

# Mobile-C Reference Manual

1.10.1

Generated by Doxygen 1.5.4

Tue Jul 1 15:29:57 2008



# Contents

<b>1</b>	<b>MobileC</b>	<b>1</b>
1.1	Introduction . . . . .	1
<b>2</b>	<b>Mobile-C Namespace Index</b>	<b>3</b>
2.1	Mobile-C Namespace List . . . . .	3
<b>3</b>	<b>Mobile-C Data Structure Index</b>	<b>5</b>
3.1	Mobile-C Data Structures . . . . .	5
<b>4</b>	<b>Mobile-C File Index</b>	<b>7</b>
4.1	Mobile-C File List . . . . .	7
<b>5</b>	<b>Mobile-C Page Index</b>	<b>11</b>
5.1	Mobile-C Related Pages . . . . .	11
<b>6</b>	<b>Mobile-C Namespace Documentation</b>	<b>13</b>
6.1	LibMC Namespace Reference . . . . .	13
6.2	LibMC::Properties Namespace Reference . . . . .	14
6.3	System Namespace Reference . . . . .	15
6.4	System::Collections::Generic Namespace Reference . . . . .	16
6.5	System::Reflection Namespace Reference . . . . .	17
6.6	System::Runtime::CompilerServices Namespace Reference . . . . .	18
6.7	System::Runtime::InteropServices Namespace Reference . . . . .	19
6.8	System::Text Namespace Reference . . . . .	20
<b>7</b>	<b>Mobile-C Data Structure Documentation</b>	<b>21</b>
7.1	_hr_time Struct Reference . . . . .	21
7.2	_x509_buf Struct Reference . . . . .	22
7.3	_x509_cert Struct Reference . . . . .	23
7.4	_x509_name Struct Reference . . . . .	27

7.5	<a href="#">_x509_time Struct Reference</a>	28
7.6	<a href="#">aes_context Struct Reference</a>	30
7.7	<a href="#">agency_s Struct Reference</a>	31
7.8	<a href="#">agent_datastate_s Struct Reference</a>	34
7.9	<a href="#">agent_mailbox_s Struct Reference</a>	37
7.10	<a href="#">agent_s Struct Reference</a>	38
7.11	<a href="#">agent_task_s Struct Reference</a>	43
7.12	<a href="#">arc4_context Struct Reference</a>	46
7.13	<a href="#">barrier_node_s Struct Reference</a>	47
7.14	<a href="#">barrier_queue_s Struct Reference</a>	48
7.15	<a href="#">cmd_prompt_s Struct Reference</a>	49
7.16	<a href="#">command_s Struct Reference</a>	50
7.17	<a href="#">connection_s Struct Reference</a>	51
7.18	<a href="#">des3_context Struct Reference</a>	53
7.19	<a href="#">des_context Struct Reference</a>	54
7.20	<a href="#">dhm_context Struct Reference</a>	55
7.21	<a href="#">dynstring_s Struct Reference</a>	57
7.22	<a href="#">fipa_acl_envelope_Received_s Struct Reference</a>	58
7.23	<a href="#">fipa_acl_envelope_s Struct Reference</a>	60
7.24	<a href="#">fipa_acl_message_s Struct Reference</a>	61
7.25	<a href="#">fipa_acl_Param_s Struct Reference</a>	64
7.26	<a href="#">fipa_agent_identifier_s Struct Reference</a>	66
7.27	<a href="#">fipa_agent_identifier_set_s Struct Reference</a>	67
7.28	<a href="#">fipa_DateTime_s Struct Reference</a>	68
7.29	<a href="#">fipa_expression_s Struct Reference</a>	70
7.30	<a href="#">fipa_expression_s::content_u Union Reference</a>	71
7.31	<a href="#">fipa_message_string_s Struct Reference</a>	72
7.32	<a href="#">fipa_number_s Struct Reference</a>	73
7.33	<a href="#">fipa_string_s Struct Reference</a>	74
7.34	<a href="#">fipa_url_s Struct Reference</a>	75
7.35	<a href="#">fipa_url_sequence_s Struct Reference</a>	76
7.36	<a href="#">fipa_word_s Struct Reference</a>	77
7.37	<a href="#">foo_c Class Reference</a>	78
7.38	<a href="#">foo_s Struct Reference</a>	81
7.39	<a href="#">havege_state Struct Reference</a>	83
7.40	<a href="#">host_id_s Struct Reference</a>	85

7.41 hr_time Struct Reference . . . . .	86
7.42 interpreter_variable_data_s Struct Reference . . . . .	87
7.43 list_s Struct Reference . . . . .	89
7.44 listNode_s Struct Reference . . . . .	90
7.45 mc_platform_s Struct Reference . . . . .	91
7.46 mc_rwlock_s Struct Reference . . . . .	96
7.47 LibMC::MCAclMessage Class Reference . . . . .	98
7.48 LibMC::MCAgency Class Reference . . . . .	104
7.49 LibMC::MCAgency::ChOptions_t Struct Reference . . . . .	132
7.50 LibMC::MCAgency::MCAgency_t Struct Reference . . . . .	133
7.51 LibMC::MCAgency::MCAgencyOptions_t Struct Reference . . . . .	135
7.52 MCAgencyOptions_s Struct Reference . . . . .	136
7.53 LibMC::MCAGENT Class Reference . . . . .	138
7.54 md2_context Struct Reference . . . . .	148
7.55 md4_context Struct Reference . . . . .	150
7.56 md5_context Struct Reference . . . . .	152
7.57 message_s Struct Reference . . . . .	154
7.58 mpi Struct Reference . . . . .	157
7.59 mtp_http_content_s Struct Reference . . . . .	159
7.60 mtp_http_s Struct Reference . . . . .	160
7.61 mxml_attr_s Struct Reference . . . . .	163
7.62 mxml_custom_s Struct Reference . . . . .	164
7.63 mxml_fdbuf_s Struct Reference . . . . .	165
7.64 mxml_index_s Struct Reference . . . . .	166
7.65 mxml_node_s Struct Reference . . . . .	168
7.66 mxml_text_s Struct Reference . . . . .	170
7.67 mxml_value_s Struct Reference . . . . .	171
7.68 mxml_value_u Union Reference . . . . .	172
7.69 rsa_context Struct Reference . . . . .	174
7.70 LibMC::Properties::Settings Class Reference . . . . .	177
7.71 sha1_context Struct Reference . . . . .	179
7.72 sha2_context Struct Reference . . . . .	181
7.73 sha4_context Struct Reference . . . . .	183
7.74 ssl_context Struct Reference . . . . .	185
7.75 syncList_s Struct Reference . . . . .	196
7.76 syncListNode_s Struct Reference . . . . .	197

<b>8 Mobile-C File Documentation</b>	<b>199</b>
8.1 /home/dko/projects/mobilec/trunk/src/acc.c File Reference . . . . .	199
8.2 /home/dko/projects/mobilec/trunk/src/agent.c File Reference . . . . .	202
8.3 /home/dko/projects/mobilec/trunk/src/agent_datastate.c File Reference . . . . .	205
8.4 /home/dko/projects/mobilec/trunk/src/agent_mailbox.c File Reference . . . . .	206
8.5 /home/dko/projects/mobilec/trunk/src/agent_return_data.c File Reference . . . . .	208
8.6 /home/dko/projects/mobilec/trunk/src/agent_task.c File Reference . . . . .	210
8.7 /home/dko/projects/mobilec/trunk/src/ams.c File Reference . . . . .	211
8.8 /home/dko/projects/mobilec/trunk/src/barrier.c File Reference . . . . .	213
8.9 /home/dko/projects/mobilec/trunk/src/cmd_prompt.c File Reference . . . . .	215
8.10 /home/dko/projects/mobilec/trunk/src/connection.c File Reference . . . . .	218
8.11 /home/dko/projects/mobilec/trunk/src/data_structures.c File Reference . . . . .	219
8.12 /home/dko/projects/mobilec/trunk/src/df.c File Reference . . . . .	221
8.13 /home/dko/projects/mobilec/trunk/src/dynstring.c File Reference . . . . .	225
8.14 /home/dko/projects/mobilec/trunk/src/fipa_acl.c File Reference . . . . .	226
8.15 /home/dko/projects/mobilec/trunk/src/fipa_envelope.c File Reference . . . . .	238
8.16 /home/dko/projects/mobilec/trunk/src/include/acc.h File Reference . . . . .	246
8.17 /home/dko/projects/mobilec/trunk/src/include/agent.h File Reference . . . . .	249
8.18 /home/dko/projects/mobilec/trunk/src/include/agent_datastate.h File Reference . . . . .	252
8.19 /home/dko/projects/mobilec/trunk/src/include/agent_lib.h File Reference . . . . .	254
8.20 /home/dko/projects/mobilec/trunk/src/include/agent_mailbox.h File Reference . . . . .	263
8.21 /home/dko/projects/mobilec/trunk/src/include/agent_task.h File Reference . . . . .	265
8.22 /home/dko/projects/mobilec/trunk/src/include/ams.h File Reference . . . . .	267
8.23 /home/dko/projects/mobilec/trunk/src/include/ap_queue_template.h File Reference . . . . .	269
8.24 /home/dko/projects/mobilec/trunk/src/include/barrier.h File Reference . . . . .	272
8.25 /home/dko/projects/mobilec/trunk/src/include/cmd_prompt.h File Reference . . . . .	275
8.26 /home/dko/projects/mobilec/trunk/src/include/commands.h File Reference . . . . .	277
8.27 /home/dko/projects/mobilec/trunk/src/include/commands.x.h File Reference . . . . .	279
8.28 /home/dko/projects/mobilec/trunk/src/include/connection.h File Reference . . . . .	281
8.29 /home/dko/projects/mobilec/trunk/src/include/data_structures.h File Reference . . . . .	283
8.30 /home/dko/projects/mobilec/trunk/src/include/df.h File Reference . . . . .	286
8.31 /home/dko/projects/mobilec/trunk/src/include/df_request.x.h File Reference . . . . .	292
8.32 /home/dko/projects/mobilec/trunk/src/include/dynstring.h File Reference . . . . .	293
8.33 /home/dko/projects/mobilec/trunk/src/include/fipa_acl.h File Reference . . . . .	295
8.34 /home/dko/projects/mobilec/trunk/src/include/fipa_acl_envelope.h File Reference . . . . .	310
8.35 /home/dko/projects/mobilec/trunk/src/include/host_id.h File Reference . . . . .	318

8.36	/home/dko/projects/mobilec/trunk/src/include/interpreter_variable_data.h File Reference	319
8.37	/home/dko/projects/mobilec/trunk/src/include/libmc.h File Reference	321
8.38	/home/dko/projects/mobilec/trunk/src/include/macros.h File Reference	356
8.39	/home/dko/projects/mobilec/trunk/src/include/mc_error.h File Reference	367
8.40	/home/dko/projects/mobilec/trunk/src/include/mc_platform.h File Reference	369
8.41	/home/dko/projects/mobilec/trunk/src/include/mc_rwlock.h File Reference	371
8.42	/home/dko/projects/mobilec/trunk/src/include/message.h File Reference	373
8.43	/home/dko/projects/mobilec/trunk/src/include/mtp_http.h File Reference	376
8.44	/home/dko/projects/mobilec/trunk/src/include/xml_compose.h File Reference	382
8.45	/home/dko/projects/mobilec/trunk/src/include/xml_helper.h File Reference	385
8.46	/home/dko/projects/mobilec/trunk/src/include/xml_parser.h File Reference	387
8.47	/home/dko/projects/mobilec/trunk/src/libmc.c File Reference	391
8.48	/home/dko/projects/mobilec/trunk/src/mc_list/list.c File Reference	431
8.49	/home/dko/projects/mobilec/trunk/src/mc_list/list.h File Reference	433
8.50	/home/dko/projects/mobilec/trunk/src/mc_platform.c File Reference	436
8.51	/home/dko/projects/mobilec/trunk/src/mc_rwlock.c File Reference	438
8.52	/home/dko/projects/mobilec/trunk/src/mc_sync/sync_list.c File Reference	440
8.53	/home/dko/projects/mobilec/trunk/src/mc_sync/sync_list.h File Reference	442
8.54	/home/dko/projects/mobilec/trunk/src/message.c File Reference	445
8.55	/home/dko/projects/mobilec/trunk/src/message_queue.c File Reference	448
8.56	/home/dko/projects/mobilec/trunk/src/mtp_http.c File Reference	449
8.57	/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/config.h File Reference	453
8.58	/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/vcnet/config.h File Reference	454
8.59	/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/vcnet2005/config.h File Reference	455
8.60	/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-attr.c File Reference	457
8.61	/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-entity.c File Reference	458
8.62	/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-file.c File Reference	460
8.63	/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-index.c File Reference	467
8.64	/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-node.c File Reference	469
8.65	/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-private.c File Reference	472
8.66	/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-search.c File Reference	474
8.67	/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-set.c File Reference	475
8.68	/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-string.c File Reference	477
8.69	/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml.h File Reference	478
8.70	/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxmldoc.c File Reference	490
8.71	/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/test/class.cxx File Reference	494

8.72	/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/test/enum.cxx File Reference . . . . .	495
8.73	/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/test/function.cxx File Reference . . . . .	496
8.74	/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/test/struct.cxx File Reference . . . . .	497
8.75	/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/testmxml.c File Reference . . . . .	498
8.76	/home/dko/projects/mobilec/trunk/src/security/asm.c File Reference . . . . .	500
8.77	/home/dko/projects/mobilec/trunk/src/security/asm.h File Reference . . . . .	501
8.78	/home/dko/projects/mobilec/trunk/src/security/asm_message_composer.c File Reference .	502
8.79	/home/dko/projects/mobilec/trunk/src/security/asm_message_composer.h File Reference .	503
8.80	/home/dko/projects/mobilec/trunk/src/security/asm_message_parser.c File Reference . . .	504
8.81	/home/dko/projects/mobilec/trunk/src/security/asm_message_parser.h File Reference . . .	505
8.82	/home/dko/projects/mobilec/trunk/src/security/asm_node.c File Reference . . . . .	506
8.83	/home/dko/projects/mobilec/trunk/src/security/asm_node.h File Reference . . . . .	507
8.84	/home/dko/projects/mobilec/trunk/src/security/mc_dh.c File Reference . . . . .	508
8.85	/home/dko/projects/mobilec/trunk/src/security/mc_dh.h File Reference . . . . .	509
8.86	/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/aes.h File Reference	510
8.87	/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/arc4.h File Reference	513
8.88	/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/base64.h File Reference . . . . .	515
8.89	/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/bignum.h File Reference . . . . .	517
8.90	/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/bn_asm.h File Reference . . . . .	533
8.91	/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/certs.h File Reference	535
8.92	/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/des.h File Reference	537
8.93	/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/dhm.h File Reference	542
8.94	/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/havege.h File Reference . . . . .	547
8.95	/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/md2.h File Reference	549
8.96	/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/md4.h File Reference	553
8.97	/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/md5.h File Reference	557
8.98	/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/net.h File Reference	562
8.99	/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/rsa.h File Reference	567
8.100	/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/sha1.h File Reference	579
8.101	/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/sha2.h File Reference	584
8.102	/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/sha4.h File Reference	589
8.103	/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/ssl.h File Reference	594
8.104	/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/ssl_conf.h File Reference . . . . .	612



8.105/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/timing.h File Reference . . . . .	613
8.106/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/x509.h File Reference . . . . .	615
8.107/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/aes.c File Reference . . . . .	628
8.108/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/arc4.c File Reference . . . . .	635
8.109/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/base64.c File Reference . . . . .	637
8.110/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/bignum.c File Reference . . . . .	640
8.111/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/certs.c File Reference . . . . .	655
8.112/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/des.c File Reference . . . . .	659
8.113/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/dhm.c File Reference . . . . .	671
8.114/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/dsa.c File Reference . . . . .	675
8.115/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/havege.c File Reference . . . . .	680
8.116/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/md2.c File Reference . . . . .	683
8.117/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/md4.c File Reference . . . . .	688
8.118/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/md5.c File Reference . . . . .	694
8.119/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/net.c File Reference . . . . .	700
8.120/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/rsa.c File Reference . . . . .	704
8.121/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/sha1.c File Reference . . . . .	711
8.122/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/sha2.c File Reference . . . . .	718
8.123/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/sha4.c File Reference . . . . .	725
8.124/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/ssl_cli.c File Reference . . . . .	732
8.125/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/ssl_srv.c File Reference . . . . .	735
8.126/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/ssl_tls.c File Reference . . . . .	739
8.127/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/timing.c File Reference . . . . .	749
8.128/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/x509read.c File Reference . . . . .	752
8.129/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/aes/aescript2.c File Reference . . . . .	761
8.130/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/hash/hello.c File Reference . . . . .	763
8.131/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/hash/md5sum.c File Reference . . . . .	764
8.132/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/hash/sha1sum.c File Reference . . . . .	766
8.133/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/hash/sha2sum.c File Reference . . . . .	768
8.134/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/dh_client.c File Reference . . . . .	770
8.135/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/dh_genprime.c File Reference . . . . .	772

8.136/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/dh_server.c	File	
Reference		773
8.137/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/mpi_demo.c	File	
Reference		775
8.138/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/rsa_genkey.c	File	
Reference		776
8.139/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/rsa_sign.c	File	
Reference		777
8.140/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/rsa_verify.c	File	
Reference		778
8.141/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/ssl/ssl_client1.c	File	
Reference		779
8.142/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/ssl/ssl_client2.c	File	
Reference		781
8.143/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/ssl/ssl_server.c	File	
Reference		783
8.144/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/test/benchmark.c	File	
Reference		786
8.145/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/test/selftest.c	File	
Reference		788
8.146/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAclMessage.cs	File	
Reference		789
8.147/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAgency.cs	File	
Reference		790
8.148/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAgent.cs	File	
Reference		791
8.149/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCEExports.cs	File	
Reference		792
8.150/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/Properties/AssemblyInfo.cs	File	
Reference		793
8.151/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/Properties/Settings.Designer.cs	File	
Reference		794
8.152/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/Settings.cs	File	
Reference		795
8.153/home/dko/projects/mobilec/trunk/src/winconfig.h	File	
Reference		796
8.154/home/dko/projects/mobilec/trunk/src/xml_compose.c	File	
Reference		797
8.155/home/dko/projects/mobilec/trunk/src/xml_helper.c	File	
Reference		801
8.156/home/dko/projects/mobilec/trunk/src/xml_parser.c	File	
Reference		803
<b>9 Mobile-C Example Documentation</b>		<b>807</b>
9.1 LibMCCConsole/Program.cs		807
9.2 LibMCCppEx/LibMCCppEx.cpp		808
9.3 LibMCFipaTest/Program.cs		809
9.4 LibMCGui/Form1.cs		810

---

9.5	LibMCMiscTest/Program.cs . . . . .	811
9.6	LibMCVbEx/Form1.vb . . . . .	812
<b>10</b>	<b>Mobile-C Page Documentation</b>	<b>813</b>
10.1	MCACLMessage . . . . .	813
10.2	Installing LibMC.NET . . . . .	814
10.3	Getting Started . . . . .	816
10.4	Using LibMC.NET . . . . .	817
10.5	Common Operations . . . . .	818
10.6	MCAgency . . . . .	819
10.7	MCAgent . . . . .	821
10.8	Todo List . . . . .	822
10.9	Bug List . . . . .	823



# Chapter 1

# MobileC

## 1.1 Introduction

Welcome to the MobileC documentation. For a quick user-api reference, please refer to the file [libmc.h](#).

This documentation is provided supplementary to the main Mobile-C User Guide. The official user guide may be obtained at <http://www.mobilec.org> , and should also be included with your Mobile-C Distribution package in the file docs/mobilec.pdf .



## Chapter 2

# Mobile-C Namespace Index

### 2.1 Mobile-C Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">LibMC</a> (Namespace for the .NET wrapper for Mobile-C ) . . . . .	13
<a href="#">LibMC::Properties</a> (Namespace for the .NET wrapper properties class ) . . . . .	14
<a href="#">System</a> . . . . .	15
<a href="#">System::Collections::Generic</a> . . . . .	16
<a href="#">System::Reflection</a> . . . . .	17
<a href="#">System::Runtime::CompilerServices</a> . . . . .	18
<a href="#">System::Runtime::InteropServices</a> . . . . .	19
<a href="#">System::Text</a> . . . . .	20





## Chapter 3

# Mobile-C Data Structure Index

### 3.1 Mobile-C Data Structures

Here are the data structures with brief descriptions:

<a href="#">_hr_time</a>	21
<a href="#">_x509_buf</a>	22
<a href="#">_x509_cert</a>	23
<a href="#">_x509_name</a>	27
<a href="#">_x509_time</a>	28
<a href="#">aes_context</a> (AES context structure )	30
<a href="#">agency_s</a> (The agency handle )	31
<a href="#">agent_datastate_s</a>	34
<a href="#">agent_mailbox_s</a>	37
<a href="#">agent_s</a>	38
<a href="#">agent_task_s</a>	43
<a href="#">arc4_context</a> (ARC4 context structure )	46
<a href="#">barrier_node_s</a>	47
<a href="#">barrier_queue_s</a>	48
<a href="#">cmd_prompt_s</a>	49
<a href="#">command_s</a>	50
<a href="#">connection_s</a>	51
<a href="#">des3_context</a> (Triple-DES context structure )	53
<a href="#">des_context</a> (DES context structure )	54
<a href="#">dhm_context</a>	55
<a href="#">dynstring_s</a>	57
<a href="#">fipa_acl_envelope_Received_s</a>	58
<a href="#">fipa_acl_envelope_s</a>	60
<a href="#">fipa_acl_message_s</a>	61
<a href="#">fipa_acl_Param_s</a>	64
<a href="#">fipa_agent_identifier_s</a>	66
<a href="#">fipa_agent_identifier_set_s</a>	67
<a href="#">fipa_DateTime_s</a>	68
<a href="#">fipa_expression_s</a>	70
<a href="#">fipa_expression_s::content_u</a>	71
<a href="#">fipa_message_string_s</a>	72
<a href="#">fipa_number_s</a>	73
<a href="#">fipa_string_s</a>	74

<a href="#">fipa_url_s</a>	75
<a href="#">fipa_url_sequence_s</a>	76
<a href="#">fipa_word_s</a>	77
<a href="#">foo_c</a>	78
<a href="#">foo_s</a>	81
<a href="#">havege_state</a> (HAVEGE state structure)	83
<a href="#">host_id_s</a>	85
<a href="#">hr_time</a> (Timer structure)	86
<a href="#">interpreter_variable_data_s</a>	87
<a href="#">list_s</a>	89
<a href="#">listNode_s</a>	90
<a href="#">mc_platform_s</a>	91
<a href="#">mc_rwlock_s</a>	96
<a href="#">LibMC::MCAclMessage</a> (Encapsulates ACL messages in the Mobile-C library)	98
<a href="#">LibMC::MCAgency</a> (Wrapper class for <a href="#">MCAgency_t</a> structure)	104
<a href="#">LibMC::MCAgency::ChOptions_t</a> (ChOptions structures)	132
<a href="#">LibMC::MCAgency::MCAgency_t</a>	133
<a href="#">LibMC::MCAgency::MCAgencyOptions_t</a>	135
<a href="#">MCAgencyOptions_s</a> (User modifiable agency options)	136
<a href="#">LibMC::MCAgent</a> (Wrapper class for <a href="#">MCAgent_t</a> structure)	138
<a href="#">md2_context</a> (MD2 context structure)	148
<a href="#">md4_context</a> (MD4 context structure)	150
<a href="#">md5_context</a> (MD5 context structure)	152
<a href="#">message_s</a>	154
<a href="#">mpi</a> (MPI structure)	157
<a href="#">mtp_http_content_s</a>	159
<a href="#">mtp_http_s</a>	160
<a href="#">mxml_attr_s</a>	163
<a href="#">mxml_custom_s</a>	164
<a href="#">mxml_fdbuf_s</a>	165
<a href="#">mxml_index_s</a>	166
<a href="#">mxml_node_s</a>	168
<a href="#">mxml_text_s</a>	170
<a href="#">mxml_value_s</a>	171
<a href="#">mxml_value_u</a>	172
<a href="#">rsa_context</a>	174
<a href="#">LibMC::Properties::Settings</a>	177
<a href="#">sha1_context</a> (SHA-1 context structure)	179
<a href="#">sha2_context</a> (SHA-256 context structure)	181
<a href="#">sha4_context</a> (SHA-512 context structure)	183
<a href="#">ssl_context</a>	185
<a href="#">syncList_s</a>	196
<a href="#">syncListNode_s</a>	197

