\_encode, 198  
 base64.h  
     base64\_decode, 199  
     base64\_encode, 199  
 base64\_decode  
     base64.cpp, 197  
     base64.h, 199  
 base64\_encode  
     base64.cpp, 198  
     base64.h, 199  
 Base64EncodedString  
     types.h, 205  
 Base64EncodedStringToByteArray  
     types.cpp, 203  
     types.h, 206  
 BinaryToHexString  
     hex\_string.h, 195  
 ByteArray  
     types.h, 205  
 ByteArrayToBase64EncodedString  
     types.cpp, 203  
     types.h, 206  
 ByteArrayToHexEncodedString  
     types.cpp, 203  
     types.h, 206  
 ByteArrayToStr  
     utils.cpp, 208  
     utils.h, 209  
 ByteArrayToString  
     types.cpp, 204  
     types.h, 206  
 ByteArrayToStringArray  
     types.cpp, 204  
     types.h, 207



- calculate\_datahash
  - avalon\_crypto\_utils::signature::ClientSignature, 30
- calculate\_request\_hash
  - avalon\_crypto\_utils::signature::ClientSignature, 30
- cc\_invoke
  - avalon\_sdk::connector::blockchains::fabric::tx\_↔  
committer::TxCommitter, 135
- cc\_query
  - avalon\_sdk::connector::blockchains::fabric::tx\_↔  
committer::TxCommitter, 135
- Clone
  - WorkloadProcessor, 151
- compile\_source\_file
  - avalon\_sdk::connector::blockchains::ethereum\_↔  
::ethereum\_wrapper::EthereumWrapper, 51
- ComputeMessageHash
  - crypto\_utils.h, 178
- config
  - avalon\_sdk::connector::blockchains::fabric\_↔  
::event\_listener::EventListener, 56
- create\_receipt
  - avalon\_sdk::work\_order\_receipt::work\_order\_↔  
receipt::WorkOrderReceiptRequest, 174
- CreateHexEncodedEncryptionKey
  - crypto\_utils.h, 178
- CreateWorkloadProcessor
  - WorkloadProcessor, 151
- crypto\_utils.h
  - ComputeMessageHash, 178
  - CreateHexEncodedEncryptionKey, 178
  - DecryptData, 178
  - EncryptData, 179
  - RandomBitString, 179
- DecryptData
  - crypto\_utils.h, 178
- DecryptMessage
  - tcf::crypto::pkenc::PrivateKey, 109
  - tcf::crypto::skenc, 22, 23
- Delete
  - file\_io\_wrapper.cpp, 213
  - file\_io\_wrapper.h, 215
- deploy\_contract
  - avalon\_sdk::connector::blockchains::ethereum\_↔  
::ethereum\_wrapper::EthereumWrapper, 51
- Deserialize
  - tcf::crypto::pkenc::PrivateKey, 110
  - tcf::crypto::pkenc::PublicKey, 115
  - tcf::crypto::sig::PrivateKey, 111
  - tcf::crypto::sig::PublicKey, 113
- deserializeECDSAPrivateKey
  - sig\_private\_key.cpp, 187
- deserializeECDSAPublicKey
  - sig\_public\_key.cpp, 189
- deserializeRSAPrivateKey
  - pkenc\_private\_key.cpp, 180
- deserializeRSAPublicKey
  - pkenc\_public\_key.cpp, 182
- DeserializeXYFromHex
  - tcf::crypto::sig::PublicKey, 113
- EnclaveType
  - types.h, 206
- EncryptData
  - crypto\_utils.h, 179
- EncryptMessage
  - tcf::crypto::pkenc::PublicKey, 115
  - tcf::crypto::skenc, 23
- encryption\_key\_get
  - avalon\_sdk::connector::blockchains::ethereum\_↔  
::ethereum\_work\_order::EthereumWork\_↔  
OrderProxyImpl, 48
  - avalon\_sdk::connector::blockchains::fabric\_↔  
::fabric\_work\_order::FabricWorkOrderImpl, 71
  - avalon\_sdk::connector::direct::jrpc::jrpc\_work\_↔  
order::JRPCWorkOrderImpl, 93
  - avalon\_sdk::connector::interfaces::work\_order::↔  
WorkOrder, 154
- encryption\_key\_retrieve
  - avalon\_sdk::connector::blockchains::ethereum\_↔  
::ethereum\_work\_order::EthereumWork\_↔  
OrderProxyImpl, 48
- encryption\_key\_set
  - avalon\_sdk::connector::blockchains::ethereum\_↔  