## Chapter 4

# Mobile-C File Index

### 4.1 Mobile-C File List

Here is a list of all files with brief descriptions:

/home/dko/projects/mobilec/trunk/src/acc.c	199
/home/dko/projects/mobilec/trunk/src/agent.c	202
/home/dko/projects/mobilec/trunk/src/agent_datastate.c	205
/home/dko/projects/mobilec/trunk/src/agent_mailbox.c	206
/home/dko/projects/mobilec/trunk/src/agent_return_data.c	208
/home/dko/projects/mobilec/trunk/src/agent_task.c	210
/home/dko/projects/mobilec/trunk/src/ams.c	211
/home/dko/projects/mobilec/trunk/src/barrier.c	213
/home/dko/projects/mobilec/trunk/src/cmd_prompt.c	215
/home/dko/projects/mobilec/trunk/src/connection.c	218
/home/dko/projects/mobilec/trunk/src/data_structures.c	219
/home/dko/projects/mobilec/trunk/src/df.c	221
/home/dko/projects/mobilec/trunk/src/dynstring.c	225
/home/dko/projects/mobilec/trunk/src/fipa_acl.c	226
/home/dko/projects/mobilec/trunk/src/fipa_envelope.c	238
/home/dko/projects/mobilec/trunk/src/libmc.c	391
/home/dko/projects/mobilec/trunk/src/mc_platform.c	436
/home/dko/projects/mobilec/trunk/src/mc_rwlock.c	438
/home/dko/projects/mobilec/trunk/src/message.c	445
/home/dko/projects/mobilec/trunk/src/message_queue.c	448
/home/dko/projects/mobilec/trunk/src/mtp_http.c	449
/home/dko/projects/mobilec/trunk/src/winconfig.h	796
/home/dko/projects/mobilec/trunk/src/xml_compose.c	797
/home/dko/projects/mobilec/trunk/src/xml_helper.c	801
/home/dko/projects/mobilec/trunk/src/xml_parser.c	803
/home/dko/projects/mobilec/trunk/src/include/acc.h	246
/home/dko/projects/mobilec/trunk/src/include/agent.h	249
/home/dko/projects/mobilec/trunk/src/include/agent_datastate.h	252
/home/dko/projects/mobilec/trunk/src/include/agent_lib.h	254
/home/dko/projects/mobilec/trunk/src/include/agent_mailbox.h	263
/home/dko/projects/mobilec/trunk/src/include/agent_task.h	265
/home/dko/projects/mobilec/trunk/src/include/ams.h	267
/home/dko/projects/mobilec/trunk/src/include/ap_queue_template.h	269

/home/dko/projects/mobilec/trunk/src/include/barrier.h	272
/home/dko/projects/mobilec/trunk/src/include/cmd_prompt.h	275
/home/dko/projects/mobilec/trunk/src/include/commands.h	277
/home/dko/projects/mobilec/trunk/src/include/commands.x.h	279
/home/dko/projects/mobilec/trunk/src/include/connection.h	281
/home/dko/projects/mobilec/trunk/src/include/data_structures.h	283
/home/dko/projects/mobilec/trunk/src/include/df.h	286
/home/dko/projects/mobilec/trunk/src/include/df_request.x.h	292
/home/dko/projects/mobilec/trunk/src/include/dynstring.h	293
/home/dko/projects/mobilec/trunk/src/include/fipa_acl.h	295
/home/dko/projects/mobilec/trunk/src/include/fipa_acl_envelope.h	310
/home/dko/projects/mobilec/trunk/src/include/host_id.h	318
/home/dko/projects/mobilec/trunk/src/include/interpreter_variable_data.h	319
/home/dko/projects/mobilec/trunk/src/include/libmc.h (MobileC api header file)	321
/home/dko/projects/mobilec/trunk/src/include/macros.h	356
/home/dko/projects/mobilec/trunk/src/include/mc_error.h	367
/home/dko/projects/mobilec/trunk/src/include/mc_platform.h	369
/home/dko/projects/mobilec/trunk/src/include/mc_rwlock.h	371
/home/dko/projects/mobilec/trunk/src/include/message.h	373
/home/dko/projects/mobilec/trunk/src/include/mtp_http.h	376
/home/dko/projects/mobilec/trunk/src/include/xml_compose.h	382
/home/dko/projects/mobilec/trunk/src/include/xml_helper.h	385
/home/dko/projects/mobilec/trunk/src/include/xml_parser.h	387
/home/dko/projects/mobilec/trunk/src/mc_list/list.c	431
/home/dko/projects/mobilec/trunk/src/mc_list/list.h	433
/home/dko/projects/mobilec/trunk/src/mc_sync/sync_list.c	440
/home/dko/projects/mobilec/trunk/src/mc_sync/sync_list.h	442
/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/config.h	453
/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-attr.c	457
/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-entity.c	458
/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-file.c	460
/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-index.c	467
/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-node.c	469
/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-private.c	472
/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-search.c	474
/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-set.c	475
/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-string.c	477
/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml.h	478
/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxmldoc.c	490
/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/testmxml.c	498
/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/test/class.cxx	494
/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/test/enum.cxx	495
/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/test/function.cxx	496
/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/test/struct.cxx	497
/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/vcnet/config.h	454
/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/vcnet2005/config.h	455
/home/dko/projects/mobilec/trunk/src/security/asm.c	500
/home/dko/projects/mobilec/trunk/src/security/asm.h	501
/home/dko/projects/mobilec/trunk/src/security/asm_message_composer.c	502
/home/dko/projects/mobilec/trunk/src/security/asm_message_composer.h	503
/home/dko/projects/mobilec/trunk/src/security/asm_message_parser.c	504
/home/dko/projects/mobilec/trunk/src/security/asm_message_parser.h	505
/home/dko/projects/mobilec/trunk/src/security/asm_node.c	506
/home/dko/projects/mobilec/trunk/src/security/asm_node.h	507

/home/dko/projects/mobilec/trunk/src/security/mc_dh.c . . . . .	508
/home/dko/projects/mobilec/trunk/src/security/mc_dh.h . . . . .	509
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/aes.h . . . . .	510
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/arc4.h . . . . .	513
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/base64.h . . . . .	515
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/bignum.h . . . . .	517
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/bn_asm.h . . . . .	533
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/certs.h . . . . .	535
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/des.h . . . . .	537
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/dhm.h . . . . .	542
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/havege.h . . . . .	547
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/md2.h . . . . .	549
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/md4.h . . . . .	553
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/md5.h . . . . .	557
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/net.h . . . . .	562
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/rsa.h . . . . .	567
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/sha1.h . . . . .	579
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/sha2.h . . . . .	584
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/sha4.h . . . . .	589
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/ssl.h . . . . .	594
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/ssl_conf.h . . . . .	612
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/timing.h . . . . .	613
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/x509.h . . . . .	615
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/aes.c . . . . .	628
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/arc4.c . . . . .	635
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/base64.c . . . . .	637
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/bignum.c . . . . .	640
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/certs.c . . . . .	655
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/des.c . . . . .	659
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/dhm.c . . . . .	671
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/dsa.c . . . . .	675
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/havege.c . . . . .	680
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/md2.c . . . . .	683
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/md4.c . . . . .	688
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/md5.c . . . . .	694
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/net.c . . . . .	700
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/rsa.c . . . . .	704
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/sha1.c . . . . .	711
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/sha2.c . . . . .	718
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/sha4.c . . . . .	725
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/ssl_cli.c . . . . .	732
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/ssl_srv.c . . . . .	735
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/ssl_tls.c . . . . .	739
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/timing.c . . . . .	749
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/x509read.c . . . . .	752
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/aes/aescrypt2.c . . . . .	761
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/hash/hello.c . . . . .	763
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/hash/md5sum.c . . . . .	764
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/hash/sha1sum.c . . . . .	766
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/hash/sha2sum.c . . . . .	768
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/dh_client.c . . . . .	770
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/dh_genprime.c . . . . .	772
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/dh_server.c . . . . .	773
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/mpi_demo.c . . . . .	775

/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/rsa_genkey.c . . . . .	776
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/rsa_sign.c . . . . .	777
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/rsa_verify.c . . . . .	778
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/ssl/ssl_client1.c . . . . .	779
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/ssl/ssl_client2.c . . . . .	781
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/ssl/ssl_server.c . . . . .	783
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/test/benchmark.c . . . . .	786
/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/test/selftest.c . . . . .	788
/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAclMessage.cs . . . . .	789
/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAgency.cs . . . . .	790
/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAgent.cs . . . . .	791
/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCExports.cs . . . . .	792
/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/Settings.cs . . . . .	795
/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/Properties/AssemblyInfo.cs . .	793
/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/Properties/Settings.Designer.cs	794

# Chapter 5

## Mobile-C Page Index

### 5.1 Mobile-C Related Pages

Here is a list of all related documentation pages:

Installing LibMC.NET . . . . .	<a href="#">814</a>
Getting Started . . . . .	<a href="#">816</a>
Using LibMC.NET . . . . .	<a href="#">817</a>
Common Operations . . . . .	<a href="#">818</a>
Todo List . . . . .	<a href="#">822</a>
Bug List . . . . .	<a href="#">823</a>





## Chapter 6

# Mobile-C Namespace Documentation

### 6.1 LibMC Namespace Reference

Namespace for the .NET wrapper for Mobile-C.

#### Data Structures

- class [MCAclMessage](#)  
*Encapsulates ACL messages in the Mobile-C library.*
- class [MCAgency](#)  
*Wrapper class for [MCAgency\\_t](#) structure.*
- class [MCAgent](#)  
*Wrapper class for [MCAgent\\_t](#) structure.*

#### Namespaces

- namespace [Properties](#)  
*Namespace for the .NET wrapper properties class.*

#### 6.1.1 Detailed Description

Namespace for the .NET wrapper for Mobile-C.

[LibMC](#) encapsulates the Mobile-C DLL for windows in an .NET class library. .NET programs can access the library to create agencies, connect to agencies, interact with agents, etc.

## 6.2 LibMC::Properties Namespace Reference

Namespace for the .NET wrapper properties class.

### Data Structures

- class [Settings](#)

### 6.2.1 Detailed Description

Namespace for the .NET wrapper properties class.

Any user or global properties that should preserved from session to session can be added here through the designer. There are currently no properties in use.

## 6.3 System Namespace Reference

## 6.4 System::Collections::Generic Namespace Reference

## 6.5 System::Reflection Namespace Reference

## 6.6 System::Runtime::CompilerServices Namespace Reference

## 6.7 System::Runtime::InteropServices Namespace Reference

## 6.8 System::Text Namespace Reference



## Chapter 7

# Mobile-C Data Structure Documentation

### 7.1 `_hr_time` Struct Reference

#### Data Fields

- struct timeval [start](#)

#### 7.1.1 Detailed Description

Definition at line 47 of file `timing.c`.

#### 7.1.2 Field Documentation

##### 7.1.2.1 `struct timeval _hr_time::start` [read]

Definition at line 49 of file `timing.c`.

Referenced by `set_timer()`.

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

- `/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/timing.c`

## 7.2 `_x509_buf` Struct Reference

```
#include <x509.h>
```

### Data Fields

- `int tag`
- `int len`
- `unsigned char * p`

### 7.2.1 Detailed Description

Definition at line 88 of file `x509.h`.

### 7.2.2 Field Documentation

#### 7.2.2.1 `int _x509_buf::tag`

Definition at line 90 of file `x509.h`.

Referenced by `x509_get_alg()`, `x509_get_ext()`, `x509_get_name()`, `x509_get_serial()`, `x509_get_sig()`, and `x509_get_uid()`.

#### 7.2.2.2 `int _x509_buf::len`

Definition at line 91 of file `x509.h`.

Referenced by `ssl_write_certificate()`, `ssl_write_certificate_request()`, `x509_add_certs()`, `x509_cert_info()`, `x509_dn_gets()`, `x509_get_alg()`, `x509_get_ext()`, `x509_get_name()`, `x509_get_pubkey()`, `x509_get_serial()`, `x509_get_sig()`, `x509_get_uid()`, and `x509_verify_cert()`.

#### 7.2.2.3 `unsigned char* _x509_buf::p`

Definition at line 92 of file `x509.h`.

Referenced by `ssl_write_certificate()`, `ssl_write_certificate_request()`, `x509_add_certs()`, `x509_cert_info()`, `x509_dn_gets()`, `x509_free_cert()`, `x509_get_alg()`, `x509_get_ext()`, `x509_get_name()`, `x509_get_pubkey()`, `x509_get_serial()`, `x509_get_sig()`, `x509_get_uid()`, and `x509_verify_cert()`.

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

- `/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/x509.h`

## 7.3 `_x509_cert` Struct Reference

```
#include <x509.h>
```

### Data Fields

- `x509_buf raw`
- `x509_buf tbs`
- `int version`
- `x509_buf serial`
- `x509_buf sig_oid1`
- `x509_buf issuer_raw`
- `x509_buf subject_raw`
- `x509_name issuer`
- `x509_name subject`
- `x509_time valid_from`
- `x509_time valid_to`
- `x509_buf pk_oid`
- `rsa_context rsa`
- `x509_buf issuer_id`
- `x509_buf subject_id`
- `x509_buf v3_ext`
- `int ca_istrue`
- `int max_pathlen`
- `x509_buf sig_oid2`
- `x509_buf sig`
- `struct _x509_cert * next`

### 7.3.1 Detailed Description

Definition at line 111 of file `x509.h`.

### 7.3.2 Field Documentation

#### 7.3.2.1 `x509_buf _x509_cert::raw`

Definition at line 113 of file `x509.h`.

Referenced by `ssl_write_certificate()`, `x509_add_certs()`, and `x509_free_cert()`.

#### 7.3.2.2 `x509_buf _x509_cert::tbs`

Definition at line 114 of file `x509.h`.

Referenced by `x509_add_certs()`, and `x509_verify_cert()`.

#### 7.3.2.3 `int _x509_cert::version`

Definition at line 116 of file `x509.h`.

Referenced by `x509_add_certs()`, `x509_cert_info()`, and `x509_verify_cert()`.

**7.3.2.4 x509\_buf\_x509\_cert::serial**

Definition at line 117 of file x509.h.

Referenced by x509\_add\_certs(), and x509\_cert\_info().

**7.3.2.5 x509\_buf\_x509\_cert::sig\_oid1**

Definition at line 118 of file x509.h.

Referenced by x509\_add\_certs(), x509\_cert\_info(), and x509\_verify\_cert().

**7.3.2.6 x509\_buf\_x509\_cert::issuer\_raw**

Definition at line 120 of file x509.h.

Referenced by x509\_add\_certs(), and x509\_verify\_cert().

**7.3.2.7 x509\_buf\_x509\_cert::subject\_raw**

Definition at line 121 of file x509.h.

Referenced by ssl\_write\_certificate\_request(), x509\_add\_certs(), and x509\_verify\_cert().

**7.3.2.8 x509\_name\_x509\_cert::issuer**

Definition at line 123 of file x509.h.

Referenced by x509\_add\_certs(), x509\_cert\_info(), and x509\_free\_cert().

**7.3.2.9 x509\_name\_x509\_cert::subject**

Definition at line 124 of file x509.h.

Referenced by x509\_add\_certs(), x509\_cert\_info(), x509\_free\_cert(), and x509\_verify\_cert().

**7.3.2.10 x509\_time\_x509\_cert::valid\_from**

Definition at line 126 of file x509.h.

Referenced by x509\_add\_certs(), and x509\_cert\_info().

**7.3.2.11 x509\_time\_x509\_cert::valid\_to**

Definition at line 127 of file x509.h.

Referenced by x509\_add\_certs(), x509\_cert\_info(), and x509\_is\_cert\_expired().

**7.3.2.12 x509\_buf\_x509\_cert::pk\_oid**

Definition at line 129 of file x509.h.

Referenced by x509\_add\_certs().

**7.3.2.13 rsa\_context \_x509\_cert::rsa**

Definition at line 130 of file x509.h.

Referenced by ssl\_parse\_certificate\_verify(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_client\_key\_exchange(), x509\_add\_certs(), x509\_cert\_info(), x509\_free\_cert(), and x509\_verify\_cert().

**7.3.2.14 x509\_buf \_x509\_cert::issuer\_id**

Definition at line 132 of file x509.h.

Referenced by x509\_add\_certs().

**7.3.2.15 x509\_buf \_x509\_cert::subject\_id**

Definition at line 133 of file x509.h.

Referenced by x509\_add\_certs().

**7.3.2.16 x509\_buf \_x509\_cert::v3\_ext**

Definition at line 134 of file x509.h.

Referenced by x509\_add\_certs().

**7.3.2.17 int \_x509\_cert::ca\_istrue**

Definition at line 136 of file x509.h.

Referenced by x509\_add\_certs(), and x509\_verify\_cert().

**7.3.2.18 int \_x509\_cert::max\_pathlen**

Definition at line 137 of file x509.h.

Referenced by x509\_add\_certs(), and x509\_verify\_cert().

**7.3.2.19 x509\_buf \_x509\_cert::sig\_oid2**

Definition at line 139 of file x509.h.

Referenced by x509\_add\_certs().

**7.3.2.20 x509\_buf \_x509\_cert::sig**

Definition at line 140 of file x509.h.

Referenced by x509\_add\_certs(), and x509\_verify\_cert().

**7.3.2.21 struct \_x509\_cert\* \_x509\_cert::next** [read]

Definition at line 142 of file x509.h.

Referenced by `main()`, `ssl_write_certificate()`, `ssl_write_certificate_request()`, `x509_add_certs()`, `x509_free_cert()`, and `x509_verify_cert()`.

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

- `/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/x509.h`

## 7.4 `_x509_name` Struct Reference

```
#include <x509.h>
```

### Data Fields

- [x509\\_buf oid](#)
- [x509\\_buf val](#)
- [struct `\_x509\_name` \\* next](#)

### 7.4.1 Detailed Description

Definition at line 96 of file `x509.h`.

### 7.4.2 Field Documentation

#### 7.4.2.1 `x509_buf _x509_name::oid`

Definition at line 98 of file `x509.h`.

Referenced by `x509_dn_gets()`, `x509_get_name()`, and `x509_verify_cert()`.

#### 7.4.2.2 `x509_buf _x509_name::val`

Definition at line 99 of file `x509.h`.

Referenced by `x509_dn_gets()`, `x509_get_name()`, and `x509_verify_cert()`.

#### 7.4.2.3 `struct _x509_name* _x509_name::next` [read]

Definition at line 100 of file `x509.h`.

Referenced by `x509_dn_gets()`, `x509_free_cert()`, `x509_get_name()`, and `x509_verify_cert()`.

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

- `/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/x509.h`

## 7.5 `_x509_time` Struct Reference

```
#include <x509.h>
```

### Data Fields

- `int year`
- `int mon`
- `int day`
- `int hour`
- `int min`
- `int sec`

#### 7.5.1 Detailed Description

Definition at line 104 of file `x509.h`.

#### 7.5.2 Field Documentation

##### 7.5.2.1 `int _x509_time::year`

Definition at line 106 of file `x509.h`.

Referenced by `x509_cert_info()`, `x509_get_dates()`, and `x509_is_cert_expired()`.

##### 7.5.2.2 `int _x509_time::mon`

Definition at line 106 of file `x509.h`.

Referenced by `x509_cert_info()`, `x509_get_dates()`, and `x509_is_cert_expired()`.

##### 7.5.2.3 `int _x509_time::day`

Definition at line 106 of file `x509.h`.

Referenced by `x509_cert_info()`, `x509_get_dates()`, and `x509_is_cert_expired()`.

##### 7.5.2.4 `int _x509_time::hour`

Definition at line 107 of file `x509.h`.

Referenced by `x509_cert_info()`, and `x509_get_dates()`.

##### 7.5.2.5 `int _x509_time::min`

Definition at line 107 of file `x509.h`.

Referenced by `x509_cert_info()`, and `x509_get_dates()`.



### 7.5.2.6 `int _x509_time::sec`

Definition at line 107 of file `x509.h`.

Referenced by `x509_cert_info()`, and `x509_get_dates()`.

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

- `/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/x509.h`

## 7.6 aes\_context Struct Reference

AES context structure.

```
#include <aes.h>
```

### Data Fields

- unsigned long [erk](#) [64]
- unsigned long [drk](#) [64]
- [int](#) [nr](#)

### 7.6.1 Detailed Description

AES context structure.

Definition at line 17 of file [aes.h](#).

### 7.6.2 Field Documentation

#### 7.6.2.1 unsigned long aes\_context::erk[64]

encryption round keys

Definition at line 19 of file [aes.h](#).

Referenced by [aes\\_set\\_key\(\)](#).

#### 7.6.2.2 unsigned long aes\_context::drk[64]

decryption round keys

Definition at line 20 of file [aes.h](#).

Referenced by [aes\\_set\\_key\(\)](#).

#### 7.6.2.3 int aes\_context::nr

number of rounds

Definition at line 21 of file [aes.h](#).

Referenced by [aes\\_set\\_key\(\)](#).

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

- [/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/aes.h](#)

## 7.7 agency\_s Struct Reference

The agency handle.

```
#include <libmc.h>
```

### Data Fields

- [int client](#)
- [int server](#)
- [char \\* hostName](#)
- [char \\* filename](#)
- [int portno](#)
- [int portnoc](#)
- [struct mc\\_platform\\_s \\* mc\\_platform](#)
- [int default\\_agentstatus](#)
- [int threads](#)
- [int enable\\_security](#)
- [int stack\\_size](#) [MC\_THREAD\_ALL]
- [error\\_code\\_t last\\_error](#)

### 7.7.1 Detailed Description

The agency handle.

Definition at line 215 of file libmc.h.

### 7.7.2 Field Documentation

#### 7.7.2.1 int agency\_s::client

Definition at line 216 of file libmc.h.

Referenced by MC\_Initialize().

#### 7.7.2.2 int agency\_s::server

Definition at line 217 of file libmc.h.

Referenced by MC\_Initialize().

#### 7.7.2.3 char\* agency\_s::hostName

Local Hostname

Definition at line 218 of file libmc.h.

Referenced by MC\_End(), and MC\_Initialize().

#### 7.7.2.4 char\* agency\_s::filename

Definition at line 219 of file libmc.h.

#### 7.7.2.5 int agency\_s::portno

Local port number

Definition at line 220 of file libmc.h.

Referenced by MC\_Initialize(), and mc\_platform\_Initialize().

#### 7.7.2.6 int agency\_s::portnoc

Definition at line 221 of file libmc.h.

#### 7.7.2.7 struct mc\_platform\_s\* agency\_s::mc\_platform [read]

Local MobileC Platform

Definition at line 222 of file libmc.h.