::ethereum\_work\_order::EthereumWork\_↔  
OrderProxyImpl, 48
  - avalon\_sdk::connector::blockchains::fabric\_↔  
::fabric\_work\_order::FabricWorkOrderImpl, 71
  - avalon\_sdk::connector::direct::jrpc::jrpc\_work\_↔  
order::JRPCWorkOrderImpl, 93
  - avalon\_sdk::connector::interfaces::work\_order::↔  
WorkOrder, 155
- encryption\_key\_start
  - avalon\_sdk::connector::blockchains::ethereum\_↔  
::ethereum\_work\_order::EthereumWork\_↔  
OrderProxyImpl, 49
  - avalon\_sdk::connector::blockchains::fabric\_↔  
::fabric\_work\_order::FabricWorkOrderImpl, 71
  - avalon\_sdk::connector::interfaces::work\_order\_↔  
proxy::WorkOrderProxy, 167
- execute\_transaction
  - avalon\_sdk::connector::blockchains::ethereum\_↔  
::ethereum\_wrapper::EthereumWrapper, 52
- execute\_unsigned\_transaction
  - avalon\_sdk::connector::blockchains::ethereum\_↔  
::ethereum\_wrapper::EthereumWrapper, 52
- file\_io\_wrapper.cpp
  - Delete, 213
  - Read, 214
  - Write, 214
- file\_io\_wrapper.h
  - Delete, 215
  - Read, 216
  - Write, 216

FileClose  
     FileIoExecutor, 80  
 FileDelete  
     FileIoExecutor, 81  
 FileIoExecutor, 80  
     FileClose, 80  
     FileDelete, 81  
     FileOpen, 81  
     FileRead, 81  
     FileSeek, 82  
     FileTell, 82  
     FileWrite, 83  
     GetIoHandlerId, 83  
     GetMaxFileSize, 84  
     GetMaxIoResultSize, 84  
 FileOpen  
     FileIoExecutor, 81  
 FileRead  
     FileIoExecutor, 81  
 FileSeek  
     FileIoExecutor, 82  
 FileTell  
     FileIoExecutor, 82  
 FileWrite  
     FileIoExecutor, 83  
  
 Generate  
     tcf::crypto::pkenc::PrivateKey, 110  
     tcf::crypto::sig::PrivateKey, 111  
 generate\_client\_signature  
     avalon\_crypto\_utils::signature::ClientSignature, 30  
 generate\_signature  
     avalon\_crypto\_utils::signature::ClientSignature, 31  
 GenerateIV  
     tcf::crypto::skenc, 23  
 GenerateKey  
     tcf::crypto::skenc, 23  
 get\_account\_address  
     avalon\_sdk::connector::blockchains::ethereum↔  
         ::ethereum\_wrapper::EthereumWrapper, 52  
 get\_bytes\_from\_hex  
     avalon\_sdk::connector::blockchains::ethereum↔  
         ::ethereum\_wrapper::EthereumWrapper, 53  
 get\_chain\_id  
     avalon\_sdk::connector::blockchains::ethereum↔  
         ::ethereum\_wrapper::EthereumWrapper, 53  
 get\_contract\_instance  
     avalon\_sdk::connector::blockchains::ethereum↔  
         ::ethereum\_wrapper::EthereumWrapper, 53  
 get\_contract\_instance\_from\_json  
     avalon\_sdk::connector::blockchains::ethereum↔  
         ::ethereum\_wrapper::EthereumWrapper, 53  
 get\_event\_handler  
     avalon\_sdk::connector::blockchains::fabric↔  
         ::fabric\_wrapper::FabricWrapper, 79  
 get\_event\_synchronously  
     avalon\_sdk::connector::blockchains::ethereum↔  
         ::ethereum\_listener::EventProcessor, 58  
 get\_gas\_limit  
     avalon\_sdk::connector::blockchains::ethereum↔  
         ::ethereum\_wrapper::EthereumWrapper, 54  
 get\_gas\_price  
     avalon\_sdk::connector::blockchains::ethereum↔  
         ::ethereum\_wrapper::EthereumWrapper, 54  
 get\_in\_data  
     avalon\_sdk::work\_order::work\_order\_params::↔  
         WorkOrderParams, 160  
 get\_out\_data  
     avalon\_sdk::work\_order::work\_order\_params::↔  
         WorkOrderParams, 160  
 get\_params  
     avalon\_sdk::work\_order::work\_order\_params::↔  