Referenced by MC\_AclSend\_chdl(), MC\_AddAgent(), MC\_AddAgent\_chdl(), MC\_Barrier(), MC\_Barrier\_chdl(), MC\_BarrierDelete(), MC\_BarrierDelete\_chdl(), MC\_BarrierInit(), MC\_BarrierInit\_chdl(), MC\_ChInitializeOptions(), MC\_CondBroadcast(), MC\_CondBroadcast\_chdl(), MC\_CondReset(), MC\_CondReset\_chdl(), MC\_CondSignal(), MC\_CondSignal\_chdl(), MC\_CondWait(), MC\_CondWait\_chdl(), MC\_DeregisterService(), MC\_DeregisterService\_chdl(), MC\_End(), MC\_End\_chdl(), MC\_FindAgentByID(), MC\_FindAgentByID\_chdl(), MC\_FindAgentByName(), MC\_FindAgentByName\_chdl(), MC\_GetAllAgents(), MC\_HaltAgency(), MC\_HaltAgency\_chdl(), MC\_Initialize(), MC\_LoadAgentFromFile(), MC\_MainLoop(), MC\_MutexLock(), MC\_MutexUnlock(), MC\_RegisterService(), MC\_ResetSignal(), MC\_ResumeAgency(), MC\_RetrieveAgent(), MC\_SearchForService(), MC\_SemaphorePost(), MC\_SemaphoreWait(), MC\_SendAgentMigrationMessage(), MC\_SendAgentMigrationMessageFile(), MC\_SendSteerCommand(), MC\_SetDefaultAgentStatus(), MC\_Steer(), MC\_SyncDelete(), MC\_SyncInit(), MC\_WaitAgent(), MC\_WaitRetrieveAgent(), and MC\_WaitSignal().

#### 7.7.2.8 int agency\_s::default\_agentstatus

Agency default agent status

Definition at line 223 of file libmc.h.

Referenced by MC\_Initialize(), and mc\_platform\_Initialize().

#### 7.7.2.9 int agency\_s::threads

flag which determines which threads to start

Definition at line 224 of file libmc.h.

Referenced by MC\_End(), MC\_Initialize(), and mc\_platform\_Initialize().

#### 7.7.2.10 int agency\_s::enable\_security

Security flag

Definition at line 225 of file libmc.h.

Referenced by MC\_Initialize(), and mc\_platform\_Initialize().

**7.7.2.11 int agency\_s::stack\_size[MC\_THREAD\_ALL]**

Definition at line 226 of file libmc.h.

Referenced by MC\_Initialize(), and mc\_platform\_Initialize().

**7.7.2.12 error\_code\_t agency\_s::last\_error**

Definition at line 227 of file libmc.h.

Referenced by mc\_platform\_Initialize().

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

- /home/dko/projects/mobilec/trunk/src/include/[libmc.h](#)

## 7.8 agent\_datastate\_s Struct Reference

```
#include <agent_datastate.h>
```

### Data Fields

- char \*\* [agent\\_code\\_ids](#)
- char \*\* [agent\\_codes](#)
- char \* [agent\\_code](#)
- [agent\\_task\\_p](#) \* [tasks](#)
- [mxml\\_node\\_t](#) \* [xml\\_agent\\_root](#)
- [mxml\\_node\\_t](#) \* [xml\\_root](#)
- int [task\\_progress](#)
- int [return\\_data](#)
- int [number\\_of\\_tasks](#)
- int [persistent](#)
- int [init\\_agent\\_status](#)

### 7.8.1 Detailed Description

Definition at line 41 of file `agent_datastate.h`.

### 7.8.2 Field Documentation

#### 7.8.2.1 char\*\* agent\_datastate\_s::agent\_code\_ids

Definition at line 43 of file `agent_datastate.h`.

Referenced by `agent_datastate_Copy()`, `agent_datastate_Destroy()`, `agent_xml_compose__agent_code()`, `agent_xml_parse__agent_code()`, `agent_xml_parse__tasks()`, and `MC_ComposeAgent()`.

#### 7.8.2.2 char\*\* agent\_datastate\_s::agent\_codes

Definition at line 44 of file `agent_datastate.h`.

Referenced by `agent_datastate_Copy()`, `agent_datastate_Destroy()`, `agent_xml_compose__agent_code()`, `agent_xml_parse__agent_code()`, `agent_xml_parse__tasks()`, and `MC_ComposeAgent()`.

#### 7.8.2.3 char\* agent\_datastate\_s::agent\_code

Definition at line 45 of file `agent_datastate.h`.

Referenced by `agent_datastate_Copy()`, `agent_datastate_New()`, `agent_xml_parse__agent_code()`, `agent_xml_parse__tasks()`, `MC_ComposeAgent()`, `MC_PrintAgentCode()`, and `MC_RetrieveAgentCode()`.

#### 7.8.2.4 agent\_task\_p\* agent\_datastate\_s::tasks

Definition at line 48 of file `agent_datastate.h`.

Referenced by agent\_AddPersistentVariable(), agent\_datastate\_Copy(), agent\_datastate\_Destroy(), agent\_datastate\_New(), agent\_xml\_compose\_\_task(), agent\_xml\_parse\_\_agent\_code(), agent\_xml\_parse\_\_data(), agent\_xml\_parse\_\_task(), agent\_xml\_parse\_\_tasks(), interpreter\_variable\_data\_InitializeFromAgent(), MC\_AgentVariableRetrieve(), MC\_AgentVariableSave(), MC\_ComposeAgent(), MC\_GetAgentReturnData(), and message\_InitializeFromAgent().

#### 7.8.2.5 mxml\_node\_t\* agent\_datastate\_s::xml\_agent\_root

Definition at line 51 of file agent\_datastate.h.

Referenced by agent\_datastate\_New(), agent\_Initialize(), agent\_xml\_parse(), and MC\_GetAgentXMLString().

#### 7.8.2.6 mxml\_node\_t\* agent\_datastate\_s::xml\_root

Definition at line 52 of file agent\_datastate.h.

Referenced by agent\_datastate\_Destroy(), agent\_datastate\_New(), agent\_Initialize(), and agent\_return\_xml\_parse().

#### 7.8.2.7 int agent\_datastate\_s::task\_progress

Definition at line 55 of file agent\_datastate.h.

Referenced by agent\_AddPersistentVariable(), agent\_datastate\_Copy(), agent\_datastate\_New(), agent\_RunChScriptThread(), agent\_xml\_compose\_\_tasks(), agent\_xml\_parse\_\_agent\_code(), agent\_xml\_parse\_\_tasks(), interpreter\_variable\_data\_InitializeFromAgent(), MC\_AgentVariableRetrieve(), MC\_AgentVariableSave(), MC\_PrintAgentCode(), MC\_RetrieveAgentCode(), and message\_InitializeFromAgent().

#### 7.8.2.8 int agent\_datastate\_s::return\_data

Definition at line 56 of file agent\_datastate.h.

Referenced by agent\_datastate\_Copy(), and agent\_datastate\_New().

#### 7.8.2.9 int agent\_datastate\_s::number\_of\_tasks

Definition at line 59 of file agent\_datastate.h.

Referenced by agent\_datastate\_Copy(), agent\_datastate\_Destroy(), agent\_datastate\_New(), agent\_xml\_compose\_\_tasks(), agent\_xml\_parse\_\_agent\_code(), agent\_xml\_parse\_\_tasks(), MC\_ComposeAgent(), MC\_GetAgentNumTasks(), MC\_GetAgentReturnData(), MC\_PrintAgentCode(), and message\_InitializeFromAgent().

#### 7.8.2.10 int agent\_datastate\_s::persistent

Definition at line 62 of file agent\_datastate.h.

Referenced by acc\_MessageHandlerThread(), agent\_datastate\_Copy(), agent\_datastate\_New(), agent\_xml\_compose\_\_task(), and MC\_ComposeAgent().

#### 7.8.2.11 `int agent_datastate_s::init_agent_status`

Definition at line 63 of file `agent_datastate.h`.

Referenced by `agent_datastate_Copy()`, and `agent_datastate_New()`.

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

- `/home/dko/projects/mobilec/trunk/src/include/agent_datastate.h`



## 7.9 agent\_mailbox\_s Struct Reference

```
#include <agent_mailbox.h>
```

### Data Fields

- struct mail\_queue\_s \* [mail\\_queue](#)

### 7.9.1 Detailed Description

Definition at line 8 of file agent\_mailbox.h.

### 7.9.2 Field Documentation

#### 7.9.2.1 struct mail\_queue\_s\* agent\_mailbox\_s::mail\_queue [read]

Definition at line 10 of file agent\_mailbox.h.

Referenced by agent\_mailbox\_Copy(), agent\_mailbox\_Destroy(), agent\_mailbox\_New(), agent\_mailbox\_Post(), agent\_mailbox\_Retrieve(), and agent\_mailbox\_WaitRetrieve().

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

- /home/dko/projects/mobilec/trunk/src/include/[agent\\_mailbox.h](#)

## 7.10 agent\_s Struct Reference

```
#include <agent.h>
```

### Data Fields

- u\_long [id](#)
- char \* [name](#)
- u\_long [connect\\_id](#)
- time\_t [arrival\\_time](#)
- char \* [owner](#)
- char \* [home](#)
- int [home\\_port](#)
- int [orphan](#)
- [agent\\_datastate\\_p](#) [datastate](#)
- enum [MC\\_AgentType\\_e](#) [agent\\_type](#)
- enum [MC\\_AgentStatus\\_e](#) [agent\\_status](#)
- int [return\\_data](#)
- ChInterp\_t [agent\\_interp](#)
- MUTEX\_T \* [run\\_lock](#)
- int [agent\\_thread\\_id](#)
- THREAD\_T \* [agent\\_thread](#)
- [agent\\_mailbox\\_p](#) [mailbox](#)
- int [agent\\_pipe\\_active](#)
- int [agent\\_ready\\_to\\_send](#)
- int [agent\\_pipe\\_ready\\_to\\_read](#)
- int [agent\\_script\\_ready](#)
- int [agent\\_persistent](#)
- struct [mc\\_platform\\_s](#) \* [mc\\_platform](#)
- MUTEX\_T \* [lock](#)

### 7.10.1 Detailed Description

Definition at line 48 of file [agent.h](#).

### 7.10.2 Field Documentation

#### 7.10.2.1 u\_long agent\_s::id

Definition at line 51 of file [agent.h](#).

Referenced by [agent\\_Copy\(\)](#), [agent\\_Initialize\(\)](#), [agent\\_RunChScriptThread\(\)](#), [ams\\_Print\(\)](#), [AP\\_QUEUE\\_STD\\_DEFN\\_TEMPLATE\(\)](#), [MC\\_GetAgentID\(\)](#), and [MC\\_RegisterService\(\)](#).

### 7.10.2.2 char\* agent\_s::name

Definition at line 52 of file agent.h.

Referenced by acc\_MessageHandlerThread(), agent\_AddPersistentVariable(), agent\_Copy(), agent\_Destroy(), agent\_RunChScriptThread(), agent\_xml\_compose\_\_name(), agent\_xml\_parse\_\_name(), AP\_QUEUE\_STD\_DEFN\_TEMPLATE(), MC\_ComposeAgent(), MC\_GetAgentName(), and MC\_RegisterService().

### 7.10.2.3 u\_long agent\_s::connect\_id

Definition at line 53 of file agent.h.

Referenced by ams\_Print().

### 7.10.2.4 time\_t agent\_s::arrival\_time

Definition at line 55 of file agent.h.

Referenced by agent\_Copy(), agent\_Initialize(), and MC\_GetAgentArrivalTime().

### 7.10.2.5 char\* agent\_s::owner

Definition at line 59 of file agent.h.

Referenced by agent\_Copy(), agent\_Destroy(), agent\_xml\_compose\_\_owner(), agent\_xml\_parse\_\_owner(), and MC\_ComposeAgent().

### 7.10.2.6 char\* agent\_s::home

Definition at line 60 of file agent.h.

Referenced by agent\_Copy(), agent\_Destroy(), agent\_xml\_compose\_\_home(), agent\_xml\_parse\_\_home(), MC\_ComposeAgent(), and message\_InitializeFromAgent().

### 7.10.2.7 int agent\_s::home\_port

Definition at line 61 of file agent.h.

Referenced by agent\_Copy().

### 7.10.2.8 int agent\_s::orphan

Definition at line 63 of file agent.h.

Referenced by agent\_Copy(), agent\_Initialize(), ams\_ManageAgentList(), MC\_ComposeAgent(), and MC\_SetAgentStatus().

### 7.10.2.9 agent\_datastate\_p agent\_s::datastate

Definition at line 67 of file agent.h.

Referenced by `acc_MessageHandlerThread()`, `agent_AddPersistentVariable()`, `agent_Copy()`, `agent_Destroy()`, `agent_Initialize()`, `agent_return_xml_parse()`, `agent_RunChScriptThread()`, `agent_xml_compose__agent_code()`, `agent_xml_compose__task()`, `agent_xml_compose__tasks()`, `agent_xml_parse()`, `agent_xml_parse__agent_code()`, `agent_xml_parse__data()`, `agent_xml_parse__task()`, `agent_xml_parse__tasks()`, `interpreter_variable_data_InitializeFromAgent()`, `MC_AgentVariableRetrieve()`, `MC_AgentVariableSave()`, `MC_ComposeAgent()`, `MC_GetAgentNumTasks()`, `MC_GetAgentReturnData()`, `MC_GetAgentXMLString()`, `MC_PrintAgentCode()`, `MC_RetrieveAgentCode()`, and `message_InitializeFromAgent()`.

#### 7.10.2.10 `enum MC_AgentType_e agent_s::agent_type`

Definition at line 68 of file `agent.h`.

Referenced by `agent_Copy()`, `agent_Initialize()`, `agent_xml_compose__message()`, `MC_ComposeAgent()`, and `MC_GetAgentType()`.

#### 7.10.2.11 `enum MC_AgentStatus_e agent_s::agent_status`

Definition at line 69 of file `agent.h`.

Referenced by `acc_MessageHandlerThread()`, `agent_Copy()`, `agent_Destroy()`, `agent_Initialize()`, `agent_RunChScript()`, `ams_ManageAgentList()`, `ams_Print()`, `MC_ComposeAgent()`, `MC_GetAgentStatus()`, `MC_RetrieveAgent()`, and `MC_SetAgentStatus()`.

#### 7.10.2.12 `int agent_s::return_data`

Definition at line 71 of file `agent.h`.

Referenced by `agent_Copy()`.

#### 7.10.2.13 `ChInterp_t agent_s::agent_interp`

Definition at line 74 of file `agent.h`.

Referenced by `agent_AddPersistentVariable()`, `agent_Copy()`, `agent_Destroy()`, `agent_RunChScriptThread()`, `interpreter_variable_data_Initialize()`, `interpreter_variable_data_InitializeFromAgent()`, `MC_CallAgentFunc()`, `MC_GetAgentExecEngine()`, and `MC_TerminateAgent()`.

#### 7.10.2.14 `MUTEX_T* agent_s::run_lock`

Definition at line 77 of file `agent.h`.

Referenced by `agent_Copy()`, `agent_Destroy()`, `agent_Initialize()`, `agent_New()`, `interpreter_variable_data_Initialize()`, and `MC_CallAgentFunc()`.

#### 7.10.2.15 `int agent_s::agent_thread_id`

Definition at line 80 of file `agent.h`.

Referenced by `agent_Initialize()`.

**7.10.2.16** `THREAD_T* agent_s::agent_thread`

Definition at line 83 of file agent.h.

Referenced by agent\_Copy(), agent\_Destroy(), agent\_Initialize(), and agent\_RunChScript().

**7.10.2.17** `agent_mailbox_p agent_s::mailbox`

Definition at line 86 of file agent.h.

Referenced by acc\_Thread(), agent\_Destroy(), agent\_Initialize(), MC\_AclPost(), MC\_AclRetrieve(), and MC\_AclWaitRetrieve().

**7.10.2.18** `int agent_s::agent_pipe_active`

Definition at line 89 of file agent.h.

Referenced by agent\_Initialize().

**7.10.2.19** `int agent_s::agent_ready_to_send`

Definition at line 90 of file agent.h.

Referenced by agent\_Initialize().

**7.10.2.20** `int agent_s::agent_pipe_ready_to_read`

Definition at line 91 of file agent.h.

Referenced by agent\_Initialize().

**7.10.2.21** `int agent_s::agent_script_ready`

Definition at line 92 of file agent.h.

Referenced by agent\_Initialize().

**7.10.2.22** `int agent_s::agent_persistent`

Definition at line 93 of file agent.h.

Referenced by agent\_Copy().

**7.10.2.23** `struct mc_platform_s* agent_s::mc_platform` [read]

Definition at line 95 of file agent.h.

Referenced by agent\_Initialize(), agent\_RunChScript(), agent\_RunChScriptThread(), MC\_AddAgent(), and MC\_SetAgentStatus().

**7.10.2.24 MUTEX\_T\* agent\_s::lock**

Definition at line 97 of file agent.h.

Referenced by acc\_MessageHandlerThread(), agent\_Copy(), agent\_Destroy(), agent\_Initialize(), agent\_New(), ams\_ManageAgentList(), MC\_GetAgentName(), MC\_GetAgentStatus(), MC\_PrintAgentCode(), MC\_RetrieveAgentCode(), and MC\_SetAgentStatus().

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

- [/home/dko/projects/mobilec/trunk/src/include/agent.h](#)

## 7.11 agent\_task\_s Struct Reference

```
#include <agent_task.h>
```

### Data Fields

- [int task\\_completed](#)
- [int number\\_of\\_elements](#)
- [int size\\_of\\_element\\_array](#)
- [int persistent](#)
- [int init\\_agent\\_status](#)
- [char \\* var\\_name](#)
- [char \\* server\\_name](#)
- [char \\* code\\_id](#)
- [interpreter\\_variable\\_data\\_t \\* agent\\_return\\_data](#)
- [struct agent\\_variable\\_list\\_s \\* agent\\_variable\\_list](#)
- [char \\*\\* saved\\_variables](#)
- [int num\\_saved\\_variables](#)

### 7.11.1 Detailed Description

Definition at line 40 of file agent\_task.h.

### 7.11.2 Field Documentation

#### 7.11.2.1 int agent\_task\_s::task\_completed

Definition at line 43 of file agent\_task.h.

Referenced by agent\_task\_Copy(), agent\_xml\_compose\_\_task(), and agent\_xml\_parse\_\_task().

#### 7.11.2.2 int agent\_task\_s::number\_of\_elements

Definition at line 44 of file agent\_task.h.

Referenced by agent\_task\_Copy().

#### 7.11.2.3 int agent\_task\_s::size\_of\_element\_array

Definition at line 45 of file agent\_task.h.

Referenced by agent\_task\_Copy().

#### 7.11.2.4 int agent\_task\_s::persistent

Definition at line 46 of file agent\_task.h.

Referenced by agent\_task\_Copy(), agent\_xml\_compose\_\_task(), and agent\_xml\_parse\_\_data().

#### 7.11.2.5 `int agent_task_s::init_agent_status`

Definition at line 47 of file `agent_task.h`.

Referenced by `agent_task_Copy()`.

#### 7.11.2.6 `char* agent_task_s::var_name`

Definition at line 49 of file `agent_task.h`.

Referenced by `agent_task_Copy()`, `agent_task_Destroy()`, `agent_xml_compose__task()`, `agent_xml_parse__task()`, `interpreter_variable_data_InitializeFromAgent()`, and `MC_ComposeAgent()`.

#### 7.11.2.7 `char* agent_task_s::server_name`

Definition at line 50 of file `agent_task.h`.

Referenced by `agent_task_Copy()`, `agent_task_Destroy()`, `agent_xml_compose__task()`, `agent_xml_parse__task()`, `MC_ComposeAgent()`, and `message_InitializeFromAgent()`.

#### 7.11.2.8 `char* agent_task_s::code_id`

Definition at line 51 of file `agent_task.h`.

Referenced by `agent_task_Copy()`, `agent_task_Destroy()`, `agent_xml_compose__task()`, `agent_xml_parse__agent_code()`, and `agent_xml_parse__task()`.

#### 7.11.2.9 `interpreter_variable_data_t* agent_task_s::agent_return_data`

Definition at line 54 of file `agent_task.h`.

Referenced by `agent_task_Copy()`, `agent_task_Destroy()`, `agent_xml_compose__task()`, `agent_xml_parse__data()`, and `MC_GetAgentReturnData()`.

#### 7.11.2.10 `struct agent_variable_list_s* agent_task_s::agent_variable_list` [read]

Definition at line 55 of file `agent_task.h`.

Referenced by `agent_AddPersistentVariable()`, `agent_task_Destroy()`, `agent_task_New()`, `agent_xml_compose__task()`, `agent_xml_parse__data()`, and `MC_AgentVariableRetrieve()`.

#### 7.11.2.11 `char** agent_task_s::saved_variables`

Definition at line 56 of file `agent_task.h`.

Referenced by `agent_task_Destroy()`, `agent_task_New()`, and `MC_AgentVariableSave()`.

#### 7.11.2.12 `int agent_task_s::num_saved_variables`

Definition at line 57 of file `agent_task.h`.

Referenced by `agent_task_New()`, and `MC_AgentVariableSave()`.

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



- [/home/dko/projects/mobilec/trunk/src/include/agent\\_task.h](#)

## 7.12 arc4\_context Struct Reference

ARC4 context structure.

```
#include <arc4.h>
```

### Data Fields

- [int m](#) [256]
- [int x](#)
- [int y](#)

### 7.12.1 Detailed Description

ARC4 context structure.

Definition at line 17 of file arc4.h.

### 7.12.2 Field Documentation

#### 7.12.2.1 int arc4\_context::m[256]

permutation table

Definition at line 19 of file arc4.h.

Referenced by arc4\_crypt(), and arc4\_setup().

#### 7.12.2.2 int arc4\_context::x

permutation index

Definition at line 20 of file arc4.h.

Referenced by arc4\_crypt(), and arc4\_setup().

#### 7.12.2.3 int arc4\_context::y

permutation index

Definition at line 21 of file arc4.h.

Referenced by arc4\_crypt(), and arc4\_setup().

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

- [/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/arc4.h](#)

## 7.13 barrier\_node\_s Struct Reference

```
#include <barrier.h>
```

### Data Fields

- `MUTEX_T * lock`
- `COND_T * cond`
- `int id`
- `int num_registered`
- `int num_waiting`

### 7.13.1 Detailed Description

Definition at line 42 of file barrier.h.

### 7.13.2 Field Documentation

#### 7.13.2.1 `MUTEX_T* barrier_node_s::lock`

Definition at line 43 of file barrier.h.

Referenced by `barrier_node_Destroy()`, `barrier_node_Initialize()`, and `MC_Barrier()`.

#### 7.13.2.2 `COND_T* barrier_node_s::cond`

Definition at line 44 of file barrier.h.

Referenced by `barrier_node_Destroy()`, `barrier_node_Initialize()`, and `MC_Barrier()`.

#### 7.13.2.3 `int barrier_node_s::id`

Definition at line 45 of file barrier.h.

Referenced by `barrier_node_Initialize()`, `barrier_queue_Add()`, and `barrier_queue_Delete()`.

#### 7.13.2.4 `int barrier_node_s::num_registered`

Definition at line 46 of file barrier.h.

Referenced by `barrier_node_Initialize()`, and `MC_Barrier()`.

#### 7.13.2.5 `int barrier_node_s::num_waiting`

Definition at line 47 of file barrier.h.

Referenced by `barrier_node_Initialize()`, and `MC_Barrier()`.

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

- `/home/dko/projects/mobilec/trunk/src/include/barrier.h`

## 7.14 barrier\_queue\_s Struct Reference

```
#include <barrier.h>
```

### Data Fields

- `RWLOCK_T * lock`
- `list_p list`
- `int size`

### 7.14.1 Detailed Description

Definition at line 51 of file barrier.h.

### 7.14.2 Field Documentation

#### 7.14.2.1 `RWLOCK_T* barrier_queue_s::lock`

Definition at line 52 of file barrier.h.

Referenced by `barrier_queue_Add()`, `barrier_queue_Delete()`, `barrier_queue_Destroy()`, `barrier_queue_Get()`, and `barrier_queue_New()`.

#### 7.14.2.2 `list_p barrier_queue_s::list`

Definition at line 54 of file barrier.h.