         WorkOrderParams, 160  
 get\_requester\_id  
     avalon\_sdk::work\_order::work\_order\_params::↔  
         WorkOrderParams, 161  
 get\_requester\_nonce  
     avalon\_sdk::work\_order::work\_order\_params::↔  
         WorkOrderParams, 161  
 get\_session\_key\_iv  
     avalon\_sdk::work\_order::work\_order\_params::↔  
         WorkOrderParams, 161  
 get\_single\_event  
     avalon\_sdk::connector::blockchains::fabric↔  
         ::event\_listener::EventListener, 56  
 get\_transaction\_params  
     avalon\_sdk::connector::blockchains::ethereum↔  
         ::ethereum\_wrapper::EthereumWrapper, 54  
 get\_txn\_nonce  
     avalon\_sdk::connector::blockchains::ethereum↔  
         ::ethereum\_wrapper::EthereumWrapper, 54  
 get\_work\_order\_completed\_event\_handler  
     avalon\_sdk::connector::blockchains::fabric↔  
         ::fabric\_work\_order::FabricWorkOrderImpl,  
         72  
 get\_work\_order\_id  
     avalon\_sdk::work\_order::work\_order\_params::↔  
         WorkOrderParams, 161  
 get\_work\_order\_instance  
     avalon\_sdk::connector::direct::avalon\_direct\_↔  
         client::AvalonDirectClient, 26  
 get\_work\_order\_receipt\_instance  
     avalon\_sdk::connector::direct::avalon\_direct\_↔  
         client::AvalonDirectClient, 26  
 get\_work\_order\_submitted\_event\_handler  
     avalon\_sdk::connector::blockchains::fabric↔  
         ::fabric\_work\_order::FabricWorkOrderImpl,  
         72  
 get\_worker\_id  
     avalon\_sdk::work\_order::work\_order\_params::↔  
         WorkOrderParams, 161  
 get\_worker\_registry\_instance  
     avalon\_sdk::connector::direct::avalon\_direct\_↔  
         client::AvalonDirectClient, 26  
 get\_worker\_registry\_list\_instance  
     avalon\_sdk::connector::direct::avalon\_direct\_↔  
         client::AvalonDirectClient, 27

- get\_workload\_id
  - avalon\_sdk::work\_order::work\_order\_params::↔  
WorkOrderParams, 162
- GetIoHandlerId
  - FileIoExecutor, 83
- GetMaxFileSize
  - FileIoExecutor, 84
- GetMaxIoResultSize
  - FileIoExecutor, 84
- GetPublicKey
  - tcf::crypto::pkenc::PrivateKey, 110
  - tcf::crypto::sig::PrivateKey, 111
- HEX\_STRING\_SIZE
  - hex\_string.h, 195
- handler
  - avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_listener::EventProcessor, 58
- hex\_string.h
  - BinaryToHexString, 195
  - HEX\_STRING\_SIZE, 195
  - HexStringToBinary, 195
- HexEncodedString
  - types.h, 206
- HexEncodedStringToByteArray
  - types.cpp, 204
  - types.h, 207
- HexStringToBinary
  - hex\_string.h, 195
- IMPL\_WORKLOAD\_PROCESSOR\_CLONE
  - workload\_processor.h, 221
- IV\_LEN
  - tcf::crypto::constants, 21
- invoke\_chaincode
  - avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_wrapper::FabricWrapper, 79
- json\_array\_t, 100
- json\_object\_t, 101
- json\_value\_t, 102
- json\_value\_value, 103
- JsonValue, 105
- listener
  - avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_listener::EventProcessor, 58
- load\_worker
  - avalon\_sdk::worker::worker\_details::SGXWorker↔  
Details, 121
- newListener
  - avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_listener::BlockchainInterface, 27
- operator=
  - tcf::crypto::pkenc::PrivateKey, 110
  - tcf::crypto::pkenc::PublicKey, 116
  - tcf::crypto::sig::PrivateKey, 112
  - tcf::crypto::sig::PublicKey, 113
- pkenc\_private\_key.cpp
  - deserializeRSAPrivateKey, 180
- pkenc\_public\_key.cpp
  - deserializeRSAPublicKey, 182
- PrivateKey
  - tcf::crypto::pkenc::PrivateKey, 109
  - tcf::crypto::sig::PrivateKey, 111
- ProcessWorkOrder
  - WorkloadProcessor, 152
- PublicKey
  - tcf::crypto::pkenc::PublicKey, 115
  - tcf::crypto::sig::PublicKey, 113
- REGISTER\_WORKLOAD\_PROCESSOR
  - workload\_processor.h, 221
- RSA\_PLAINTEXT\_LEN
  - tcf::crypto::constants, 21
- RandomBitString
  - crypto\_utils.h, 179
- Read
  - file\_io\_wrapper.cpp, 214
  - file\_io\_wrapper.h, 216
- RegisterWorkloadProcessor
  - WorkloadProcessor, 152
- registry\_add
  - avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_worker\_registry\_list::Ethereum↔  
WorkerRegistryListImpl, 44
  - avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_worker\_registry\_list::FabricWorker↔  
RegistryListImpl, 66
  - avalon\_sdk::connector::interfaces::worker↔  
registry\_list::WorkerRegistryList, 145
- registry\_lookup
  - avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_worker\_registry\_list::Ethereum↔  
WorkerRegistryListImpl, 44
  - avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_worker\_registry\_list::FabricWorker↔  
RegistryListImpl, 66
  - avalon\_sdk::connector::interfaces::worker↔  
registry\_list::WorkerRegistryList, 145
- registry\_lookup\_next
  - avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_worker\_registry\_list::Ethereum↔  
WorkerRegistryListImpl, 44
  - avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_worker\_registry\_list::FabricWorker↔  
RegistryListImpl, 67
  - avalon\_sdk::connector::interfaces::worker↔  
registry\_list::WorkerRegistryList, 145
- registry\_retrieve
  - avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_worker\_registry\_list::Ethereum↔  
WorkerRegistryListImpl, 45
  - avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_worker\_registry\_list::FabricWorker↔  
RegistryListImpl, 67

- avalon\_sdk::connector::interfaces::worker\_↔  
registry\_list::WorkerRegistryList, 146
- registry\_set\_status
  - avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_worker\_registry\_list::Ethereum↔  
WorkerRegistryListImpl, 45
  - avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_worker\_registry\_list::FabricWorker↔  
RegistryListImpl, 68
  - avalon\_sdk::connector::interfaces::worker\_↔  
registry\_list::WorkerRegistryList, 146
- registry\_update
  - avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_worker\_registry\_list::Ethereum↔  
WorkerRegistryListImpl, 46
  - avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_worker\_registry\_list::FabricWorker↔  
RegistryListImpl, 68
  - avalon\_sdk::connector::interfaces::worker\_↔  
registry\_list::WorkerRegistryList, 147
- SAVE/verify\_certificate.h
  - verify\_certificate\_chain, 183
- SAVE/verify\_signature.h
  - verify\_signature, 185
- SYM\_KEY\_LEN
  - tcf::crypto::constants, 22
- Serialize
  - tcf::crypto::pkenc::PrivateKey, 110
  - tcf::crypto::pkenc::PublicKey, 116
  - tcf::crypto::sig::PrivateKey, 112
  - tcf::crypto::sig::PublicKey, 113
- SerializeXYToHex
  - tcf::crypto::sig::PublicKey, 114
- set\_data\_encryption\_algorithm
  - avalon\_sdk::work\_order::work\_order\_params::↔  
WorkOrderParams, 162
- set\_encrypted\_session\_key
  - avalon\_sdk::work\_order::work\_order\_params::↔  