Referenced by `barrier_queue_Add()`, `barrier_queue_Delete()`, `barrier_queue_Destroy()`, `barrier_queue_Get()`, `barrier_queue_New()`, and `barrier_queue_Pop()`.

#### 7.14.2.3 `int barrier_queue_s::size`

Definition at line 55 of file barrier.h.

Referenced by `barrier_queue_Add()`, and `barrier_queue_Delete()`.

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

- `/home/dko/projects/mobilec/trunk/src/include/barrier.h`

## 7.15 cmd\_prompt\_s Struct Reference

```
#include <cmd_prompt.h>
```

### Data Fields

- `THREAD_T` [thread](#)

### 7.15.1 Detailed Description

Definition at line 41 of file `cmd_prompt.h`.

### 7.15.2 Field Documentation

#### 7.15.2.1 `THREAD_T cmd_prompt_s::thread`

Definition at line 42 of file `cmd_prompt.h`.

Referenced by `cmd_prompt_Start()`, and `MC_End()`.

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

- `/home/dko/projects/mobilec/trunk/src/include/cmd_prompt.h`

## 7.16 `command_s` Struct Reference

```
#include <cmd_prompt.h>
```

### Data Fields

- [int index](#)
- [int num\\_args](#)
- [char \\*\\* args](#)

### 7.16.1 Detailed Description

Definition at line 46 of file `cmd_prompt.h`.

### 7.16.2 Field Documentation

#### 7.16.2.1 `int command_s::index`

Definition at line 47 of file `cmd_prompt.h`.

Referenced by `cmd_prompt_Thread()`, and `process_command()`.

#### 7.16.2.2 `int command_s::num_args`

Definition at line 48 of file `cmd_prompt.h`.

Referenced by `cmd_prompt_Thread()`, and `process_command()`.

#### 7.16.2.3 `char** command_s::args`

Definition at line 49 of file `cmd_prompt.h`.

Referenced by `cmd_prompt_Thread()`, and `process_command()`.

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

- [/home/dko/projects/mobilec/trunk/src/include/cmd\\_prompt.h](#)

## 7.17 connection\_s Struct Reference

```
#include <connection.h>
```

### Data Fields

- [int connect\\_id](#)
- [char \\* remote\\_hostname](#)
- [struct sockaddr\\_in addr](#)
- [u\\_long clientfd](#)
- [u\\_long serverfd](#)

### 7.17.1 Detailed Description

Definition at line 45 of file connection.h.

### 7.17.2 Field Documentation

#### 7.17.2.1 int connection\_s::connect\_id

Definition at line 48 of file connection.h.

Referenced by `AP_QUEUE_SEARCH_TEMPLATE()`, `connection_Copy()`, and `message_InitializeFromConnection()`.

#### 7.17.2.2 char\* connection\_s::remote\_hostname

Definition at line 51 of file connection.h.

Referenced by `AP_QUEUE_SEARCH_TEMPLATE()`, `connection_Copy()`, and `connection_Destroy()`.

#### 7.17.2.3 struct sockaddr\_in connection\_s::addr [read]

Definition at line 53 of file connection.h.

Referenced by `connection_Copy()`, and `message_InitializeFromConnection()`.

#### 7.17.2.4 u\_long connection\_s::clientfd

Definition at line 56 of file connection.h.

Referenced by `connection_Copy()`, `connection_Destroy()`, `message_InitializeFromConnection()`, and `mtp_http_InitializeFromConnection()`.

#### 7.17.2.5 u\_long connection\_s::serverfd

Definition at line 57 of file connection.h.

Referenced by `connection_Copy()`.

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

- </home/dko/projects/mobilec/trunk/src/include/connection.h>



## 7.18 des3\_context Struct Reference

Triple-DES context structure.

```
#include <des.h>
```

### Data Fields

- unsigned long [esk](#) [96]
- unsigned long [dsk](#) [96]

### 7.18.1 Detailed Description

Triple-DES context structure.

Definition at line 27 of file des.h.

### 7.18.2 Field Documentation

#### 7.18.2.1 unsigned long des3\_context::esk[96]

Triple-DES encryption subkeys

Definition at line 29 of file des.h.

Referenced by [des3\\_cbc\\_encrypt\(\)](#), [des3\\_encrypt\(\)](#), [des3\\_set\\_2keys\(\)](#), and [des3\\_set\\_3keys\(\)](#).

#### 7.18.2.2 unsigned long des3\_context::dsk[96]

Triple-DES decryption subkeys

Definition at line 30 of file des.h.

Referenced by [des3\\_cbc\\_decrypt\(\)](#), [des3\\_decrypt\(\)](#), [des3\\_set\\_2keys\(\)](#), and [des3\\_set\\_3keys\(\)](#).

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

- [/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/des.h](#)

## 7.19 des\_context Struct Reference

DES context structure.

```
#include <des.h>
```

### Data Fields

- unsigned long [esk](#) [32]
- unsigned long [dsk](#) [32]

### 7.19.1 Detailed Description

DES context structure.

Definition at line 17 of file `des.h`.

### 7.19.2 Field Documentation

#### 7.19.2.1 unsigned long des\_context::esk[32]

DES encryption subkeys

Definition at line 19 of file `des.h`.

Referenced by `des_cbc_encrypt()`, `des_encrypt()`, and `des_set_key()`.

#### 7.19.2.2 unsigned long des\_context::dsk[32]

DES decryption subkeys

Definition at line 20 of file `des.h`.

Referenced by `des_cbc_decrypt()`, `des_decrypt()`, and `des_set_key()`.

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

- `/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/des.h`

## 7.20 dhm\_context Struct Reference

```
#include <dhm.h>
```

### Data Fields

- [int len](#)
- [mpi P](#)
- [mpi G](#)
- [mpi X](#)
- [mpi GX](#)
- [mpi GY](#)
- [mpi K](#)
- [mpi RP](#)

### 7.20.1 Detailed Description

Definition at line 23 of file dhm.h.

### 7.20.2 Field Documentation

#### 7.20.2.1 int dhm\_context::len

size(P) in chars

Definition at line 25 of file dhm.h.

Referenced by `dhm_make_params()`, `dhm_make_public()`, `dhm_read_params()`, `dhm_read_public()`, `main()`, `ssl_parse_client_key_exchange()`, `ssl_parse_server_key_exchange()`, and `ssl_write_client_key_exchange()`.

#### 7.20.2.2 mpi dhm\_context::P

prime modulus

Definition at line 26 of file dhm.h.

Referenced by `dhm_calc_secret()`, `dhm_free()`, `dhm_make_params()`, `dhm_make_public()`, `dhm_read_params()`, and `main()`.

#### 7.20.2.3 mpi dhm\_context::G

generator

Definition at line 27 of file dhm.h.

Referenced by `dhm_free()`, `dhm_make_params()`, `dhm_make_public()`, `dhm_read_params()`, and `main()`.

**7.20.2.4 mpi dhm\_context::X**

secret value

Definition at line 28 of file dhm.h.

Referenced by dhm\_calc\_secret(), dhm\_free(), dhm\_make\_params(), and dhm\_make\_public().

**7.20.2.5 mpi dhm\_context::GX**

$\text{self} = G^X \bmod P$

Definition at line 29 of file dhm.h.

Referenced by dhm\_free(), dhm\_make\_params(), and dhm\_make\_public().

**7.20.2.6 mpi dhm\_context::GY**

$\text{peer} = G^Y \bmod P$

Definition at line 30 of file dhm.h.

Referenced by dhm\_calc\_secret(), dhm\_free(), dhm\_read\_params(), and dhm\_read\_public().

**7.20.2.7 mpi dhm\_context::K**

$\text{key} = GY^X \bmod P$

Definition at line 31 of file dhm.h.

Referenced by dhm\_calc\_secret(), and dhm\_free().

**7.20.2.8 mpi dhm\_context::RP**

cached  $R^2 \bmod P$

Definition at line 32 of file dhm.h.

Referenced by dhm\_calc\_secret(), dhm\_free(), dhm\_make\_params(), and dhm\_make\_public().

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

- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/[dhm.h](#)

## 7.21 dynstring\_s Struct Reference

```
#include <dynstring.h>
```

### Data Fields

- [int len](#)
- [int size](#)
- [char \\* message](#)

### 7.21.1 Detailed Description

Definition at line 9 of file dynstring.h.

### 7.21.2 Field Documentation

#### 7.21.2.1 int dynstring\_s::len

Definition at line 10 of file dynstring.h.

Referenced by `dynstring_Append()`, `dynstring_New()`, and `mtp_http_CreateMessage()`.

#### 7.21.2.2 int dynstring\_s::size

Definition at line 11 of file dynstring.h.

Referenced by `dynstring_Append()`, and `dynstring_New()`.

#### 7.21.2.3 char\* dynstring\_s::message

Definition at line 12 of file dynstring.h.

Referenced by `dynstring_Append()`, `dynstring_Destroy()`, `dynstring_New()`, `fipa_envelope_Compose__from()`, `MC_AclSend()`, and `mtp_http_CreateMessage()`.

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

- [/home/dko/projects/mobilec/trunk/src/include/dynstring.h](#)

## 7.22 fipa\_acl\_envelope\_Received\_s Struct Reference

```
#include <fipa_acl_envelope.h>
```

### Data Fields

- struct [fipa\\_url\\_s](#) \* [received\\_by](#)
- struct [fipa\\_url\\_s](#) \* [received\\_from](#)
- struct [fipa\\_DateTime\\_s](#) \* [received\\_date](#)
- char \* [received\\_id](#)
- struct [fipa\\_url\\_s](#) \* [received\\_via](#)

### 7.22.1 Detailed Description

Definition at line 7 of file `fipa_acl_envelope.h`.

### 7.22.2 Field Documentation

#### 7.22.2.1 struct [fipa\\_url\\_s](#)\* [fipa\\_acl\\_envelope\\_Received\\_s::received\\_by](#) [read]

Definition at line 9 of file `fipa_acl_envelope.h`.

Referenced by `fipa_acl_envelope_Received_Copy()`, `fipa_acl_envelope_Received_Destroy()`, and `fipa_envelope_HandleReceived()`.

#### 7.22.2.2 struct [fipa\\_url\\_s](#)\* [fipa\\_acl\\_envelope\\_Received\\_s::received\\_from](#) [read]

Definition at line 10 of file `fipa_acl_envelope.h`.

Referenced by `fipa_acl_envelope_Received_Copy()`, `fipa_acl_envelope_Received_Destroy()`, and `fipa_envelope_HandleReceived()`.

#### 7.22.2.3 struct [fipa\\_DateTime\\_s](#)\* [fipa\\_acl\\_envelope\\_Received\\_s::received\\_date](#) [read]

Definition at line 11 of file `fipa_acl_envelope.h`.

Referenced by `fipa_acl_envelope_Received_Copy()`, `fipa_acl_envelope_Received_Destroy()`, and `fipa_envelope_HandleReceived()`.

#### 7.22.2.4 char\* [fipa\\_acl\\_envelope\\_Received\\_s::received\\_id](#)

Definition at line 12 of file `fipa_acl_envelope.h`.

Referenced by `fipa_acl_envelope_Received_Copy()`, `fipa_acl_envelope_Received_Destroy()`, and `fipa_envelope_HandleReceived()`.

#### 7.22.2.5 struct [fipa\\_url\\_s](#)\* [fipa\\_acl\\_envelope\\_Received\\_s::received\\_via](#) [read]

Definition at line 13 of file `fipa_acl_envelope.h`.

Referenced by `fipa_acl_envelope_Received_Copy()`, `fipa_acl_envelope_Received_Destroy()`, and `fipa_envelope_HandleReceived()`.

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

- `/home/dko/projects/mobilec/trunk/src/include/fipa_acl_envelope.h`

## 7.23 fipa\_acl\_envelope\_s Struct Reference

```
#include <fipa_acl_envelope.h>
```

### Data Fields

- [int num\\_params](#)
- [struct fipa\\_acl\\_Param\\_s \\*\\* params](#)

### 7.23.1 Detailed Description

Definition at line 40 of file [fipa\\_acl\\_envelope.h](#).

### 7.23.2 Field Documentation

#### 7.23.2.1 int fipa\_acl\_envelope\_s::num\_params

Definition at line 42 of file [fipa\\_acl\\_envelope.h](#).

Referenced by [acc\\_Thread\(\)](#), [fipa\\_acl\\_envelope\\_Copy\(\)](#), [fipa\\_acl\\_envelope\\_Destroy\(\)](#), and [fipa\\_envelope\\_HandleParams\(\)](#).

#### 7.23.2.2 struct fipa\_acl\_Param\_s\*\* fipa\_acl\_envelope\_s::params [read]

Definition at line 43 of file [fipa\\_acl\\_envelope.h](#).

Referenced by [acc\\_Thread\(\)](#), [fipa\\_acl\\_envelope\\_Copy\(\)](#), [fipa\\_acl\\_envelope\\_Destroy\(\)](#), [fipa\\_envelope\\_HandleAclRepresentation\(\)](#), [fipa\\_envelope\\_HandleComments\(\)](#), [fipa\\_envelope\\_HandleDate\(\)](#), [fipa\\_envelope\\_HandleFrom\(\)](#), [fipa\\_envelope\\_HandleIntendedReceiver\(\)](#), [fipa\\_envelope\\_HandleParams\(\)](#), [fipa\\_envelope\\_HandlePayloadEncoding\(\)](#), [fipa\\_envelope\\_HandlePayloadLength\(\)](#), [fipa\\_envelope\\_HandleReceived\(\)](#), and [fipa\\_envelope\\_HandleTo\(\)](#).

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

- [/home/dko/projects/mobilec/trunk/src/include/fipa\\_acl\\_envelope.h](#)



## 7.24 fipa\_acl\_message\_s Struct Reference

```
#include <fipa_acl.h>
```

### Data Fields

- enum [fipa\\_performative\\_e](#) performative
- struct [fipa\\_agent\\_identifier\\_s](#) \* sender
- struct [fipa\\_agent\\_identifier\\_set\\_s](#) \* receiver
- int receiver\_num
- struct [fipa\\_agent\\_identifier\\_set\\_s](#) \* reply\_to
- struct [fipa\\_string\\_s](#) \* content
- struct [fipa\\_expression\\_s](#) \* language
- struct [fipa\\_expression\\_s](#) \* encoding
- struct [fipa\\_expression\\_s](#) \* ontology
- struct [fipa\\_word\\_s](#) \* protocol
- struct [fipa\\_expression\\_s](#) \* conversation\_id
- struct [fipa\\_expression\\_s](#) \* reply\_with
- struct [fipa\\_expression\\_s](#) \* in\_reply\_to
- struct [fipa\\_DateTime\\_s](#) \* reply\_by

### 7.24.1 Detailed Description

Definition at line 81 of file `fipa_acl.h`.

### 7.24.2 Field Documentation

#### 7.24.2.1 enum fipa\_performative\_e fipa\_acl\_message\_s::performative

Definition at line 83 of file `fipa_acl.h`.

Referenced by `fipa_acl_Compose()`, `fipa_acl_message_Copy()`, `fipa_acl_Parse()`, `MC_AclSetPerformative()`, and `MC_AclSetPerformative_chdl()`.

#### 7.24.2.2 struct fipa\_agent\_identifier\_s\* fipa\_acl\_message\_s::sender [read]

Definition at line 85 of file `fipa_acl.h`.

Referenced by `fipa_acl_Compose()`, `fipa_acl_message_Copy()`, `fipa_acl_message_Destroy()`, `fipa_envelope_Compose__from()`, `fipa_message_parameter_Parse()`, `fipa_Reply()`, and `MC_AclSetSender()`.

#### 7.24.2.3 struct fipa\_agent\_identifier\_set\_s\* fipa\_acl\_message\_s::receiver [read]

Definition at line 87 of file `fipa_acl.h`.

Referenced by `AP_QUEUE_STD_DEFN_TEMPLATE()`, `fipa_acl_Compose()`, `fipa_acl_message_Copy()`, `fipa_acl_message_Destroy()`, `fipa_envelope_Compose__intended_receiver()`, `fipa_envelope_Compose__to()`, `fipa_message_parameter_Parse()`, `fipa_Reply()`, `MC_AclAddReceiver()`, and `MC_AclSend()`.

#### 7.24.2.4 `int fipa_acl_message_s::receiver_num`

Definition at line 88 of file `fipa_acl.h`.

Referenced by `MC_AclAddReceiver()`.

#### 7.24.2.5 `struct fipa_agent_identifier_set_s* fipa_acl_message_s::reply_to` [read]

Definition at line 90 of file `fipa_acl.h`.

Referenced by `fipa_acl_Compose()`, `fipa_acl_message_Copy()`, `fipa_acl_message_Destroy()`, `fipa_message_parameter_Parse()`, `fipa_Reply()`, and `MC_AclAddReplyTo()`.

#### 7.24.2.6 `struct fipa_string_s* fipa_acl_message_s::content` [read]

Definition at line 93 of file `fipa_acl.h`.

Referenced by `fipa_acl_Compose()`, `fipa_acl_message_Copy()`, `fipa_acl_message_Destroy()`, `fipa_message_parameter_Parse()`, `MC_AclSetContent()`, and `MC_AclSetContent_chdl()`.

#### 7.24.2.7 `struct fipa_expression_s* fipa_acl_message_s::language` [read]

Definition at line 95 of file `fipa_acl.h`.

Referenced by `fipa_acl_Compose()`, `fipa_acl_message_Copy()`, `fipa_acl_message_Destroy()`, and `fipa_message_parameter_Parse()`.

#### 7.24.2.8 `struct fipa_expression_s* fipa_acl_message_s::encoding` [read]

Definition at line 97 of file `fipa_acl.h`.

Referenced by `fipa_acl_Compose()`, `fipa_acl_message_Copy()`, `fipa_acl_message_Destroy()`, and `fipa_message_parameter_Parse()`.

#### 7.24.2.9 `struct fipa_expression_s* fipa_acl_message_s::ontology` [read]

Definition at line 99 of file `fipa_acl.h`.

Referenced by `fipa_acl_Compose()`, `fipa_acl_message_Copy()`, `fipa_acl_message_Destroy()`, and `fipa_message_parameter_Parse()`.

#### 7.24.2.10 `struct fipa_word_s* fipa_acl_message_s::protocol` [read]

Definition at line 101 of file `fipa_acl.h`.

Referenced by `fipa_acl_Compose()`, `fipa_acl_message_Copy()`, `fipa_acl_message_Destroy()`, and `fipa_message_parameter_Parse()`.

#### 7.24.2.11 `struct fipa_expression_s* fipa_acl_message_s::conversation_id` [read]

Definition at line 103 of file `fipa_acl.h`.

Referenced by fipa\_acl\_Compose(), fipa\_acl\_message\_Copy(), fipa\_acl\_message\_Destroy(), and fipa\_message\_parameter\_Parse().

#### 7.24.2.12 struct fipa\_expression\_s\* fipa\_acl\_message\_s::reply\_with [read]

Definition at line 105 of file fipa\_acl.h.

Referenced by fipa\_acl\_Compose(), fipa\_acl\_message\_Copy(), fipa\_acl\_message\_Destroy(), and fipa\_message\_parameter\_Parse().

#### 7.24.2.13 struct fipa\_expression\_s\* fipa\_acl\_message\_s::in\_reply\_to [read]

Definition at line 107 of file fipa\_acl.h.

Referenced by fipa\_acl\_Compose(), fipa\_acl\_message\_Copy(), fipa\_acl\_message\_Destroy(), and fipa\_message\_parameter\_Parse().

#### 7.24.2.14 struct fipa\_DateTime\_s\* fipa\_acl\_message\_s::reply\_by [read]

Definition at line 109 of file fipa\_acl.h.

Referenced by fipa\_acl\_Compose(), fipa\_acl\_message\_Copy(), fipa\_acl\_message\_Destroy(), and fipa\_message\_parameter\_Parse().

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

- [/home/dko/projects/mobilec/trunk/src/include/fipa\\_acl.h](#)

## 7.25 fipa\_acl\_Param\_s Struct Reference

```
#include <fipa_acl_envelope.h>
```

### Data Fields

- struct [fipa\\_agent\\_identifier\\_set\\_s](#) \* to
- struct [fipa\\_agent\\_identifier\\_s](#) \* from
- char \* [comments](#)
- char \* [acl\\_representation](#)
- char \* [payload\\_length](#)
- char \* [payload\\_encoding](#)
- struct [fipa\\_DateTime\\_s](#) \* date
- struct [fipa\\_agent\\_identifier\\_set\\_s](#) \* intended\_receiver
- struct [fipa\\_acl\\_envelope\\_Received\\_s](#) \* received

### 7.25.1 Detailed Description

Definition at line 22 of file `fipa_acl_envelope.h`.

### 7.25.2 Field Documentation

#### 7.25.2.1 struct fipa\_agent\_identifier\_set\_s\* fipa\_acl\_Param\_s::to [read]

Definition at line 24 of file `fipa_acl_envelope.h`.

Referenced by `acc_Thread()`, `fipa_acl_Param_Copy()`, `fipa_acl_Param_Destroy()`, and `fipa_envelope_HandleTo()`.

#### 7.25.2.2 struct fipa\_agent\_identifier\_s\* fipa\_acl\_Param\_s::from [read]

Definition at line 25 of file `fipa_acl_envelope.h`.

Referenced by `fipa_acl_Param_Copy()`, `fipa_acl_Param_Destroy()`, and `fipa_envelope_HandleFrom()`.

#### 7.25.2.3 char\* fipa\_acl\_Param\_s::comments

Definition at line 26 of file `fipa_acl_envelope.h`.

Referenced by `fipa_acl_Param_Copy()`, `fipa_acl_Param_Destroy()`, and `fipa_envelope_HandleComments()`.

#### 7.25.2.4 char\* fipa\_acl\_Param\_s::acl\_representation

Definition at line 27 of file `fipa_acl_envelope.h`.

Referenced by `fipa_acl_Param_Copy()`, `fipa_acl_Param_Destroy()`, and `fipa_envelope_HandleAclRepresentation()`.

#### 7.25.2.5 char\* fipa\_acl\_Param\_s::payload\_length

Definition at line 28 of file fipa\_acl\_envelope.h.

Referenced by fipa\_acl\_Param\_Copy(), fipa\_acl\_Param\_Destroy(), and fipa\_envelope\_HandlePayloadLength().

#### 7.25.2.6 char\* fipa\_acl\_Param\_s::payload\_encoding

Definition at line 29 of file fipa\_acl\_envelope.h.

Referenced by fipa\_acl\_Param\_Copy(), fipa\_acl\_Param\_Destroy(), and fipa\_envelope\_HandlePayloadEncoding().

#### 7.25.2.7 struct fipa\_DateTime\_s\* fipa\_acl\_Param\_s::date [read]

Definition at line 30 of file fipa\_acl\_envelope.h.

Referenced by fipa\_acl\_Param\_Copy(), fipa\_acl\_Param\_Destroy(), and fipa\_envelope\_HandleDate().

#### 7.25.2.8 struct fipa\_agent\_identifier\_set\_s\* fipa\_acl\_Param\_s::intended\_receiver [read]

Definition at line 31 of file fipa\_acl\_envelope.h.

Referenced by fipa\_acl\_Param\_Copy(), fipa\_acl\_Param\_Destroy(), and fipa\_envelope\_HandleIntendedReceiver().

#### 7.25.2.9 struct fipa\_acl\_envelope\_Received\_s\* fipa\_acl\_Param\_s::received [read]

Definition at line 32 of file fipa\_acl\_envelope.h.

Referenced by fipa\_acl\_Param\_Copy(), fipa\_acl\_Param\_Destroy(), and fipa\_envelope\_HandleReceived().