WorkOrderParams, 162
- set\_notify\_uri
  - avalon\_sdk::work\_order::work\_order\_params::↔  
WorkOrderParams, 162
- set\_payload\_format
  - avalon\_sdk::work\_order::work\_order\_params::↔  
WorkOrderParams, 162
- set\_requester\_id
  - avalon\_sdk::work\_order::work\_order\_params::↔  
WorkOrderParams, 163
- set\_requester\_nonce
  - avalon\_sdk::work\_order::work\_order\_params::↔  
WorkOrderParams, 163
- set\_response\_timeout\_msecs
  - avalon\_sdk::work\_order::work\_order\_params::↔  
WorkOrderParams, 163
- set\_result\_uri
  - avalon\_sdk::work\_order::work\_order\_params::↔  
WorkOrderParams, 163
- set\_session\_key\_iv
  - avalon\_sdk::work\_order::work\_order\_params::↔  
WorkOrderParams, 163
- set\_verifying\_key
  - avalon\_sdk::work\_order::work\_order\_params::↔  
WorkOrderParams, 164
- set\_work\_order\_id
  - avalon\_sdk::work\_order::work\_order\_params::↔  
WorkOrderParams, 164
- set\_worker\_encryption\_key
  - avalon\_sdk::work\_order::work\_order\_params::↔  
WorkOrderParams, 164
- set\_worker\_id
  - avalon\_sdk::work\_order::work\_order\_params::↔  
WorkOrderParams, 164
- set\_workload\_id
  - avalon\_sdk::work\_order::work\_order\_params::↔  
WorkOrderParams, 164
- sig\_private\_key.cpp
  - deserializeECDSAPrivateKey, 187
- sig\_public\_key.cpp
  - deserializeECDSAPublicKey, 189
- sign\_execute\_raw\_transaction
  - avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_wrapper::EthereumWrapper, 55
- SignMessage
  - tcf::crypto::sig::PrivateKey, 112
- start
  - avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_listener::EventProcessor, 59
- start\_event\_handling
  - avalon\_sdk::connector::blockchains::fabric↔  
::event\_listener::EventListener, 57
- stop
  - avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_listener::EventProcessor, 59
- stop\_event\_handling
  - avalon\_sdk::connector::blockchains::fabric↔  
::event\_listener::EventListener, 57
- StrToByteArray
  - utils.cpp, 208
  - utils.h, 209
- StringArray, 121
- sync\_handler
  - avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_listener::EventProcessor, 59
- TAG\_LEN
  - tcf::crypto::constants, 22
- tcf::WorkOrderData, 158
- tcf::crypto::constants, 21
  - IV\_LEN, 21
  - RSA\_PLAINTEXT\_LEN, 21
  - SYM\_KEY\_LEN, 22
  - TAG\_LEN, 22
- tcf::crypto::pkenc::PrivateKey, 109
  - ~PrivateKey, 109
  - DecryptMessage, 109
  - Deserialize, 110
  - Generate, 110

- GetPublicKey, 110
- operator=, 110
- PrivateKey, 109
- Serialize, 110
- tcf::crypto::pkenc::PublicKey, 114
  - ~PublicKey, 115
  - Deserialize, 115
  - EncryptMessage, 115
  - operator=, 116
  - PublicKey, 115
  - Serialize, 116
- tcf::crypto::sig::PrivateKey, 111
  - ~PrivateKey, 111
  - Deserialize, 111
  - Generate, 111
  - GetPublicKey, 111
  - operator=, 112
  - PrivateKey, 111
  - Serialize, 112
  - SignMessage, 112
- tcf::crypto::sig::PublicKey, 112
  - ~PublicKey, 113
  - Deserialize, 113
  - DeserializeXYFromHex, 113
  - operator=, 113
  - PublicKey, 113
  - Serialize, 113
  - SerializeXYToHex, 114
  - VerifySignature, 114
- tcf::crypto::skenc, 22
  - DecryptMessage, 22, 23
  - EncryptMessage, 23
  - GenerateIV, 23
  - GenerateKey, 23
- tcf::error::CryptoError, 33
- tcf::error::DivisionByZero, 34
- tcf::error::Error, 37
- tcf::error::IOError, 87
- tcf::error::IndexError, 86
- tcf::error::MemoryError, 106