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

- /home/dko/projects/mobilec/trunk/src/include/fipa\_acl\_envelope.h

## 7.26 fipa\_agent\_identifier\_s Struct Reference

```
#include <fipa_acl.h>
```

### Data Fields

- char \* [name](#)
- struct [fipa\\_url\\_sequence\\_s](#) \* [addresses](#)
- struct [fipa\\_agent\\_identifier\\_set\\_s](#) \* [resolvers](#)

### 7.26.1 Detailed Description

Definition at line 155 of file [fipa\\_acl.h](#).

### 7.26.2 Field Documentation

#### 7.26.2.1 char\* fipa\_agent\_identifier\_s::name

Definition at line 157 of file [fipa\\_acl.h](#).

Referenced by [acc\\_Thread\(\)](#), [AP\\_QUEUE\\_STD\\_DEFN\\_TEMPLATE\(\)](#), [fipa\\_agent\\_identifier\\_Compose\(\)](#), [fipa\\_agent\\_identifier\\_Copy\(\)](#), [fipa\\_agent\\_identifier\\_Destroy\(\)](#), [fipa\\_envelope\\_Compose\\_\\_from\(\)](#), [fipa\\_envelope\\_Compose\\_\\_intended\\_receiver\(\)](#), [fipa\\_envelope\\_Compose\\_\\_to\(\)](#), [MC\\_AclAddReceiver\(\)](#), [MC\\_AclAddReplyTo\(\)](#), [MC\\_AclSend\(\)](#), and [MC\\_AclSetSender\(\)](#).

#### 7.26.2.2 struct fipa\_url\_sequence\_s\* fipa\_agent\_identifier\_s::addresses [read]

Definition at line 158 of file [fipa\\_acl.h](#).

Referenced by [fipa\\_agent\\_identifier\\_Compose\(\)](#), [fipa\\_agent\\_identifier\\_Copy\(\)](#), [fipa\\_agent\\_identifier\\_Destroy\(\)](#), [fipa\\_envelope\\_Compose\\_\\_from\(\)](#), [fipa\\_envelope\\_Compose\\_\\_intended\\_receiver\(\)](#), [fipa\\_envelope\\_Compose\\_\\_to\(\)](#), [fipa\\_envelope\\_ParseAddresses\(\)](#), [MC\\_AclAddReceiver\(\)](#), [MC\\_AclAddReplyTo\(\)](#), [MC\\_AclSend\(\)](#), and [MC\\_AclSetSender\(\)](#).

#### 7.26.2.3 struct fipa\_agent\_identifier\_set\_s\* fipa\_agent\_identifier\_s::resolvers [read]

Definition at line 159 of file [fipa\\_acl.h](#).

Referenced by [fipa\\_agent\\_identifier\\_Compose\(\)](#), [fipa\\_agent\\_identifier\\_Copy\(\)](#), [fipa\\_agent\\_identifier\\_Destroy\(\)](#), and [fipa\\_envelope\\_ParseResolvers\(\)](#).

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

- [/home/dko/projects/mobilec/trunk/src/include/fipa\\_acl.h](#)

## 7.27 fipa\_agent\_identifier\_set\_s Struct Reference

```
#include <fipa_acl.h>
```

### Data Fields

- [int num](#)
- [int retain\\_order](#)
- struct [fipa\\_agent\\_identifier\\_s](#) \*\* [fipa\\_agent\\_identifiers](#)

### 7.27.1 Detailed Description

Definition at line 141 of file [fipa\\_acl.h](#).

### 7.27.2 Field Documentation

#### 7.27.2.1 int fipa\_agent\_identifier\_set\_s::num

Definition at line 143 of file [fipa\\_acl.h](#).

Referenced by [acc\\_Thread\(\)](#), [fipa\\_agent\\_identifier\\_Compose\(\)](#), [fipa\\_agent\\_identifier\\_set\\_Compose\(\)](#), [fipa\\_agent\\_identifier\\_set\\_Copy\(\)](#), [fipa\\_agent\\_identifier\\_set\\_Destroy\(\)](#), [fipa\\_envelope\\_Compose\\_\\_to\(\)](#), [fipa\\_envelope\\_HandleIntendedReceiver\(\)](#), [fipa\\_envelope\\_HandleTo\(\)](#), [fipa\\_envelope\\_ParseResolvers\(\)](#), [fipa\\_Reply\(\)](#), [MC\\_AclAddReceiver\(\)](#), [MC\\_AclAddReplyTo\(\)](#), and [MC\\_AclSend\(\)](#).

#### 7.27.2.2 int fipa\_agent\_identifier\_set\_s::retain\_order

Definition at line 144 of file [fipa\\_acl.h](#).

Referenced by [fipa\\_agent\\_identifier\\_set\\_Copy\(\)](#), [fipa\\_agent\\_identifier\\_set\\_Parse\(\)](#), [fipa\\_envelope\\_HandleIntendedReceiver\(\)](#), [fipa\\_envelope\\_HandleTo\(\)](#), [fipa\\_envelope\\_ParseResolvers\(\)](#), and [fipa\\_Reply\(\)](#).

#### 7.27.2.3 struct fipa\_agent\_identifier\_s\*\* fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers [read]

Definition at line 145 of file [fipa\\_acl.h](#).

Referenced by [acc\\_Thread\(\)](#), [AP\\_QUEUE\\_STD\\_DEFN\\_TEMPLATE\(\)](#), [fipa\\_agent\\_identifier\\_set\\_Compose\(\)](#), [fipa\\_agent\\_identifier\\_set\\_Copy\(\)](#), [fipa\\_agent\\_identifier\\_set\\_Destroy\(\)](#), [fipa\\_envelope\\_Compose\\_\\_intended\\_receiver\(\)](#), [fipa\\_envelope\\_Compose\\_\\_to\(\)](#), [fipa\\_envelope\\_HandleIntendedReceiver\(\)](#), [fipa\\_envelope\\_HandleTo\(\)](#), [fipa\\_envelope\\_ParseResolvers\(\)](#), [fipa\\_Reply\(\)](#), [MC\\_AclAddReceiver\(\)](#), [MC\\_AclAddReplyTo\(\)](#), and [MC\\_AclSend\(\)](#).

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

- [/home/dko/projects/mobilec/trunk/src/include/fipa\\_acl.h](#)

## 7.28 fipa\_DateTime\_s Struct Reference

```
#include <fipa_acl.h>
```

### Data Fields

- [int year](#)
- [int month](#)
- [int day](#)
- [int hour](#)
- [int minute](#)
- [int second](#)
- [int millisecond](#)
- [char sign](#)
- [int is\\_utc](#)

### 7.28.1 Detailed Description

Definition at line 209 of file fipa\_acl.h.

### 7.28.2 Field Documentation

#### 7.28.2.1 int fipa\_DateTime\_s::year

Definition at line 211 of file fipa\_acl.h.

Referenced by fipa\_DateTime\_Compose().

#### 7.28.2.2 int fipa\_DateTime\_s::month

Definition at line 212 of file fipa\_acl.h.

Referenced by fipa\_DateTime\_Compose().

#### 7.28.2.3 int fipa\_DateTime\_s::day

Definition at line 213 of file fipa\_acl.h.

Referenced by fipa\_DateTime\_Compose().

#### 7.28.2.4 int fipa\_DateTime\_s::hour

Definition at line 214 of file fipa\_acl.h.

Referenced by fipa\_DateTime\_Compose().

#### 7.28.2.5 int fipa\_DateTime\_s::minute

Definition at line 215 of file fipa\_acl.h.

Referenced by fipa\_DateTime\_Compose().



**7.28.2.6 int fipa\_DateTime\_s::second**

Definition at line 216 of file fipa\_acl.h.

Referenced by fipa\_DateTime\_Compose().

**7.28.2.7 int fipa\_DateTime\_s::millisecond**

Definition at line 217 of file fipa\_acl.h.

Referenced by fipa\_DateTime\_Compose().

**7.28.2.8 char fipa\_DateTime\_s::sign**

Definition at line 218 of file fipa\_acl.h.

Referenced by fipa\_DateTime\_Compose(), and fipa\_datetime\_Parse().

**7.28.2.9 int fipa\_DateTime\_s::is\_utc**

Definition at line 219 of file fipa\_acl.h.

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

- [/home/dko/projects/mobilec/trunk/src/include/fipa\\_acl.h](#)

## 7.29 fipa\_expression\_s Struct Reference

```
#include <fipa_acl.h>
```

### Data Fields

- enum [fipa\\_expression\\_type\\_e](#) type
- union [fipa\\_expression\\_s::content\\_u](#) content

### Data Structures

- union [content\\_u](#)

#### 7.29.1 Detailed Description

Definition at line 168 of file [fipa\\_acl.h](#).

#### 7.29.2 Field Documentation

##### 7.29.2.1 enum [fipa\\_expression\\_type\\_e](#) [fipa\\_expression\\_s::type](#)

Definition at line 170 of file [fipa\\_acl.h](#).

Referenced by [fipa\\_expression\\_Compose\(\)](#), [fipa\\_expression\\_Copy\(\)](#), [fipa\\_expression\\_Destroy\(\)](#), and [fipa\\_expression\\_Parse\(\)](#).

##### 7.29.2.2 union [fipa\\_expression\\_s::content\\_u](#) [fipa\\_expression\\_s::content](#)

Referenced by [fipa\\_expression\\_Compose\(\)](#), [fipa\\_expression\\_Copy\(\)](#), and [fipa\\_expression\\_Destroy\(\)](#).

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

- [/home/dko/projects/mobilec/trunk/src/include/fipa\\_acl.h](#)

## 7.30 fipa\_expression\_s::content\_u Union Reference

```
#include <fipa_acl.h>
```

### Data Fields

- struct [fipa\\_word\\_s](#) \* [word](#)
- struct [fipa\\_string\\_s](#) \* [string](#)
- struct [fipa\\_number\\_s](#) \* [number](#)
- struct [fipa\\_DateTime\\_s](#) \* [datetime](#)
- struct [fipa\\_expression\\_s](#) \*\* [expression](#)

### 7.30.1 Detailed Description

Definition at line 171 of file [fipa\\_acl.h](#).

### 7.30.2 Field Documentation

#### 7.30.2.1 struct [fipa\\_word\\_s](#)\* [fipa\\_expression\\_s::content\\_u::word](#) [read]

Definition at line 173 of file [fipa\\_acl.h](#).

Referenced by [fipa\\_expression\\_Compose\(\)](#), [fipa\\_expression\\_Copy\(\)](#), and [fipa\\_expression\\_Destroy\(\)](#).

#### 7.30.2.2 struct [fipa\\_string\\_s](#)\* [fipa\\_expression\\_s::content\\_u::string](#) [read]

Definition at line 174 of file [fipa\\_acl.h](#).

Referenced by [fipa\\_expression\\_Compose\(\)](#), [fipa\\_expression\\_Copy\(\)](#), and [fipa\\_expression\\_Destroy\(\)](#).

#### 7.30.2.3 struct [fipa\\_number\\_s](#)\* [fipa\\_expression\\_s::content\\_u::number](#) [read]

Definition at line 175 of file [fipa\\_acl.h](#).

Referenced by [fipa\\_expression\\_Compose\(\)](#), [fipa\\_expression\\_Copy\(\)](#), and [fipa\\_expression\\_Destroy\(\)](#).

#### 7.30.2.4 struct [fipa\\_DateTime\\_s](#)\* [fipa\\_expression\\_s::content\\_u::datetime](#) [read]

Definition at line 176 of file [fipa\\_acl.h](#).

Referenced by [fipa\\_expression\\_Compose\(\)](#), [fipa\\_expression\\_Copy\(\)](#), and [fipa\\_expression\\_Destroy\(\)](#).

#### 7.30.2.5 struct [fipa\\_expression\\_s](#)\*\* [fipa\\_expression\\_s::content\\_u::expression](#) [read]

Definition at line 177 of file [fipa\\_acl.h](#).

Referenced by [fipa\\_expression\\_Compose\(\)](#), [fipa\\_expression\\_Copy\(\)](#), and [fipa\\_expression\\_Destroy\(\)](#).

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

- [/home/dko/projects/mobilec/trunk/src/include/fipa\\_acl.h](#)

## 7.31 fipa\_message\_string\_s Struct Reference

```
#include <fipa_acl.h>
```

### Data Fields

- char \* [message](#)
- char \* [parse](#)

### 7.31.1 Detailed Description

Definition at line 118 of file `fipa_acl.h`.

### 7.31.2 Field Documentation

#### 7.31.2.1 char\* fipa\_message\_string\_s::message

Definition at line 120 of file `fipa_acl.h`.

Referenced by `acc_Thread()`, `fipa_envelope_HandleDate()`, `fipa_envelope_HandleReceived()`, `fipa_message_string_Copy()`, and `fipa_message_string_Destroy()`.

#### 7.31.2.2 char\* fipa\_message\_string\_s::parse

Definition at line 121 of file `fipa_acl.h`.

Referenced by `acc_Thread()`, `fipa_agent_identifier_Parse()`, `fipa_CheckNextToken()`, `fipa_datetime_Parse()`, `fipa_envelope_HandleDate()`, `fipa_envelope_HandleReceived()`, `fipa_GetAtom()`, `fipa_GetNextWord()`, `fipa_GetWholeToken()`, `fipa_message_string_Copy()`, `fipa_string_Parse()`, and `fipa_word_Parse()`.

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

- `/home/dko/projects/mobilec/trunk/src/include/fipa_acl.h`

## 7.32 fipa\_number\_s Struct Reference

```
#include <fipa_acl.h>
```

### Data Fields

- char \* [str](#)

### 7.32.1 Detailed Description

Definition at line 239 of file fipa\_acl.h.

### 7.32.2 Field Documentation

#### 7.32.2.1 char\* fipa\_number\_s::str

Definition at line 241 of file fipa\_acl.h.

Referenced by fipa\_number\_Compose(), fipa\_number\_Copy(), and fipa\_number\_Destroy().

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

- /home/dko/projects/mobilec/trunk/src/include/[fipa\\_acl.h](#)

## 7.33 fipa\_string\_s Struct Reference

```
#include <fipa_acl.h>
```

### Data Fields

- char \* [content](#)

### 7.33.1 Detailed Description

Definition at line 198 of file fipa\_acl.h.

### 7.33.2 Field Documentation

#### 7.33.2.1 char\* fipa\_string\_s::content

Definition at line 200 of file fipa\_acl.h.

Referenced by fipa\_string\_Compose(), fipa\_string\_Copy(), fipa\_string\_Destroy(), fipa\_string\_Parse(), and MC\_AclSetContent().

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

- [/home/dko/projects/mobilec/trunk/src/include/fipa\\_acl.h](#)

## 7.34 fipa\_url\_s Struct Reference

```
#include <fipa_acl.h>
```

### Data Fields

- char \* [str](#)

### 7.34.1 Detailed Description

Definition at line 228 of file fipa\_acl.h.

### 7.34.2 Field Documentation

#### 7.34.2.1 char\* fipa\_url\_s::str

Definition at line 230 of file fipa\_acl.h.

Referenced by fipa\_envelope\_Compose\_\_intended\_receiver(), fipa\_envelope\_Compose\_\_to(), fipa\_envelope\_HandleReceived(), fipa\_envelope\_ParseAddresses(), fipa\_url\_Compose(), fipa\_url\_Copy(), fipa\_url\_Destroy(), MC\_AclAddReceiver(), MC\_AclAddReplyTo(), MC\_AclSend(), and MC\_AclSetSender().

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

- [/home/dko/projects/mobilec/trunk/src/include/fipa\\_acl.h](#)

## 7.35 fipa\_url\_sequence\_s Struct Reference

```
#include <fipa_acl.h>
```

### Data Fields

- `int num`
- `struct fipa_url_s ** urls`

### 7.35.1 Detailed Description

Definition at line 129 of file `fipa_acl.h`.

### 7.35.2 Field Documentation

#### 7.35.2.1 `int fipa_url_sequence_s::num`

Definition at line 131 of file `fipa_acl.h`.

Referenced by `fipa_agent_identifier_Compose()`, `fipa_envelope_Compose__from()`, `fipa_envelope_Compose__intended_receiver()`, `fipa_envelope_Compose__to()`, `fipa_envelope_ParseAddresses()`, `fipa_url_sequence_Compose()`, `fipa_url_sequence_Copy()`, `fipa_url_sequence_Destroy()`, `MC_AclAddReceiver()`, `MC_AclAddReplyTo()`, `MC_AclSend()`, and `MC_AclSetSender()`.

#### 7.35.2.2 `struct fipa_url_s** fipa_url_sequence_s::urls` [read]

Definition at line 132 of file `fipa_acl.h`.

Referenced by `fipa_envelope_Compose__intended_receiver()`, `fipa_envelope_Compose__to()`, `fipa_envelope_ParseAddresses()`, `fipa_url_sequence_Compose()`, `fipa_url_sequence_Copy()`, `fipa_url_sequence_Destroy()`, `MC_AclAddReceiver()`, `MC_AclAddReplyTo()`, `MC_AclSend()`, and `MC_AclSetSender()`.

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

- `/home/dko/projects/mobilec/trunk/src/include/fipa_acl.h`



## 7.36 fipa\_word\_s Struct Reference

```
#include <fipa_acl.h>
```

### Data Fields

- char \* [content](#)

### 7.36.1 Detailed Description

Definition at line 187 of file fipa\_acl.h.

### 7.36.2 Field Documentation

#### 7.36.2.1 char\* fipa\_word\_s::content

Definition at line 189 of file fipa\_acl.h.

Referenced by fipa\_agent\_identifier\_Parse(), fipa\_agent\_identifier\_set\_Parse(), fipa\_message\_parameter\_Parse(), fipa\_message\_type\_Parse(), fipa\_url\_Parse(), fipa\_url\_sequence\_Parse(), fipa\_word\_Compose(), fipa\_word\_Copy(), and fipa\_word\_Destroy().

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

- /home/dko/projects/mobilec/trunk/src/include/[fipa\\_acl.h](#)

## 7.37 foo\_c Class Reference

### Public Member Functions

- [foo\\_c](#) (float *f*, int *b*)
- [~foo\\_c](#) ()
- [int get\\_bar](#) ()
- [float get\\_foo](#) ()
- [void set\\_bar](#) (int *b*)
- [void set\\_foo](#) (float *f*)
- [void set\\_foobar](#) (float *f*, int *b*=0)
- [int get\\_barfoo](#) ()

### Protected Member Functions

- [int get\\_global](#) ()

### Static Protected Attributes

- static [int global](#)

### Private Attributes

- [float foo](#)
- [int bar](#)
- [int barfoo](#)

#### 7.37.1 Detailed Description

Definition at line 1 of file class.cxx.

#### 7.37.2 Constructor & Destructor Documentation

##### 7.37.2.1 foo\_c::foo\_c (float *f*, int *b*)

##### 7.37.2.2 foo\_c::~~foo\_c ()

Definition at line 81 of file class.cxx.

#### 7.37.3 Member Function Documentation

##### 7.37.3.1 int foo\_c::get\_bar () [inline]

Definition at line 13 of file class.cxx.

References [bar](#).

**7.37.3.2 float foo\_c::get\_foo ()** [inline]

Definition at line 20 of file class.cxx.

References foo.

**7.37.3.3 void foo\_c::set\_bar (int *b*)** [inline]

Definition at line 27 of file class.cxx.

References bar.

**7.37.3.4 void foo\_c::set\_foo (float *f*)** [inline]

Definition at line 34 of file class.cxx.

References foo.

**7.37.3.5 void foo\_c::set\_foobar (float *f*, int *b* = 0)** [inline]

Definition at line 41 of file class.cxx.

References bar, and foo.

**7.37.3.6 int foo\_c::get\_global ()** [inline, protected]

Definition at line 54 of file class.cxx.

References global.

**7.37.3.7 int foo\_c::get\_barfoo ()** [inline]

Definition at line 67 of file class.cxx.

References barfoo.

**7.37.4 Field Documentation****7.37.4.1 float foo\_c::foo** [private]

Definition at line 3 of file class.cxx.

Referenced by get\_foo(), set\_foo(), and set\_foobar().

**7.37.4.2 int foo\_c::bar** [private]

Definition at line 4 of file class.cxx.

Referenced by get\_bar(), set\_bar(), and set\_foobar().

**7.37.4.3 int foo\_c::global** [static, protected]

Definition at line 50 of file class.cxx.

Referenced by `get_global()`.

**7.37.4.4 int foo\_c::barfoo** [private]

Definition at line 61 of file class.cxx.

Referenced by `get_barfoo()`.

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

- [/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/test/class.cxx](#)

## 7.38 foo\_s Struct Reference

### Public Member Functions

- [foo\\_s](#) (float [f](#), int [b](#))
- [~foo\\_s](#) ()
- [int get\\_bar](#) ()
- [float get\\_foo](#) ()
- [void set\\_bar](#) (int [b](#))
- [void set\\_foo](#) (float [f](#))

### Data Fields

- [float foo](#)
- [int bar](#)

#### 7.38.1 Detailed Description

Definition at line 1 of file struct.cxx.

#### 7.38.2 Constructor & Destructor Documentation

##### 7.38.2.1 [foo\\_s::foo\\_s](#) (float *f*, int *b*)

Definition at line 39 of file struct.cxx.

References [bar](#), and [foo](#).

##### 7.38.2.2 [foo\\_s::~~foo\\_s](#) ()

Definition at line 46 of file struct.cxx.

#### 7.38.3 Member Function Documentation

##### 7.38.3.1 [int foo\\_s::get\\_bar](#) () [inline]

Definition at line 11 of file struct.cxx.

References [bar](#).

##### 7.38.3.2 [float foo\\_s::get\\_foo](#) () [inline]

Definition at line 18 of file struct.cxx.

References [foo](#).

**7.38.3.3 void foo\_s::set\_bar (int *b*)** [inline]

Definition at line 25 of file struct.cxx.

References bar.

**7.38.3.4 void foo\_s::set\_foo (float *f*)** [inline]

Definition at line 32 of file struct.cxx.

References foo.

**7.38.4 Field Documentation****7.38.4.1 float foo\_s::foo**

Definition at line 3 of file struct.cxx.

Referenced by foo\_s(), get\_foo(), and set\_foo().

**7.38.4.2 int foo\_s::bar**

Definition at line 4 of file struct.cxx.

Referenced by foo\_s(), get\_bar(), and set\_bar().

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

- /home/dko/projects/mobilec/trunk/src/mxml-2.2.2/test/[struct.cxx](#)

## 7.39 havege\_state Struct Reference

HAVEGE state structure.

```
#include <havege.h>
```

### Data Fields

- [int PT1](#)
- [int PT2](#)
- [int offset](#) [2]
- [int pool](#) [COLLECT\_SIZE]
- [int WALK](#) [8192]

### 7.39.1 Detailed Description

HAVEGE state structure.

Definition at line 19 of file havege.h.

### 7.39.2 Field Documentation

#### 7.39.2.1 int havege\_state::PT1

Definition at line 21 of file havege.h.

Referenced by `havege_fill()`.

#### 7.39.2.2 int havege\_state::PT2

Definition at line 21 of file havege.h.

Referenced by `havege_fill()`.

#### 7.39.2.3 int havege\_state::offset[2]

Definition at line 21 of file havege.h.

Referenced by `havege_fill()`, and `havege_rand()`.

#### 7.39.2.4 int havege\_state::pool[COLLECT\_SIZE]

Definition at line 22 of file havege.h.

Referenced by `havege_rand()`.

#### 7.39.2.5 int havege\_state::WALK[8192]

Definition at line 23 of file havege.h.

Referenced by `havege_fill()`.

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

- </home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/havege.h>



## 7.40 host\_id\_s Struct Reference

```
#include <host_id.h>
```

### Data Fields

- char \* [hostname](#)
- [int](#) port

### 7.40.1 Detailed Description

Definition at line 38 of file `host_id.h`.

### 7.40.2 Field Documentation

#### 7.40.2.1 char\* host\_id\_s::hostname

Definition at line 40 of file `host_id.h`.

#### 7.40.2.2 int host\_id\_s::port

Definition at line 41 of file `host_id.h`.