- tcf::error::OverflowError, 108
- tcf::error::RuntimeError, 119
- tcf::error::SystemBusyError, 122
- tcf::error::SystemError, 123
- tcf::error::UnknownError, 136
- tcf::error::ValueError, 137
- tcf::error::WorkloadError, 150
- tcf::utility::Timer, 133
- tcf\_err\_t
  - tcf\_error.h, 201
- tcf\_error.h
  - tcf\_err\_t, 201
- test\_ethereum\_work\_order\_impl.TestEthereumWork←
  - OrderProxyImpl, 126
- test\_ethereum\_work\_order\_impl::TestEthereumWork←
  - OrderProxyImpl
    - test\_is\_wo\_id\_in\_event\_error\_result, 127
    - test\_is\_wo\_id\_in\_event\_positive, 127
  - test\_is\_wo\_id\_in\_event\_wo\_id\_not\_matched, 128
  - test\_work\_order\_complete, 128
  - test\_work\_order\_complete\_error, 128
- test\_ethereum\_worker\_registry\_impl.TestEthereum←
  - WorkerRegistryImpl, 124
- test\_ethereum\_worker\_registry\_list\_impl.TestEthereum←
  - WorkerRegistryListImpl, 125
- test\_fabric\_worker\_registry\_impl.TestFabricWorker←
  - RegistryImpl, 129
- test\_fabric\_worker\_registry\_list\_impl.TestFabric←
  - WorkerRegistryListImpl, 130
- test\_is\_wo\_id\_in\_event\_error\_result
  - test\_ethereum\_work\_order\_impl::TestEthereum←
    - WorkOrderProxyImpl, 127
- test\_is\_wo\_id\_in\_event\_positive
  - test\_ethereum\_work\_order\_impl::TestEthereum←
    - WorkOrderProxyImpl, 127
- test\_is\_wo\_id\_in\_event\_wo\_id\_not\_matched
  - test\_ethereum\_work\_order\_impl::TestEthereum←
    - WorkOrderProxyImpl, 128
- test\_work\_order\_complete
  - test\_ethereum\_work\_order\_impl::TestEthereum←
    - WorkOrderProxyImpl, 128
- test\_work\_order\_complete\_error
  - test\_ethereum\_work\_order\_impl::TestEthereum←
    - WorkOrderProxyImpl, 128
- test\_work\_order\_encryption\_key\_jrpc\_impl.TestWork←
  - OrderEncryptionKeyJRPCImpl, 132
- test\_work\_order\_jrpc\_impl.TestWorkOrderJRPCImpl, 133
- test\_worker\_registry\_jrpc\_impl.TestWorkerRegistryJ←
  - RPCImpl, 131
- to\_jrpc\_string
  - avalon\_sdk::work\_order::work\_order\_params::←
    - WorkOrderParams, 165
- to\_string
  - avalon\_sdk::work\_order::work\_order\_params::←
    - WorkOrderParams, 165
- types.cpp
  - Base64EncodedStringToByteArray, 203
  - ByteArrayToBase64EncodedString, 203
  - ByteArrayToHexEncodedString, 203
  - ByteArrayToString, 204
  - ByteArrayToStringArray, 204
  - HexEncodedStringToByteArray, 204
- types.h
  - Base64EncodedString, 205
  - Base64EncodedStringToByteArray, 206
  - ByteArray, 205
  - ByteArrayToBase64EncodedString, 206
  - ByteArrayToHexEncodedString, 206
  - ByteArrayToString, 206
  - ByteArrayToStringArray, 207
  - EnclaveType, 206
  - HexEncodedString, 206
  - HexEncodedStringToByteArray, 207
- update\_receipt

- avalon\_sdk::work\_order\_receipt::work\_order\_↔  
receipt::WorkOrderReceiptRequest, 174
- utils.cpp
  - ByteArrayToStr, 208
  - StrToByteArray, 208
- utils.h
  - ByteArrayToStr, 209
  - StrToByteArray, 209
- validate\_data\_format
  - avalon\_sdk::work\_order::work\_order\_request\_↔  
validator::WorkOrderRequestValidator, 176
- validate\_parameters
  - avalon\_sdk::work\_order::work\_order\_request\_↔  
validator::WorkOrderRequestValidator, 176
- validate\_worker\_details
  - avalon\_sdk::worker::worker\_details::Worker↔  
Details, 139
- verify\_certificate.h
  - verify\_certificate\_chain, 184
- verify\_certificate\_chain
  - SAVE/verify\_certificate.h, 183