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

- `/home/dko/projects/mobilec/trunk/src/include/host\_id.h`

## 7.41 hr\_time Struct Reference

timer structure

```
#include <timing.h>
```

### Data Fields

- unsigned char [opaque](#) [32]

### 7.41.1 Detailed Description

timer structure

Definition at line 19 of file timing.h.

### 7.41.2 Field Documentation

#### 7.41.2.1 unsigned char hr\_time::opaque[32]

Definition at line 21 of file timing.h.

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

- [/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/timing.h](#)

## 7.42 interpreter\_variable\_data\_s Struct Reference

```
#include <interpreter_variable_data.h>
```

### Data Fields

- char \* [name](#)
- [int](#) [size](#)
- ChType\_t [data\\_type](#)
- [int](#) [array\\_dim](#)
- [int](#) \* [array\\_extent](#)
- void \* [data](#)

### 7.42.1 Detailed Description

Definition at line 41 of file `interpreter_variable_data.h`.

### 7.42.2 Field Documentation

#### 7.42.2.1 char\* interpreter\_variable\_data\_s::name

Definition at line 42 of file `interpreter_variable_data.h`.

Referenced by `agent_AddPersistentVariable()`, `agent_xml_compose__data()`, `agent_xml_parse__data()`, `interpreter_variable_data_Copy()`, `interpreter_variable_data_Destroy()`, `interpreter_variable_data_Initialize()`, `interpreter_variable_data_InitializeFromAgent()`, and `interpreter_variable_data_New()`.

#### 7.42.2.2 int interpreter\_variable\_data\_s::size

Definition at line 43 of file `interpreter_variable_data.h`.

Referenced by `agent_AddPersistentVariable()`, `interpreter_variable_data_Copy()`, `interpreter_variable_data_Initialize()`, `interpreter_variable_data_InitializeFromAgent()`, and `interpreter_variable_data_New()`.

#### 7.42.2.3 ChType\_t interpreter\_variable\_data\_s::data\_type

Definition at line 44 of file `interpreter_variable_data.h`.

Referenced by `agent_AddPersistentVariable()`, `agent_xml_compose__data()`, `agent_xml_compose__row()`, `agent_xml_parse__data()`, `agent_xml_parse__row()`, `interpreter_variable_data_Copy()`, `interpreter_variable_data_Initialize()`, `interpreter_variable_data_InitializeFromAgent()`, `interpreter_variable_data_New()`, and `MC_GetAgentReturnData()`.

#### 7.42.2.4 int interpreter\_variable\_data\_s::array\_dim

Definition at line 45 of file `interpreter_variable_data.h`.

Referenced by `agent_AddPersistentVariable()`, `agent_xml_compose__data()`, `agent_xml_compose__row()`, `agent_xml_parse__data()`, `agent_xml_parse__row()`, `interpreter_variable_data_Copy()`, `interpreter_variable_data_Initialize()`, `interpreter_variable_data_InitializeFromAgent()`, `interpreter_variable_data_New()`, and `MC_GetAgentReturnData()`.

#### 7.42.2.5 `int* interpreter_variable_data_s::array_extent`

Definition at line 46 of file `interpreter_variable_data.h`.

Referenced by `agent_AddPersistentVariable()`, `agent_xml_compose__row()`, `agent_xml_parse__row()`, `interpreter_variable_data_Copy()`, `interpreter_variable_data_Destroy()`, `interpreter_variable_data_Initialize()`, `interpreter_variable_data_InitializeFromAgent()`, `interpreter_variable_data_New()`, and `MC_GetAgentReturnData()`.

#### 7.42.2.6 `void* interpreter_variable_data_s::data`

Definition at line 47 of file `interpreter_variable_data.h`.

Referenced by `agent_AddPersistentVariable()`, `agent_xml_compose__data()`, `agent_xml_compose__row()`, `agent_xml_parse__data()`, `agent_xml_parse__row()`, `interpreter_variable_data_Copy()`, `interpreter_variable_data_Destroy()`, `interpreter_variable_data_Initialize()`, `interpreter_variable_data_InitializeFromAgent()`, `interpreter_variable_data_New()`, and `MC_AgentVariableRetrieve()`.

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

- `/home/dko/projects/mobilec/trunk/src/include/interpreter_variable_data.h`

## 7.43 list\_s Struct Reference

```
#include <list.h>
```

### Data Fields

- [listNode\\_p listhead](#)
- [int size](#)

#### 7.43.1 Detailed Description

Definition at line 41 of file list.h.

#### 7.43.2 Field Documentation

##### 7.43.2.1 listNode\_p list\_s::listhead

Definition at line 43 of file list.h.

Referenced by `barrier_queue_Add()`, `barrier_queue_Get()`, `ListAdd()`, `ListDelete()`, `ListGetHead()`, `ListInitialize()`, `ListPop()`, `ListSearch()`, `ListTerminate()`, `syncListAddNode()`, and `syncListFind()`.

##### 7.43.2.2 int list\_s::size

Definition at line 44 of file list.h.

Referenced by `barrier_queue_Delete()`, `list_pGetSize()`, `ListAdd()`, `ListDelete()`, `ListInitialize()`, `ListPop()`, `ListSearch()`, `ListTerminate()`, `syncListDelete()`, and `syncListRemove()`.

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

- `/home/dko/projects/mobilec/trunk/src/mc_list/list.h`

## 7.44 listNode\_s Struct Reference

```
#include <list.h>
```

### Data Fields

- DATA [node\\_data](#)
- struct [listNode\\_s](#) \* [next](#)

### 7.44.1 Detailed Description

Definition at line 32 of file [list.h](#).

### 7.44.2 Field Documentation

#### 7.44.2.1 DATA listNode\_s::node\_data

Definition at line 34 of file [list.h](#).

Referenced by [AP\\_QUEUE\\_STD\\_DEFN\\_TEMPLATE\(\)](#), [barrier\\_queue\\_Add\(\)](#), [barrier\\_queue\\_Get\(\)](#), [df\\_SearchForService\(\)](#), [ListAdd\(\)](#), [ListDelete\(\)](#), [ListGetHead\(\)](#), [ListPop\(\)](#), [ListSearch\(\)](#), [syncListAddNode\(\)](#), and [syncListFind\(\)](#).

#### 7.44.2.2 struct listNode\_s\* listNode\_s::next [read]

Definition at line 35 of file [list.h](#).

Referenced by [AP\\_QUEUE\\_STD\\_DEFN\\_TEMPLATE\(\)](#), [barrier\\_queue\\_Add\(\)](#), [barrier\\_queue\\_Get\(\)](#), [df\\_SearchForService\(\)](#), [ListAdd\(\)](#), [ListDelete\(\)](#), [ListPop\(\)](#), [ListSearch\(\)](#), [syncListAddNode\(\)](#), and [syncListFind\(\)](#).

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

- [/home/dko/projects/mobilec/trunk/src/mc\\_list/list.h](#)

## 7.45 mc\_platform\_s Struct Reference

```
#include <mc_platform.h>
```

### Data Fields

- [int](#) `err`
- `char *` [hostname](#)
- `unsigned short` [port](#)
- `message_queue_p` [asm\\_message\\_queue](#)
- `message_queue_p` [message\\_queue](#)
- `agent_queue_p` [agent\\_queue](#)
- `connection_queue_p` [connection\\_queue](#)
- `df_p` [df](#)
- `ams_p` [ams](#)
- `acc_p` [acc](#)
- `cmd_prompt_p` [cmd\\_prompt](#)
- `syncList_p` [syncList](#)
- `barrier_queue_p` [barrier\\_queue](#)
- `listen_thread_arg_p` [listen\\_thread\\_arg](#)
- `listen_thread_arg_p` [client\\_thread\\_arg](#)
- `int` [default\\_agentstatus](#)
- `int` [stack\\_size](#) [MC\_THREAD\_ALL]
- `ChOptions_t *` [interp\\_options](#)
- `COND_T *` [MC\\_signal\\_cond](#)
- `COND_T *` [MC\\_sync\\_cond](#)
- `MUTEX_T *` [MC\\_signal\\_lock](#)
- `MUTEX_T *` [MC\\_sync\\_lock](#)
- `enum` [MC\\_Signal\\_e](#) [MC\\_signal](#)
- `enum` [MC\\_SteerCommand\\_e](#) [MC\\_steer\\_command](#)
- `MUTEX_T *` [MC\\_steer\\_lock](#)
- `COND_T *` [MC\\_steer\\_cond](#)
- `int` [giant](#)
- `MUTEX_T *` [giant\\_lock](#)
- `COND_T *` [giant\\_cond](#)
- `int` [quit](#)
- `MUTEX_T *` [quit\\_lock](#)

### 7.45.1 Detailed Description

Definition at line 49 of file `mc_platform.h`.

### 7.45.2 Field Documentation

#### 7.45.2.1 `int mc_platform_s::err`

Definition at line 53 of file `mc_platform.h`.

Referenced by `agent_Initialize()`, `mc_platform_Initialize()`, `message_InitializeFromAgent()`, and `message_InitializeFromString()`.

#### 7.45.2.2 `char* mc_platform_s::hostname`

Definition at line 58 of file `mc_platform.h`.

Referenced by `fipa_envelope_Compose__from()`, `mc_platform_Initialize()`, `message_InitializeFromAgent()`, `message_InitializeFromString()`, and `message_queue_SendOutgoing()`.

#### 7.45.2.3 `unsigned short mc_platform_s::port`

Definition at line 59 of file `mc_platform.h`.

Referenced by `fipa_envelope_Compose__from()`, `listen_Thread()`, `mc_platform_Initialize()`, `message_InitializeFromAgent()`, `message_InitializeFromString()`, and `message_queue_SendOutgoing()`.

#### 7.45.2.4 `message_queue_p mc_platform_s::asm_message_queue`

Definition at line 62 of file `mc_platform.h`.

Referenced by `acc_MessageHandlerThread()`, `acc_Thread()`, `mc_platform_Destroy()`, and `mc_platform_Initialize()`.

#### 7.45.2.5 `message_queue_p mc_platform_s::message_queue`

Definition at line 66 of file `mc_platform.h`.

Referenced by `acc_MessageHandlerThread()`, `acc_Thread()`, `MC_LoadAgentFromFile()`, `mc_platform_Destroy()`, `mc_platform_Initialize()`, `MC_SendAgentMigrationMessage()`, and `MC_SendAgentMigrationMessageFile()`.

#### 7.45.2.6 `agent_queue_p mc_platform_s::agent_queue`

Definition at line 67 of file `mc_platform.h`.

Referenced by `acc_MessageHandlerThread()`, `acc_Thread()`, `MC_AddAgent()`, `MC_FindAgentByID()`, `MC_FindAgentByName()`, `MC_GetAllAgents()`, `mc_platform_Destroy()`, `mc_platform_Initialize()`, `MC_RetrieveAgent()`, `MC_WaitAgent()`, and `MC_WaitRetrieveAgent()`.

#### 7.45.2.7 `connection_queue_p mc_platform_s::connection_queue`

Definition at line 68 of file `mc_platform.h`.

Referenced by `acc_Thread()`, `listen_Thread()`, `MC_End()`, `mc_platform_Destroy()`, and `mc_platform_Initialize()`.

#### 7.45.2.8 `df_p mc_platform_s::df`

Definition at line 70 of file `mc_platform.h`.

Referenced by `df_ProcessRequest()`, `df_Start()`, `df_Thread()`, `MC_DeregisterService()`, `MC_End()`, `mc_platform_Destroy()`, `mc_platform_Initialize()`, `MC_RegisterService()`, `MC_SearchForService()`, and `request_handler_REGISTER()`.



#### 7.45.2.9 ams\_p mc\_platform\_s::ams

Definition at line 71 of file mc\_platform.h.

Referenced by acc\_MessageHandlerThread(), agent\_RunChScriptThread(), ams\_Start(), ams\_Thread(), MC\_AddAgent(), MC\_End(), MC\_MainLoop(), mc\_platform\_Destroy(), mc\_platform\_Initialize(), and MC\_SetAgentStatus().

#### 7.45.2.10 acc\_p mc\_platform\_s::acc

Definition at line 72 of file mc\_platform.h.

Referenced by acc\_Start(), listen\_Thread(), MC\_End(), mc\_platform\_Destroy(), and mc\_platform\_Initialize().

#### 7.45.2.11 cmd\_prompt\_p mc\_platform\_s::cmd\_prompt

Definition at line 73 of file mc\_platform.h.

Referenced by cmd\_prompt\_Start(), MC\_End(), mc\_platform\_Destroy(), and mc\_platform\_Initialize().

#### 7.45.2.12 syncList\_p mc\_platform\_s::syncList

Definition at line 78 of file mc\_platform.h.

Referenced by MC\_CondBroadcast(), MC\_CondReset(), MC\_CondSignal(), MC\_CondWait(), MC\_MutexLock(), MC\_MutexUnlock(), mc\_platform\_Initialize(), MC\_SemaphorePost(), MC\_SemaphoreWait(), MC\_SyncDelete(), and MC\_SyncInit().

#### 7.45.2.13 barrier\_queue\_p mc\_platform\_s::barrier\_queue

Definition at line 79 of file mc\_platform.h.

Referenced by MC\_Barrier(), MC\_BarrierDelete(), MC\_BarrierInit(), mc\_platform\_Destroy(), and mc\_platform\_Initialize().

#### 7.45.2.14 listen\_thread\_arg\_p mc\_platform\_s::listen\_thread\_arg

Definition at line 81 of file mc\_platform.h.

#### 7.45.2.15 listen\_thread\_arg\_p mc\_platform\_s::client\_thread\_arg

Definition at line 82 of file mc\_platform.h.

#### 7.45.2.16 int mc\_platform\_s::default\_agentstatus

Definition at line 84 of file mc\_platform.h.

Referenced by agent\_Initialize(), mc\_platform\_Initialize(), and MC\_SetDefaultAgentStatus().

**7.45.2.17 int mc\_platform\_s::stack\_size[MC\_THREAD\_ALL]**

Definition at line 86 of file mc\_platform.h.

Referenced by acc\_Start(), agent\_RunChScript(), ams\_Start(), cmd\_prompt\_Start(), df\_Start(), and mc\_platform\_Initialize().

**7.45.2.18 ChOptions\_t\* mc\_platform\_s::interp\_options**

Definition at line 88 of file mc\_platform.h.

Referenced by agent\_RunChScriptThread(), MC\_ChInitializeOptions(), mc\_platform\_Destroy(), and mc\_platform\_Initialize().

**7.45.2.19 COND\_T\* mc\_platform\_s::MC\_signal\_cond**

Definition at line 91 of file mc\_platform.h.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), agent\_RunChScriptThread(), mc\_platform\_Destroy(), mc\_platform\_Initialize(), and MC\_WaitSignal().

**7.45.2.20 COND\_T\* mc\_platform\_s::MC\_sync\_cond**

Definition at line 92 of file mc\_platform.h.

Referenced by mc\_platform\_Destroy(), and mc\_platform\_Initialize().

**7.45.2.21 MUTEX\_T\* mc\_platform\_s::MC\_signal\_lock**

Definition at line 93 of file mc\_platform.h.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), agent\_RunChScriptThread(), mc\_platform\_Destroy(), mc\_platform\_Initialize(), and MC\_WaitSignal().

**7.45.2.22 MUTEX\_T\* mc\_platform\_s::MC\_sync\_lock**

Definition at line 94 of file mc\_platform.h.

Referenced by mc\_platform\_Destroy(), and mc\_platform\_Initialize().

**7.45.2.23 enum MC\_Signal\_e mc\_platform\_s::MC\_signal**

Definition at line 95 of file mc\_platform.h.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), agent\_RunChScriptThread(), mc\_platform\_Initialize(), MC\_ResetSignal(), and MC\_WaitSignal().

**7.45.2.24 enum MC\_SteerCommand\_e mc\_platform\_s::MC\_steer\_command**

Definition at line 98 of file mc\_platform.h.

Referenced by MC\_SendSteerCommand(), MC\_Steer(), and MC\_SteerControl().

**7.45.2.25 MUTEX\_T\* mc\_platform\_s::MC\_steer\_lock**

Definition at line 99 of file mc\_platform.h.

Referenced by mc\_platform\_Destroy(), mc\_platform\_Initialize(), MC\_SendSteerCommand(), MC\_Steer(), and MC\_SteerControl().

**7.45.2.26 COND\_T\* mc\_platform\_s::MC\_steer\_cond**

Definition at line 100 of file mc\_platform.h.

Referenced by mc\_platform\_Destroy(), mc\_platform\_Initialize(), MC\_SendSteerCommand(), and MC\_SteerControl().

**7.45.2.27 int mc\_platform\_s::giant**

Definition at line 104 of file mc\_platform.h.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), MC\_GetAllAgents(), MC\_HaltAgency(), mc\_platform\_Initialize(), MC\_ResetSignal(), MC\_ResumeAgency(), and MC\_WaitSignal().

**7.45.2.28 MUTEX\_T\* mc\_platform\_s::giant\_lock**

Definition at line 105 of file mc\_platform.h.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), MC\_GetAllAgents(), MC\_HaltAgency(), mc\_platform\_Destroy(), mc\_platform\_Initialize(), MC\_ResetSignal(), MC\_ResumeAgency(), and MC\_WaitSignal().

**7.45.2.29 COND\_T\* mc\_platform\_s::giant\_cond**

Definition at line 106 of file mc\_platform.h.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), mc\_platform\_Destroy(), mc\_platform\_Initialize(), and MC\_ResetSignal().

**7.45.2.30 int mc\_platform\_s::quit**

Definition at line 108 of file mc\_platform.h.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), ams\_Thread(), df\_Thread(), MC\_End(), and mc\_platform\_Initialize().

**7.45.2.31 MUTEX\_T\* mc\_platform\_s::quit\_lock**

Definition at line 109 of file mc\_platform.h.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), ams\_Thread(), df\_Thread(), MC\_End(), mc\_platform\_Destroy(), and mc\_platform\_Initialize().

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

- /home/dko/projects/mobilec/trunk/src/include/[mc\\_platform.h](#)

## 7.46 mc\_rwlock\_s Struct Reference

```
#include <mc_rwlock.h>
```

### Data Fields

- [int num\\_readers](#)
- [int write\\_flag](#)
- [int write\\_request](#)
- `MUTEX_T * lock`
- `COND_T * cond`

### 7.46.1 Detailed Description

Definition at line 39 of file mc\_rwlock.h.

### 7.46.2 Field Documentation

#### 7.46.2.1 int mc\_rwlock\_s::num\_readers

Definition at line 40 of file mc\_rwlock.h.

Referenced by mc\_rwlock\_init(), mc\_rwlock\_rdlock(), mc\_rwlock\_rdunlock(), and mc\_rwlock\_wrlock().

#### 7.46.2.2 int mc\_rwlock\_s::write\_flag

Definition at line 41 of file mc\_rwlock.h.

Referenced by mc\_rwlock\_init(), mc\_rwlock\_rdlock(), mc\_rwlock\_wrlock(), and mc\_rwlock\_wrunlock().

#### 7.46.2.3 int mc\_rwlock\_s::write\_request

Definition at line 42 of file mc\_rwlock.h.

Referenced by mc\_rwlock\_init(), mc\_rwlock\_rdlock(), and mc\_rwlock\_wrlock().

#### 7.46.2.4 MUTEX\_T\* mc\_rwlock\_s::lock

Definition at line 43 of file mc\_rwlock.h.

Referenced by mc\_rwlock\_destroy(), mc\_rwlock\_init(), mc\_rwlock\_rdlock(), mc\_rwlock\_rdunlock(), mc\_rwlock\_wrlock(), and mc\_rwlock\_wrunlock().

#### 7.46.2.5 COND\_T\* mc\_rwlock\_s::cond

Definition at line 44 of file mc\_rwlock.h.

Referenced by mc\_rwlock\_destroy(), mc\_rwlock\_init(), mc\_rwlock\_rdlock(), mc\_rwlock\_rdunlock(), mc\_rwlock\_wrlock(), and mc\_rwlock\_wrunlock().

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

- /home/dko/projects/mobilec/trunk/src/include/[mc\\_rwlock.h](#)

## 7.47 LibMC::MCAclMessage Class Reference

Encapsulates ACL messages in the Mobile-C library.

### Public Types

- enum [MC\\_FipaPerformative\\_e](#)  
*Enum for describing the type of an ACL message.*

### Public Member Functions

- [MCAclMessage](#) ()  
*Default constructor.*
- void [New](#) ()  
*Creates a new, blank ACL message.*
- [MCAclMessage Reply](#) ([MCAclMessage](#) acl\_message)  
*Creates an ACL message that is a response to the argument.*
- int [SetPerformative](#) ([MC\\_FipaPerformative\\_e](#) performative)  
*Sets the performative field of the message.*
- int [SetSender](#) (String name, String address)  
*Sets the sender field of the message.*
- int [AddReceiver](#) (String name, String address)  
*Adds a receiver to the list of receivers.*
- int [AddReplyTo](#) (String name, String address)  
*Adds a "reply-to" field to the message.*
- int [SetContent](#) (String content)  
*Sets the content field of the message.*
- int [Destroy](#) ()  
*Destroys a message.*

### Properties

- internal IntPtr [AclMsg](#) [get, set]

### Private Member Functions

- internal [MCAclMessage](#) (IntPtr msg)

## Private Attributes

- IntPtr [aclmsg\\_p](#)

### 7.47.1 Detailed Description

Encapsulates ACL messages in the Mobile-C library.

This class contains a pointer to an ACL message in the Mobile-C library. Functions are provided to send the message, set its various fields, and destroy the message.

Definition at line 72 of file MCAclMessage.cs.

### 7.47.2 Member Enumeration Documentation

#### 7.47.2.1 enum LibMC::MCAclMessage::MC\_FipaPerformative\_e

Enum for describing the type of an ACL message.

##### Note:

This enum is pulled directly from the Mobile-C library.

##### Enumerator:

**FIPA\_ERROR** Fipa performative enum value  
**FIPA\_ZERO** Fipa performative enum value  
**FIPA\_ACCEPT\_PROPOSAL** Fipa performative enum value  
**FIPA\_AGREE** Fipa performative enum value  
**FIPA\_CANCEL** Fipa performative enum value  
**FIPA\_CALL\_FOR\_PROPOSAL** Fipa performative enum value  
**FIPA\_CONFIRM** Fipa performative enum value  
**FIPA\_DISCONFIRM** Fipa performative enum value  
**FIPA\_FAILURE** Fipa performative enum value  
**FIPA\_INFORM** Fipa performative enum value  
**FIPA\_INFORM\_IF** Fipa performative enum value  
**FIPA\_INFORM\_REF** Fipa performative enum value  
**FIPA\_NOT\_UNDERSTOOD** Fipa performative enum value  
**FIPA\_PROPOGATE** Fipa performative enum value  
**FIPA\_PROPOSE** Fipa performative enum value  
**FIPA\_PROXY** Fipa performative enum value  
**FIPA\_QUERY\_IF** Fipa performative enum value  
**FIPA\_QUERY\_REF** Fipa performative enum value  
**FIPA\_REFUSE** Fipa performative enum value  
**FIPA\_REJECT\_PROPOSAL** Fipa performative enum value  
**FIPA\_REQUEST** Fipa performative enum value  
**FIPA\_REQUEST\_WHEN** Fipa performative enum value  
**FIPA\_REQUEST\_WHENEVER** Fipa performative enum value  
**FIPA\_SUBSCRIBE** Fipa performative enum value

Definition at line 79 of file MCAclMessage.cs.

### 7.47.3 Constructor & Destructor Documentation

#### 7.47.3.1 LibMC::MCAclMessage::MCAclMessage () [inline]

Default constructor.

Creates an empty ACL message object.

Definition at line 114 of file MCAclMessage.cs.

References `aclmsg_p`.

Referenced by `Reply()`.