  - verify\_certificate.h, 184
- verify\_create\_receipt\_signature
  - avalon\_crypto\_utils::signature::ClientSignature, 31
- verify\_signature
  - avalon\_crypto\_utils::signature::ClientSignature, 31
  - SAVE/verify\_signature.h, 185
  - verify\_signature.h, 186
- verify\_signature.h
  - verify\_signature, 186
- verify\_update\_receipt\_signature
  - avalon\_crypto\_utils::signature::ClientSignature, 32
- VerifySignature
  - tcf::crypto::sig::PublicKey, 114
- work\_order\_complete
  - avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_work\_order::EthereumWork↔  
OrderProxyImpl, 49
  - avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_work\_order::FabricWorkOrderImpl,  
72
  - avalon\_sdk::connector::interfaces::work\_order\_↔  
proxy::WorkOrderProxy, 167
- work\_order\_get\_result
  - avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_work\_order::EthereumWork↔  
OrderProxyImpl, 49
  - avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_work\_order::FabricWorkOrderImpl,  
73
  - avalon\_sdk::connector::direct::jrpc::jrpc\_work\_↔  
order::JRPCWorkOrderImpl, 94
  - avalon\_sdk::connector::interfaces::work\_order::↔  
WorkOrder, 155
- work\_order\_get\_result\_nonblocking
  - avalon\_sdk::connector::direct::jrpc::jrpc\_work\_↔  
order::JRPCWorkOrderImpl, 94
- work\_order\_receipt\_create
  - avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_work\_order\_receipt::FabricWork↔  
OrderReceiptImpl, 75
  - avalon\_sdk::connector::direct::jrpc::jrpc\_work\_↔  
order\_receipt::JRPCWorkOrderReceiptImpl,  
96
  - avalon\_sdk::connector::interfaces::work\_order\_↔  
receipt::WorkOrderReceipt, 169
- work\_order\_receipt\_lookup
  - avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_work\_order\_receipt::FabricWork↔  
OrderReceiptImpl, 76
  - avalon\_sdk::connector::direct::jrpc::jrpc\_work\_↔  
order\_receipt::JRPCWorkOrderReceiptImpl,  
97
  - avalon\_sdk::connector::interfaces::work\_order\_↔  
receipt::WorkOrderReceipt, 169
- work\_order\_receipt\_lookup\_next
  - avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_work\_order\_receipt::FabricWork↔  
OrderReceiptImpl, 76
  - avalon\_sdk::connector::direct::jrpc::jrpc\_work\_↔  
order\_receipt::JRPCWorkOrderReceiptImpl,  
97
  - avalon\_sdk::connector::interfaces::work\_order\_↔  
receipt::WorkOrderReceipt, 170
- work\_order\_receipt\_retrieve
  - avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_work\_order\_receipt::FabricWork↔  
OrderReceiptImpl, 77
  - avalon\_sdk::connector::direct::jrpc::jrpc\_work\_↔  
order\_receipt::JRPCWorkOrderReceiptImpl,  
98
  - avalon\_sdk::connector::interfaces::work\_order\_↔  
receipt::WorkOrderReceipt, 171
- work\_order\_receipt\_update
  - avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_work\_order\_receipt::FabricWork↔  
OrderReceiptImpl, 77
  - avalon\_sdk::connector::direct::jrpc::jrpc\_work\_↔  
order\_receipt::JRPCWorkOrderReceiptImpl,  
98
  - avalon\_sdk::connector::interfaces::work\_order\_↔  
receipt::WorkOrderReceipt, 171, 172
- work\_order\_receipt\_update\_retrieve
  - avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_work\_order\_receipt::FabricWork↔  
OrderReceiptImpl, 78
  - avalon\_sdk::connector::direct::jrpc::jrpc\_work\_↔  
order\_receipt::JRPCWorkOrderReceiptImpl,  
99
  - avalon\_sdk::connector::interfaces::work\_order\_↔  