#### 7.47.3.2 internal LibMC::MCAclMessage::MCAclMessage (IntPtr msg) [inline, private]

Definition at line 120 of file MCAclMessage.cs.

References `AclMsg`.

### 7.47.4 Member Function Documentation

#### 7.47.4.1 void LibMC::MCAclMessage::New () [inline]

Creates a new, blank ACL message.

Creates a new ACL message. The message is blank but valid.

Definition at line 153 of file MCAclMessage.cs.

References `AclMsg`.

#### 7.47.4.2 MCAclMessage LibMC::MCAclMessage::Reply (MCAclMessage acl\_message) [inline]

Creates an ACL message that is a response to the argument.

Creates an ACL message to respond to the argument.

##### Parameters:

*acl\_message* The message from which to create the reply.

##### Returns:

A new ACL message that is a response to the argument or an empty message if there is an error.

Definition at line 167 of file MCAclMessage.cs.

References `AclMsg`, and `MCAclMessage()`.

#### 7.47.4.3 int LibMC::MCAclMessage::SetPerformative (MC\_FipaPerformative\_e performative) [inline]

Sets the performative field of the message.

Sets the performative field of the message.



**Parameters:**

*performative* The fipa\_performative\_e enum describing the message.

**Returns:**

The return value of the underlying MC\_AclSetPerformative function.

**Note:**

The message must be a valid message or this function will fail.

Definition at line 186 of file MCAclMessage.cs.

References AclMsg.

**7.47.4.4 int LibMC::MCAclMessage::SetSender (String *name*, String *address*) [inline]**

Sets the sender field of the message.

Sets the performative field of the message.

**Parameters:**

*name* The name of the sending entity.

*address* The address of the sending entity.

**Returns:**

The return value of the underlying MC\_AclSetSender function.

**Note:**

The message must be a valid message or this function will fail.

Definition at line 202 of file MCAclMessage.cs.

References AclMsg.

**7.47.4.5 int LibMC::MCAclMessage::AddReceiver (String *name*, String *address*) [inline]**

Adds a receiver to the list of receivers.

Adds a receiver to the list of receivers for the message.

**Parameters:**

*name* The name of the receiver.

*address* The address of the receiver.

**Returns:**

The return value of the underlying MC\_AclAddReceiver function.

**Note:**

The message must be a valid message or this function will fail.

Definition at line 218 of file MCAclMessage.cs.

References AclMsg.

**7.47.4.6 int LibMC::MCAclMessage::AddReplyTo (String *name*, String *address*) [inline]**

Adds a "reply-to" field to the message.

Adds a "reply-to" field to the message. The reply-to field overrides the sender field when creating a reply.

**Parameters:**

*name* The name of the receiver.

*address* The address of the receiver.

**Returns:**

The return value of the underlying MC\_AclAddAddReplyTo function.

**Note:**

The message must be a valid message or this function will fail.

Definition at line 235 of file MCAclMessage.cs.

References AclMsg.

**7.47.4.7 int LibMC::MCAclMessage::SetContent (String *content*) [inline]**

Sets the content field of the message.

Sets the content field of the message.

**Parameters:**

*content* The string to copy to the content field.

**Returns:**

The return value of the underlying MC\_AclSetContent function.

**Note:**

The message must be a valid message or this function will fail.

Definition at line 250 of file MCAclMessage.cs.

References AclMsg.

**7.47.4.8 int LibMC::MCAclMessage::Destroy () [inline]**

Destroys a message.

This function destroys a message in the Mobile-C library. It releases the underlying memory and must be called when the message is no longer needed.

**Returns:**

The return value of the underlying MC\_AclDestroy function.

**Note:**

The message must be a valid message or this function will fail. In addition, messages are not automatically destroyed by the garbage collector. Use care when creating messages and ensure they are properly destroyed.

Definition at line 269 of file MCAclMessage.cs.

References AclMsg, and aclmsg\_p.

### 7.47.5 Field Documentation

#### 7.47.5.1 IntPtr LibMC::MCAclMessage::aclmsg\_p [private]

Definition at line 105 of file MCAclMessage.cs.

Referenced by Destroy(), and MCAclMessage().

### 7.47.6 Property Documentation

#### 7.47.6.1 internal IntPtr LibMC::MCAclMessage::AclMsg [get, set, private]

Definition at line 129 of file MCAclMessage.cs.

Referenced by LibMC::MCAgent::AclPost(), LibMC::MCAgency::AclSend(), AddReceiver(), AddReplyTo(), Destroy(), MCAclMessage(), New(), Reply(), SetContent(), SetPerformative(), and SetSender().

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

- [/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAclMessage.cs](#)

## 7.48 LibMC::MCAgency Class Reference

Wrapper class for [MCAgency\\_t](#) structure.

### Public Types

- enum [MCAgencyState](#)  
*Enum for describing the state of the agency.*
- enum [ChShellType](#)  
*Ch shell type.*
- enum [MC\\_ThreadIndex\\_e](#)  
*Enum for describing the different threads that Mobile-C uses.*
- enum [MC\\_SteerCommand\\_e](#)  
*Available commands for MC\_Steer.*

### Public Member Functions

- [MCAgency](#) ()  
*Default constructor.*
- [int Initialize](#) ()  
*Starts the agency.*
- [int End](#) ()  
*Stops and destroys the agency.*
- [int ChInitializeOptions](#) ([ChShellType](#) shellType, String home)  
*Initializes Ch options for the agency.*
- [int SetThreadsAllOn](#) ()
- [int SetThreadsAllOff](#) ()  
*Sets all threads for the agency to "off".*
- [int SetThreadOn](#) ([MC\\_ThreadIndex\\_e](#) index)  
*Sets an individual thread for the agency to "on".*
- [int SetThreadOff](#) ([MC\\_ThreadIndex\\_e](#) index)  
*Sets an individual thread for the agency to "off".*
- [int HaltAgency](#) ()  
*Temporarily halts the agency.*
- [int ResumeAgency](#) ()  
*Resumes a halted agency.*

- [int SetDefaultAgentStatus](#) ([MCAgent.MC\\_AgentStatus\\_e](#) status)  
*Sets the default state of an agent in the agency.*
- [MCAgent WaitRetrieveAgent](#) ()  
*Waits for an agent to arrive and returns the agent.*
- [int WaitAgent](#) ()  
*Waits for an agent to arrive.*
- [int SendAgentMigrationMessageFile](#) (String filename, String hostname, [int](#) port)  
*Sends an agent migration message file to an agency.*
- [int LoadAgentMigrationMessageFile](#) (String filename)  
*Load an agent migration message.*
- [int SendAgentMigrationMessage](#) (String message, String hostname, [int](#) port)  
*Sends an agent migration message to an agency.*
- [int CondBroadcast](#) ([int](#) id)  
*Broadcast a condition signal.*
- [int CondSignal](#) ([int](#) id)  
*Signal a condition.*
- [int CondReset](#) ([int](#) id)  
*Reset a condition signal.*
- [int CondWait](#) ([int](#) id)  
*Wait for a condition signal.*
- [int MutexLock](#) ([int](#) id)  
*Lock a mutex.*
- [int MutexUnlock](#) ([int](#) id)  
*Unlock a mutex.*
- [int SemaphorePost](#) ([int](#) id)  
*Posts a semaphore.*
- [int SemaphoreWait](#) ([int](#) id)  
*Wait for a semaphore to be posted.*
- [int ResetSignal](#) ()  
*Reset an agency signal.*
- [int SyncDelete](#) ([int](#) id)  
*Delete a synchronization variable.*
- [int SyncInit](#) ([int](#) id)  
*Create a new synchronization variable.*

- [int WaitSignal](#) ([int](#) signals)  
*Wait for agency signals.*
- [int BarrierDelete](#) ([int](#) id)  
*Delete a barrier object.*
- [int BarrierInit](#) ([int](#) id, [int](#) num\_procs)  
*Create a new barrier.*
- [MC\\_SteerCommand\\_e SteerControl](#) ()  
*Steering control function.*
- [int Steer](#) ([IntPtr](#) funcptr, [IntPtr](#) arg)  
*Steering control function.*
- [int RegisterService](#) ([MCAgent](#) agent, [int](#) agentID, [String](#) agentName, [String\[\]](#) serviceNames, [int](#) numServices)  
*Registers services in the agency.*
- [int SearchForService](#) ([String](#) searchString, [IntPtr](#) agentNames, [IntPtr](#) serviceNames, [IntPtr](#) agentIDs, [IntPtr](#) numResults)  
*Searches for services in the agency.*
- [int AddAgent](#) ([MCAgent](#) agent)  
*Add an agent to the agency.*
- [MCAgent FindAgentByName](#) ([String](#) name)  
*Finds an agent by its name.*
- [MCAgent FindAgentByID](#) ([int](#) id)  
*Find an agent by its ID.*
- [MCAgent RetrieveAgent](#) ()  
*Retrieve an agent from the agency.*
- [int AclSend](#) ([MCAclMessage](#) acl\_message)  
*Send an ACL message to the agency.*
- [int MainLoop](#) ()  
*Makes the agency wait indefinitely.*

## Properties

- [IntPtr Agency](#) [get, set]
- [int Port](#) [get, set]  
*Accessor for the port number of the agency.*
- [MCAgencyState State](#) [get]  
*Accessor for the agency state.*

## Private Types

- enum [MC\\_Signal\\_e](#)  
*MobileC system signals.*

## Private Member Functions

- static [IntPtr \\_MC\\_Initialize](#) (int port, ref [MCAgencyOptions\\_t](#) options)
- static [int \\_MC\\_End](#) (IntPtr agency)
- static [int \\_MC\\_ChInitializeOptions](#) (IntPtr agency, [ChOptions\\_t](#) options)
- static [int \\_MC\\_InitializeAgencyOptions](#) (ref [MCAgencyOptions\\_t](#) options)
- static [int \\_MC\\_SetThreadsAllOn](#) (ref [MCAgencyOptions\\_t](#) options)
- static [int \\_MC\\_SetThreadsAllOff](#) (ref [MCAgencyOptions\\_t](#) options)
- static [int \\_MC\\_SetThreadOn](#) (ref [MCAgencyOptions\\_t](#) options, [MC\\_ThreadIndex\\_e](#) index)
- static [int \\_MC\\_SetThreadOff](#) (ref [MCAgencyOptions\\_t](#) options, [MC\\_ThreadIndex\\_e](#) index)
- static [int \\_MC\\_HaltAgency](#) (IntPtr agency)
- static [int \\_MC\\_ResumeAgency](#) (IntPtr agency)
- static [int \\_MC\\_SetDefaultAgentStatus](#) (IntPtr agency, [MCAgent.MC\\_AgentStatus\\_e](#) status)
- static [IntPtr \\_MC\\_WaitRetrieveAgent](#) (IntPtr agency)
- static [int \\_MC\\_WaitAgent](#) (IntPtr agency)
- static [int \\_MC\\_MainLoop](#) (IntPtr agency)
- static [int \\_MC\\_SendAgentMigrationMessageFile](#) (IntPtr agency, String filename, String hostname, int port)
- static [int \\_MC\\_SendAgentMigrationMessage](#) (IntPtr agency, String message, String hostname, int port)
- static [int \\_MC\\_CondBroadcast](#) (IntPtr agency, int id)
- static [int \\_MC\\_CondSignal](#) (IntPtr agency, int id)
- static [int \\_MC\\_CondReset](#) (IntPtr agency, int id)
- static [int \\_MC\\_CondWait](#) (IntPtr agency, int id)
- static [int \\_MC\\_MutexLock](#) (IntPtr agency, int id)
- static [int \\_MC\\_MutexUnlock](#) (IntPtr agency, int id)
- static [int \\_MC\\_SemaphorePost](#) (IntPtr agency, int id)
- static [int \\_MC\\_SemaphoreWait](#) (IntPtr agency, int id)
- static [int \\_MC\\_ResetSignal](#) (IntPtr agency)
- static [int \\_MC\\_SyncDelete](#) (IntPtr agency, int id)
- static [int \\_MC\\_SyncInit](#) (IntPtr agency, int id)
- static [int \\_MC\\_WaitSignal](#) (IntPtr agency, int signals)
- static [int \\_MC\\_BarrierDelete](#) (IntPtr agency, int id)
- static [int \\_MC\\_BarrierInit](#) (IntPtr agency, int id, int num\_procs)
- static [MC\\_SteerCommand\\_e \\_MC\\_SteerControl](#) ()
- static [int \\_MC\\_Steer](#) (IntPtr agency, IntPtr funcptr, IntPtr arg)
- static [int \\_MC\\_RegisterService](#) (IntPtr agency, IntPtr agent, int agentID, String agentName, String[ ] serviceNames, int numServices)
- static [int \\_MC\\_SearchForService](#) (IntPtr agency, String searchString, IntPtr agentNames, IntPtr serviceNames, IntPtr agentIDs, IntPtr numResults)
- static [int \\_MC\\_AddAgent](#) (IntPtr agency, IntPtr agent)
- static internal [int \\_MC\\_DeleteAgent](#) (IntPtr agent)
- static [IntPtr \\_MC\\_FindAgentByName](#) (IntPtr agency, String name)
- static [IntPtr \\_MC\\_FindAgentByID](#) (IntPtr agency, int ID)

- static IntPtr [\\_MC\\_RetrieveAgent](#) (IntPtr agency)
- static internal [int \\_MC\\_GetAgentID](#) (IntPtr agent)
- static internal [String \\_MC\\_GetAgentName](#) (IntPtr agent)
- static internal [int \\_MC\\_GetAgentNumTasks](#) (IntPtr agent)
- static internal [MCAgent.MC\\_AgentStatus\\_e \\_MC\\_GetAgentStatus](#) (IntPtr agent)
- static internal [MCAgent.MC\\_AgentType\\_e \\_MC\\_GetAgentType](#) (IntPtr agent)
- static internal [String \\_MC\\_GetAgentXMLString](#) (IntPtr agent)
- static internal [int \\_MC\\_PrintAgentCode](#) (IntPtr agent)
- static internal [String \\_MC\\_RetrieveAgentCode](#) (IntPtr agent)
- static internal [int \\_MC\\_SetAgentStatus](#) (IntPtr agent, [MCAgent.MC\\_AgentStatus\\_e](#) status)
- static internal [int \\_MC\\_TerminateAgent](#) (IntPtr agent)
- static internal [int \\_MC\\_CallAgentFunc](#) (IntPtr agent, String funcName, IntPtr returnVal, IntPtr varg)
- static internal IntPtr [\\_MC\\_GetAgentExecEngine](#) (IntPtr agent)
- static internal [int \\_MC\\_GetAgentReturnData](#) (IntPtr agent, [int](#) task\_num, IntPtr data, IntPtr dim, IntPtr extent)
- static internal IntPtr [\\_MC\\_AclNew](#) ()
- static internal [int \\_MC\\_AclPost](#) (IntPtr agent, IntPtr message)
- static internal IntPtr [\\_MC\\_AclReply](#) (IntPtr acl\_message)
- static internal IntPtr [\\_MC\\_AclRetrieve](#) (IntPtr agent)
- static internal [int \\_MC\\_AclSend](#) (IntPtr agency, IntPtr acl\_message)
- static internal IntPtr [\\_MC\\_AclWaitRetrieve](#) (IntPtr agent)
- static internal [int \\_MC\\_AclSetPerformative](#) (IntPtr acl, [MCAclMessage.MC\\_FipaPerformative\\_e](#) performative)
- static internal [int \\_MC\\_AclSetSender](#) (IntPtr acl, String name, String address)
- static internal [int \\_MC\\_AclAddReceiver](#) (IntPtr acl, String name, String address)
- static internal [int \\_MC\\_AclAddReplyTo](#) (IntPtr acl, String name, String address)
- static internal [int \\_MC\\_AclSetContent](#) (IntPtr acl, String content)
- static internal [int \\_MC\\_AclDestroy](#) (IntPtr acl)

## Private Attributes

- IntPtr [agency\\_p](#) = IntPtr.Zero
- [MCAgencyOptions\\_t](#) options
- [int](#) port = -1
- [MCAgencyState](#) state = MCAgencyState.NoState
- const String [mcDll](#)

## Data Structures

- struct [ChOptions\\_t](#)  
*ChOptions structures.*
- struct [MCAgency\\_t](#)
- struct [MCAgencyOptions\\_t](#)



### 7.48.1 Detailed Description

Wrapper class for [MCAgency\\_t](#) structure.

This class provides an interface to the Mobile-C agency. Member functions for the class are generally overloaded versions of the respective functions in the Mobile-C library. The class maintains a pointer to the Mobile-C agency in unmanaged memory. The pointer is not accessible by the user.

Definition at line 329 of file MCAgency.cs.

### 7.48.2 Member Enumeration Documentation

#### 7.48.2.1 enum LibMC::MCAgency::MCAgencyState

Enum for describing the state of the agency.

This enum is used to determine whether or not certain actions should be permitted, such as halting, resuming, and ending an agency

**Enumerator:**

- NoState* Default, uninitialized state
- Initialized* Agency initialized, but not started
- Running* Agency is running
- Halted* Agency has been stopped (can be resumed)
- Ended* Agency is stopped (destroyed)

Definition at line 337 of file MCAgency.cs.

#### 7.48.2.2 enum LibMC::MCAgency::ChShellType

Ch shell type.

Used to set the shell type for the Ch interpreter.

**Enumerator:**

- CH\_REGULARCH* Default, regular shell
- CH\_SAFECH* Safe shell

Definition at line 351 of file MCAgency.cs.

#### 7.48.2.3 enum LibMC::MCAgency::MC\_ThreadIndex\_e

Enum for describing the different threads that Mobile-C uses.

These enums can be used to turn threads on and off before an agency is initialized.

**Note:**

This enum is pulled directly from the Mobile-C library.

**Enumerator:**

- MC\_THREAD\_DF* Directory Facilitator

*MC\_THREAD\_AMS* Agent Managment system  
*MC\_THREAD\_ACC* Agency communications  
*MC\_THREAD\_CP* Command Prompt  
*MC\_THREAD\_AGENT* Agent threads  
*MC\_THREAD\_ALL*

Definition at line 79 of file MCEExports.cs.

#### 7.48.2.4 enum LibMC::MCAgency::MC\_SteerCommand\_e

Available commands for MC\_Steer.

**Note:**

This enum is pulled directly from the Mobile-C library.

**Enumerator:**

*MC\_RUN* Continue the algorithm  
*MC\_SUSPEND* Suspend/pause the algorithm  
*MC\_RESTART* Restart the algorithm from the beginning  
*MC\_STOP* Stop the algorithm

Definition at line 94 of file MCEExports.cs.

#### 7.48.2.5 enum LibMC::MCAgency::MC\_Signal\_e [private]

MobileC system signals.

Each signal is activated after the corresponding action. i.e. The 'MC\_RECV\_MESSAGE' signal is activated after a message is received.

**Note:**

This enum is pulled directly from the Mobile-C library.

**See also:**

[MC\\_WaitSignal\(\)](#), [MC\\_ResetSignal\(\)](#)

**Enumerator:**

*MC\_NO\_SIGNAL*  
*MC\_RECV\_CONNECTION*  
*MC\_RECV\_MESSAGE*  
*MC\_RECV\_AGENT*  
*MC\_RECV\_RETURN*  
*MC\_EXEC\_AGENT*  
*MC\_ALL\_SIGNALS*

Definition at line 111 of file MCEExports.cs.

### 7.48.3 Constructor & Destructor Documentation

#### 7.48.3.1 LibMC::MCAgency::MCAgency () [inline]

Default constructor.

The default constructor for the [MCAgency](#) class. It creates a new agency, default options for the agency, and initializes the agency. It does not start the agency.

Definition at line 369 of file MCAgency.cs.

References `_MC_InitializeAgencyOptions()`, `options`, and `state`.

### 7.48.4 Member Function Documentation

#### 7.48.4.1 int LibMC::MCAgency::Initialize () [inline]

Starts the agency.

Starts the agency and sets the agency state.

##### Returns:

0 on success, -1 on failure.

##### Note:

The agency port and any other options must be set before calling this function.

Definition at line 445 of file MCAgency.cs.

References `Agency`, `options`, `port`, and `state`.

#### 7.48.4.2 int LibMC::MCAgency::End () [inline]

Stops and destroys the agency.

Stops the agency and sets the agency state appropriately.

##### Returns:

The return value of the underlying `MC_End` function.

##### Note:

This call will fail if the underlying Mobile-C agency is not in the correct state.

Definition at line 467 of file MCAgency.cs.

References `Agency`, and `state`.

#### 7.48.4.3 int LibMC::MCAgency::ChInitializeOptions (ChShellType *shellType*, String *home*) [inline]

Initializes Ch options for the agency.

Can be used to set the home directory and shell mode for the Ch interpreter.

**Parameters:**

*shellType* The type of shell Ch should use: CH\_REGULARCH or CH\_SAFECH.

*home* The home directory Ch should use.

**Returns:**

The return value of the underlying MC\_ChInitializeOptions function.

**Note:**

This function must be called before the agency is started.

Definition at line 485 of file MCAgency.cs.

References \_MC\_ChInitializeOptions(), Agency, LibMC::MCAgency::ChOptions\_t::chhome, int, options, and LibMC::MCAgency::ChOptions\_t::shelltype.

**7.48.4.4 int LibMC::MCAgency::SetThreadsAllOn () [inline]**

Definition at line 502 of file MCAgency.cs.

References \_MC\_SetThreadsAllOn(), and options.

**7.48.4.5 int LibMC::MCAgency::SetThreadsAllOff () [inline]**

Sets all threads for the agency to "off".

Sets all threads for the agency to "off." Not recommended for use.

**Returns:**

The return value of the underlying MC\_SetThreadsAllOff function.

**Note:**

This function must be called before the agency is started.

Definition at line 516 of file MCAgency.cs.

References \_MC\_SetThreadsAllOff(), and options.

**7.48.4.6 int LibMC::MCAgency::SetThreadOn (MC\_ThreadIndex\_e index) [inline]**

Sets an individual thread for the agency to "on".

Threads are on by default. If they have been turned off, this function turns them on again.

**Parameters:**

*index* The enum that identifies the thread to be turned on.

**Returns:**

The return value of the underlying MC\_SetThreadOn function.

**Note:**

This function must be called before the agency is started.

Definition at line 532 of file MCAgency.cs.

References `_MC_SetThreadOn()`, and options.

**7.48.4.7 int LibMC::MCAgency::SetThreadOff (MC\_ThreadIndex\_e *index*) [inline]**

Sets an individual thread for the agency to "off".

Most commonly used to turn the command prompt thread off.

**Parameters:**

*index* The enum that identifies the thread to be turned off.

**Returns:**

The return value of the underlying `MC_SetThreadOff` function.

**Note:**

This function must be called before the agency is started.

Definition at line 547 of file MCAgency.cs.

References `_MC_SetThreadOff()`, and options.

**7.48.4.8 int LibMC::MCAgency::HaltAgency () [inline]**

Temporarily halts the agency.

Halts the agency until it is resumed or ended.

**Returns:**

The return value of the underlying `MC_HaltAgency` function.

**Note:**

The underlying Mobile-C agency must be in the correct state to call this function or it will fail.

Definition at line 562 of file MCAgency.cs.

References `_MC_HaltAgency()`, `Agency`, and state.

**7.48.4.9 int LibMC::MCAgency::ResumeAgency () [inline]**

Resumes a halted agency.

Resumes a halted agency. Cannot be used on ended agencies.

**Returns:**

The return value of the underlying `MC_ResumeAgency` function.

**Note:**

The underlying Mobile-C agency must be in the correct state to call this function or it will fail.

Definition at line 579 of file MCAgency.cs.

References `_MC_ResumeAgency()`, `Agency`, and `state`.

**7.48.4.10** `int LibMC::MCAgency::SetDefaultAgentStatus (MCAgent.MC_AgentStatus_e status)`  
[inline]

Sets the default state of an agent in the agency.