receipt::WorkOrderReceipt, 173
- work\_order\_submit
  - avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_work\_order::EthereumWork↔  
OrderProxyImpl, 50



- avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_work\_order::FabricWorkOrderImpl,  
73
- avalon\_sdk::connector::direct::jrpc::jrpc\_work\_order↔  
order::JRPCWorkOrderImpl, 95
- avalon\_sdk::connector::interfaces::work\_order↔  
WorkOrder, 155
- worker\_lookup
  - avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_worker\_registry::Ethereum↔  
WorkerRegistryImpl, 39
  - avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_worker\_registry::FabricWorker↔  
RegistryImpl, 61
  - avalon\_sdk::connector::direct::jrpc::jrpc\_worker↔  
registry::JRPCWorkerRegistryImpl, 89
  - avalon\_sdk::connector::interfaces::worker↔  
registry::WorkerRegistry, 140
- worker\_lookup\_next
  - avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_worker\_registry::Ethereum↔  
WorkerRegistryImpl, 40
  - avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_worker\_registry::FabricWorker↔  
RegistryImpl, 62
  - avalon\_sdk::connector::direct::jrpc::jrpc\_worker↔  
registry::JRPCWorkerRegistryImpl, 89
  - avalon\_sdk::connector::interfaces::worker↔  
registry::WorkerRegistry, 141
- worker\_register
  - avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_worker\_registry::Ethereum↔  
WorkerRegistryImpl, 41
  - avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_worker\_registry::FabricWorker↔  
RegistryImpl, 62
  - avalon\_sdk::connector::direct::jrpc::jrpc\_worker↔  
registry::JRPCWorkerRegistryImpl, 90
  - avalon\_sdk::connector::interfaces::worker↔  
registry::WorkerRegistry, 142
- worker\_retrieve
  - avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_worker\_registry::Ethereum↔  
WorkerRegistryImpl, 41
  - avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_worker\_registry::FabricWorker↔  
RegistryImpl, 63
  - avalon\_sdk::connector::direct::jrpc::jrpc\_worker↔  
registry::JRPCWorkerRegistryImpl, 90
  - avalon\_sdk::connector::interfaces::worker↔  
registry::WorkerRegistry, 142
- worker\_set\_status
  - avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_worker\_registry::Ethereum↔  
WorkerRegistryImpl, 42
  - avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_worker\_registry::FabricWorker↔  
RegistryImpl, 63
- avalon\_sdk::connector::direct::jrpc::jrpc\_worker↔  
registry::JRPCWorkerRegistryImpl, 91
- avalon\_sdk::connector::interfaces::worker↔  
registry::WorkerRegistry, 143
- worker\_update
  - avalon\_sdk::connector::blockchains::ethereum↔  
::ethereum\_worker\_registry::Ethereum↔  
WorkerRegistryImpl, 42
  - avalon\_sdk::connector::blockchains::fabric↔  
::fabric\_worker\_registry::FabricWorker↔  
RegistryImpl, 64
  - avalon\_sdk::connector::direct::jrpc::jrpc\_worker↔  
registry::JRPCWorkerRegistryImpl, 91
  - avalon\_sdk::connector::interfaces::worker↔  
registry::WorkerRegistry, 143
- workload\_processor.h
  - IMPL\_WORKLOAD\_PROCESSOR\_CLONE, 221
  - REGISTER\_WORKLOAD\_PROCESSOR, 221
- workload\_processor\_table
  - WorkloadProcessor, 153
- WorkloadProcessor, 151
  - Clone, 151
  - CreateWorkloadProcessor, 151
  - ProcessWorkOrder, 152
  - RegisterWorkloadProcessor, 152
  - workload\_processor\_table, 153
- Write
  - file\_io\_wrapper.cpp, 214
  - file\_io\_wrapper.h, 216