Can be used to set the default status of agents, but most agents managed their state on their own.

**Parameters:**

*status* The enum that identifies the desired agent state.

**Returns:**

The return value of the underlying `MC_SetDefaultAgentStatus` function.

Definition at line 595 of file MCAgency.cs.

References `_MC_SetDefaultAgentStatus()`, and `Agency`.

**7.48.4.11** `MCAgent LibMC::MCAgency::WaitRetrieveAgent ()` [inline]

Waits for an agent to arrive and returns the agent.

Waits for an agent to arrive in the agency, then returns that agent. The agent is not allowed to execute.

**Returns:**

The agent that was retrieved or an empty agent if it fails.

Definition at line 608 of file MCAgency.cs.

References `_MC_WaitRetrieveAgent()`, and `Agency`.

**7.48.4.12** `int LibMC::MCAgency::WaitAgent ()` [inline]

Waits for an agent to arrive.

Waits for an agent to arrive in the agency. The agent is allowed to execute normally.

**Returns:**

The return value of the underlying `MC_WaitAgent` function.

Definition at line 625 of file MCAgency.cs.

References `_MC_WaitAgent()`, and `Agency`.

**7.48.4.13 int LibMC::MCAgency::SendAgentMigrationMessageFile (String *filename*, String *hostname*, int *port*) [inline]**

Sends an agent migration message file to an agency.

Sends the specified XML file to another agency (local or remote).

**Parameters:**

*filename* The name of the file to send (fully qualified).

*hostname* The URL, IP address, or other identifier for the agency host.

*port* The port to send to.

**Returns:**

The return value of the underlying MC\_SendAgentMigrationMessageFile function.

Definition at line 644 of file MCAgency.cs.

References Agency.

**7.48.4.14 int LibMC::MCAgency::LoadAgentMigrationMessageFile (String *filename*) [inline]**

Load an agent migration message.

Loads the specified XML file to this agency automatically. There is no need to specify a port or agency location.

**Parameters:**

*filename* The name of the file to send (fully qualified).

**Returns:**

The return value of the underlying MC\_SendAgentMigrationMessageFile function.

Definition at line 658 of file MCAgency.cs.

References Agency, and Port.

**7.48.4.15 int LibMC::MCAgency::SendAgentMigrationMessage (String *message*, String *hostname*, int *port*) [inline]**

Sends an agent migration message to an agency.

Sends an agent migration message to another agency (local or remote).

**Parameters:**

*message* The agent migration message.

*hostname* The URL, IP address, or other identifier for the agency host.

*port* The port to send to.

**Returns:**

The return value of the underlying MC\_SendAgentMigrationMessageFile function.

Definition at line 673 of file MCAgency.cs.

References `_MC_SendAgentMigrationMessage()`, and `Agency`.

#### 7.48.4.16 `int LibMC::MCAgency::CondBroadcast (int id)` `[inline]`

Broadcast a condition signal.

Broadcasts a signal in the agency. The parameter "id" is the ID of the agency sync variable created with [SyncInit\(\)](#).

##### Parameters:

*id* The ID number of the condition to signal.

##### Returns:

The return value of the underlying `MC_CondBroadcast` function.

Definition at line 691 of file MCAgency.cs.

References `_MC_CondBroadcast()`, and `Agency`.

#### 7.48.4.17 `int LibMC::MCAgency::CondSignal (int id)` `[inline]`

Signal a condition.

Signals a condition in the agency. The parameter "id" is the ID of the agency sync variable to signal that was created with [SyncInit\(\)](#).

##### Parameters:

*id* The ID number of the condition to signal.

##### Returns:

The return value of the underlying `MC_CondSignal` function.

Definition at line 705 of file MCAgency.cs.

References `_MC_CondSignal()`, and `Agency`.

#### 7.48.4.18 `int LibMC::MCAgency::CondReset (int id)` `[inline]`

Reset a condition signal.

Resets a signal in the agency. The parameter "id" is the ID of the agency sync variable created with [SyncInit\(\)](#). This function must be called after a condition is received in order to clear it.

##### Parameters:

*id* The ID number of the condition to reset.

##### Returns:

The return value of the underlying `MC_CondReset` function.

Definition at line 720 of file MCAgency.cs.

References `_MC_CondReset()`, and `Agency`.



**7.48.4.19 int LibMC::MCAgency::CondWait (int *id*)** [inline]

Wait for a condition signal.

Waits for a condition signal in the agency. The parameter "id" is the ID of the agency sync variable created with [SyncInit\(\)](#). This function blocks until the signal is received.

**Parameters:**

*id* The ID number of the condition to wait for.

**Returns:**

The return value of the underlying MC\_CondWait function.

Definition at line 735 of file MCAgency.cs.

References `_MC_CondWait()`, and `Agency`.

**7.48.4.20 int LibMC::MCAgency::MutexLock (int *id*)** [inline]

Lock a mutex.

Locks a mutex in the agency. The parameter "id" is the ID of the agency sync variable created with [SyncInit\(\)](#). This function blocks until the mutex is locked.

**Parameters:**

*id* The ID number of the mutex to lock.

**Returns:**

The return value of the underlying MC\_MutexLock function.

Definition at line 750 of file MCAgency.cs.

References `_MC_MutexLock()`, and `Agency`.

**7.48.4.21 int LibMC::MCAgency::MutexUnlock (int *id*)** [inline]

Unlock a mutex.

Locks a mutex in the agency. The parameter "id" is the ID of the agency sync variable created with [SyncInit\(\)](#).

**Parameters:**

*id* The ID number of the mutex to unlock.

**Returns:**

The return value of the underlying MC\_MutexUnlock function.

Definition at line 764 of file MCAgency.cs.

References `_MC_MutexUnlock()`, and `Agency`.

**7.48.4.22** `int LibMC::MCAgency::SemaphorePost (int id)` `[inline]`

Posts a semaphore.

Posts a semaphore in the agency. The parameter "id" is the ID of the agency sync variable created with [SyncInit\(\)](#).

**Parameters:**

*id* The ID number of the semaphore to post.

**Returns:**

The return value of the underlying MC\_SemaphorePost function.

Definition at line 778 of file MCAgency.cs.

References `_MC_SemaphorePost()`, and `Agency`.

**7.48.4.23** `int LibMC::MCAgency::SemaphoreWait (int id)` `[inline]`

Wait for a semaphore to be posted.

Wait for a semaphore in the agency to be posted. The parameter "id" is the ID of the agency sync variable created with [SyncInit\(\)](#). This function blocks until the semaphore is posted.

**Parameters:**

*id* The ID number of the semaphore to wait for.

**Returns:**

The return value of the underlying MC\_SemaphoreWait function.

Definition at line 793 of file MCAgency.cs.

References `_MC_SemaphoreWait()`, and `Agency`.

**7.48.4.24** `int LibMC::MCAgency::ResetSignal ()` `[inline]`

Reset an agency signal.

Resets a signal in the agency. The parameter "id" is the ID of the agency sync variable created with [SyncInit\(\)](#).

**Returns:**

The return value of the underlying MC\_ResetSignal function.

Definition at line 806 of file MCAgency.cs.

References `_MC_ResetSignal()`, and `Agency`.

**7.48.4.25** `int LibMC::MCAgency::SyncDelete (int id)` `[inline]`

Delete a synchronization variable.

Deletes a synchronization variable in the agency. The parameter "id" is the ID of the agency sync variable created with [SyncInit\(\)](#).

**Parameters:**

*id* The ID number of the variable to delete.

**Returns:**

The return value of the underlying MC\_SyncDelete function.

Definition at line 820 of file MCAgency.cs.

References `_MC_SyncDelete()`, and `Agency`.

**7.48.4.26 int LibMC::MCAgency::SyncInit (int *id*) [inline]**

Create a new synchronization variable.

Creates a new synchronization variable in the agency. The parameter "id" is desired ID of the variable. A random ID is returned if "id" is already in use.

**Parameters:**

*id* The ID number of the condition to signal.

**Returns:**

The return value of the underlying MC\_CondBroadcast function- either a random ID or the desired ID if the desired ID is already in use.

Definition at line 836 of file MCAgency.cs.

References `_MC_SyncInit()`, and `Agency`.

**7.48.4.27 int LibMC::MCAgency::WaitSignal (int *signals*) [inline]**

Wait for agency signals.

Waits for signals to occur in the agency.

**Parameters:**

*signals* The ID number of the condition to signal.

**Returns:**

The return value of the underlying MC\_WaitSignal function.

Definition at line 849 of file MCAgency.cs.

References `_MC_WaitSignal()`, and `Agency`.

**7.48.4.28 int LibMC::MCAgency::BarrierDelete (int *id*) [inline]**

Delete a barrier object.

Deletes a barrier object from the agency. The parameter "id" is the ID of the agency sync variable created with [BarrierInit\(\)](#).

**Parameters:**

*id* The ID number of the barrier to delete.

**Returns:**

The return value of the underlying MC\_BarrierDelete function.

Definition at line 863 of file MCAgency.cs.

References \_MC\_BarrierDelete(), and Agency.

**7.48.4.29 int LibMC::MCAgency::BarrierInit (int *id*, int *num\_procs*) [inline]**

Create a new barrier.

Creates a new barrier object in the agency.

**Parameters:**

*id* The ID number of the condition to signal.

*num\_procs* the number of process to block (?)

**Returns:**

The return value of the underlying MC\_BarrierInit function.

Definition at line 877 of file MCAgency.cs.

References \_MC\_BarrierInit(), and Agency.

**7.48.4.30 MC\_SteerCommand\_e LibMC::MCAgency::SteerControl () [inline]**

Steering control function.

Really not sure.

**Returns:**

The return value of the underlying MC\_SteerControl function.

**Todo**

Test MC\_SteerControl, MC\_Steer.

Definition at line 895 of file MCAgency.cs.

References \_MC\_SteerControl().

**7.48.4.31 int LibMC::MCAgency::Steer (IntPtr *funcptr*, IntPtr *arg*) [inline]**

Steering control function.

Really not sure.

**Parameters:**

*funcptr* Pointer to the steering function

*arg* Argument to function

#### Returns:

The return value of the underlying `_MC_Steer` function.

#### Note:

This function does nothing but throw an exception right now.

#### Bug

`MC_Steer` is not yet implemented.

#### Todo

Implement `MC_Steer`

Definition at line 913 of file `MCAgency.cs`.

#### 7.48.4.32 `int LibMC::MCAgency::RegisterService (MCAgent agent, int agentID, String agentName, String[ ] serviceNames, int numServices)` [inline]

Registers services in the agency.

Registers services provided by agents with the agency. Not really useful in binary space.

#### Parameters:

*agent* The agent providing the services.

*agentID* The agent ID number.

*agentName* The agent name.

*serviceNames* An array of service names.

*numServices* The number of services provided.

#### Returns:

The return value of the underlying `MC_RegisterService` function.

#### Todo

Test `MC_RegisterService` and `MC_SearchForService`.

Definition at line 938 of file `MCAgency.cs`.

References `_MC_RegisterService()`, `Agency`, and `LibMC::MCAgent::Agent`.

#### 7.48.4.33 `int LibMC::MCAgency::SearchForService (String searchString, IntPtr agentNames, IntPtr serviceNames, IntPtr agentIDs, IntPtr numResults)` [inline]

Searches for services in the agency.

Searches for services provided by agents with the agency. Not really useful in binary space.

**Parameters:**

*searchString* The agent providing the services.  
*agentNames* The agent ID number.  
*serviceNames* The agent name.  
*agentIDs* An array of service names.  
*numResults* The number of services provided.

**Returns:**

The return value of the underlying MC\_SearchForService function.

**Note:**

This function does nothing but throw an exception right now.

**Bug**

MC\_SearchForService is not yet implemented.

**Todo**

Implement SearchForService

Definition at line 961 of file MCAgency.cs.

**7.48.4.34 int LibMC::MCAgency::AddAgent (MCAgent *agent*) [inline]**

Add an agent to the agency.

Adds an agent to the agency.

**Parameters:**

*agent* The agent to add.

**Returns:**

The return value of the underlying MC\_AddAgent function.

Definition at line 979 of file MCAgency.cs.

References \_MC\_AddAgent(), Agency, and LibMC::MCAgent::Agent.

**7.48.4.35 MCAgent LibMC::MCAgency::FindAgentByName (String *name*) [inline]**

Finds an agent by its name.

Finds an agent in the agency by its name.

**Parameters:**

*name* The name of the agent to search for.

**Returns:**

The return value of the underlying MC\_FindAgentByName function.

Definition at line 992 of file MCAgency.cs.

References \_MC\_FindAgentByName(), and Agency.

**7.48.4.36 MCAgent LibMC::MCAgency::FindAgentByID (int *id*)** [inline]

Find an agent by its ID.

Finds an agent in the agency by its ID number.

**Parameters:**

*id* The ID number of the condition to signal.

**Returns:**

The return value of the underlying MC\_FindAgentByID function.

Definition at line 1005 of file MCAgency.cs.

References \_MC\_FindAgentByID(), and Agency.

**7.48.4.37 MCAgent LibMC::MCAgency::RetrieveAgent ()** [inline]

Retrieve an agent from the agency.

Really not sure.

**Returns:**

The return value of the underlying MC\_CondBroadcast function.

Definition at line 1017 of file MCAgency.cs.

References \_MC\_RetrieveAgent(), and Agency.

**7.48.4.38 int LibMC::MCAgency::AclSend (MCAclMessage *acl\_message*)** [inline]

Send an ACL message to the agency.

Sends an ACL message to the agency. The message is delivered appropriately.

**Parameters:**

*acl\_message* The message to send.

**Returns:**

The return value of the underlying MC\_CondBroadcast function.

Definition at line 1034 of file MCAgency.cs.

References LibMC::MCAclMessage::AclMsg, and Agency.

**7.48.4.39 int LibMC::MCAgency::MainLoop ()** [inline]

Makes the agency wait indefinitely.

Makes the agency wait indefinitely until it receives a "quit" command or is otherwise terminated.

**Returns:**

The return value of the underlying MC\_MainLoop function.

Definition at line 1047 of file MCAgency.cs.

References \_MC\_MainLoop(), and Agency.

**7.48.4.40** static IntPtr LibMC::MCAgency::\_MC\_Initialize (int *port*, ref MCAgencyOptions\_t *options*) [private]

**7.48.4.41** static int LibMC::MCAgency::\_MC\_End (IntPtr *agency*) [private]

**7.48.4.42** static int LibMC::MCAgency::\_MC\_ChInitializeOptions (IntPtr *agency*, ChOptions\_t *options*) [private]

Referenced by ChInitializeOptions().

**7.48.4.43** static int LibMC::MCAgency::\_MC\_InitializeAgencyOptions (ref MCAgencyOptions\_t *options*) [private]

Referenced by MCAgency().

**7.48.4.44** static int LibMC::MCAgency::\_MC\_SetThreadsAllOn (ref MCAgencyOptions\_t *options*) [private]

Referenced by SetThreadsAllOn().

**7.48.4.45** static int LibMC::MCAgency::\_MC\_SetThreadsAllOff (ref MCAgencyOptions\_t *options*) [private]

Referenced by SetThreadsAllOff().

**7.48.4.46** static int LibMC::MCAgency::\_MC\_SetThreadOn (ref MCAgencyOptions\_t *options*, MC\_ThreadIndex\_e *index*) [private]

Referenced by SetThreadOn().

**7.48.4.47** static int LibMC::MCAgency::\_MC\_SetThreadOff (ref MCAgencyOptions\_t *options*, MC\_ThreadIndex\_e *index*) [private]

Referenced by SetThreadOff().

**7.48.4.48** static int LibMC::MCAgency::\_MC\_HaltAgency (IntPtr *agency*) [private]

Referenced by HaltAgency().

**7.48.4.49** static int LibMC::MCAgency::\_MC\_ResumeAgency (IntPtr *agency*) [private]

Referenced by ResumeAgency().



**7.48.4.50** static int LibMC::MCAgency::\_MC\_SetDefaultAgentStatus (IntPtr *agency*, MCAgent.MC\_AgentStatus\_e *status*) [private]

Referenced by SetDefaultAgentStatus().

**7.48.4.51** static IntPtr LibMC::MCAgency::\_MC\_WaitRetrieveAgent (IntPtr *agency*) [private]

Referenced by WaitRetrieveAgent().

**7.48.4.52** static int LibMC::MCAgency::\_MC\_WaitAgent (IntPtr *agency*) [private]

Referenced by WaitAgent().

**7.48.4.53** static int LibMC::MCAgency::\_MC\_MainLoop (IntPtr *agency*) [private]

Referenced by MainLoop().

**7.48.4.54** static int LibMC::MCAgency::\_MC\_SendAgentMigrationMessageFile (IntPtr *agency*, String *filename*, String *hostname*, int *port*) [private]

**7.48.4.55** static int LibMC::MCAgency::\_MC\_SendAgentMigrationMessage (IntPtr *agency*, String *message*, String *hostname*, int *port*) [private]

Referenced by SendAgentMigrationMessage().

**7.48.4.56** static int LibMC::MCAgency::\_MC\_CondBroadcast (IntPtr *agency*, int *id*) [private]

Referenced by CondBroadcast().

**7.48.4.57** static int LibMC::MCAgency::\_MC\_CondSignal (IntPtr *agency*, int *id*) [private]

Referenced by CondSignal().

**7.48.4.58** static int LibMC::MCAgency::\_MC\_CondReset (IntPtr *agency*, int *id*) [private]

Referenced by CondReset().

**7.48.4.59** static int LibMC::MCAgency::\_MC\_CondWait (IntPtr *agency*, int *id*) [private]

Referenced by CondWait().

**7.48.4.60** static int LibMC::MCAgency::\_MC\_MutexLock (IntPtr *agency*, int *id*) [private]

Referenced by MutexLock().

**7.48.4.61** `static int LibMC::MCAgency::_MC_MutexUnlock (IntPtr agency, int id)` [private]

Referenced by MutexUnlock().

**7.48.4.62** `static int LibMC::MCAgency::_MC_SemaphorePost (IntPtr agency, int id)`  
[private]

Referenced by SemaphorePost().

**7.48.4.63** `static int LibMC::MCAgency::_MC_SemaphoreWait (IntPtr agency, int id)`  
[private]

Referenced by SemaphoreWait().

**7.48.4.64** `static int LibMC::MCAgency::_MC_ResetSignal (IntPtr agency)` [private]

Referenced by ResetSignal().

**7.48.4.65** `static int LibMC::MCAgency::_MC_SyncDelete (IntPtr agency, int id)` [private]

Referenced by SyncDelete().

**7.48.4.66** `static int LibMC::MCAgency::_MC_SyncInit (IntPtr agency, int id)` [private]

Referenced by SyncInit().

**7.48.4.67** `static int LibMC::MCAgency::_MC_WaitSignal (IntPtr agency, int signals)`  
[private]

Referenced by WaitSignal().

**7.48.4.68** `static int LibMC::MCAgency::_MC_BarrierDelete (IntPtr agency, int id)` [private]

Referenced by BarrierDelete().

**7.48.4.69** `static int LibMC::MCAgency::_MC_BarrierInit (IntPtr agency, int id, int num_procs)`  
[private]

Referenced by BarrierInit().

**7.48.4.70** `static MC_SteerCommand_e LibMC::MCAgency::_MC_SteerControl ()` [private]

Referenced by SteerControl().

**7.48.4.71** `static int LibMC::MCAgency::_MC_Steer (IntPtr agency, IntPtr funcptr, IntPtr arg)`  
[private]

**7.48.4.72** `static int LibMC::MCAgency::_MC_RegisterService (IntPtr agency, IntPtr agent, int agentID, String agentName, String[] serviceNames, int numServices)` [private]

Referenced by RegisterService().

**7.48.4.73** `static int LibMC::MCAgency::_MC_SearchForService (IntPtr agency, String searchString, IntPtr agentNames, IntPtr serviceNames, IntPtr agentIDs, IntPtr numResults)` [private]

**7.48.4.74** `static int LibMC::MCAgency::_MC_AddAgent (IntPtr agency, IntPtr agent)`  
[private]

Referenced by AddAgent().

**7.48.4.75** `static internal int LibMC::MCAgency::_MC_DeleteAgent (IntPtr agent)` [private]

**7.48.4.76** `static IntPtr LibMC::MCAgency::_MC_FindAgentByName (IntPtr agency, String name)` [private]

Referenced by FindAgentByName().

**7.48.4.77** `static IntPtr LibMC::MCAgency::_MC_FindAgentByID (IntPtr agency, int ID)`  
[private]

Referenced by FindAgentByID().

**7.48.4.78** `static IntPtr LibMC::MCAgency::_MC_RetrieveAgent (IntPtr agency)` [private]

Referenced by RetrieveAgent().



- 
- 7.48.4.79 **static internal int** LibMC::MCAgency::\_MC\_GetAgentID (IntPtr *agent*) [private]
  - 7.48.4.80 **static internal String** LibMC::MCAgency::\_MC\_GetAgentName (IntPtr *agent*) [private]
  - 7.48.4.81 **static internal int** LibMC::MCAgency::\_MC\_GetAgentNumTasks (IntPtr *agent*) [private]
  - 7.48.4.82 **static internal MCAgent.MC\_AgentStatus\_e** LibMC::MCAgency::\_MC\_GetAgentStatus (IntPtr *agent*) [private]
  - 7.48.4.83 **static internal MCAgent.MC\_AgentType\_e** LibMC::MCAgency::\_MC\_GetAgentType (IntPtr *agent*) [private]
  - 7.48.4.84 **static internal String** LibMC::MCAgency::\_MC\_GetAgentXMLString (IntPtr *agent*) [private]
  - 7.48.4.85 **static internal int** LibMC::MCAgency::\_MC\_PrintAgentCode (IntPtr *agent*) [private]
  - 7.48.4.86 **static internal String** LibMC::MCAgency::\_MC\_RetrieveAgentCode (IntPtr *agent*) [private]
  - 7.48.4.87 **static internal int** LibMC::MCAgency::\_MC\_SetAgentStatus (IntPtr *agent*, MCAgent.MC\_AgentStatus\_e *status*) [private]
  - 7.48.4.88 **static internal int** LibMC::MCAgency::\_MC\_TerminateAgent (IntPtr *agent*) [private]
  - 7.48.4.89 **static internal int** LibMC::MCAgency::\_MC\_CallAgentFunc (IntPtr *agent*, String *funcName*, IntPtr *returnVal*, IntPtr *varg*) [private]
  - 7.48.4.90 **static internal IntPtr** LibMC::MCAgency::\_MC\_GetAgentExecEngine (IntPtr *agent*) [private]
  - 7.48.4.91 **static internal int** LibMC::MCAgency::\_MC\_GetAgentReturnData (IntPtr *agent*, int *task\_num*, IntPtr *data*, IntPtr *dim*, IntPtr *extent*) [private]
  - 7.48.4.92 **static internal IntPtr** LibMC::MCAgency::\_MC\_AclNew () [private]
  - 7.48.4.93 **static internal int** LibMC::MCAgency::\_MC\_AclPost (IntPtr *agent*, IntPtr *message*) [private]
  - 7.48.4.94 **static internal IntPtr** LibMC::MCAgency::\_MC\_AclReply (IntPtr *acl\_message*) [private]
  - 7.48.4.95 **static internal IntPtr** LibMC::MCAgency::\_MC\_AclRetrieve (IntPtr *agent*) [private]
  - 7.48.4.96 **static internal int** LibMC::MCAgency::\_MC\_AclSend (IntPtr *agency*, IntPtr *acl\_message*) [private]
  - 7.48.4.97 **static internal IntPtr** LibMC::MCAgency::\_MC\_AclWaitRetrieve (IntPtr *agent*) [private]
- 
- Generated on Tue Jul 1 15:29:57 2008 for Mobile-C by Doxygen
- 7.48.4.98 **static internal int** LibMC::MCAgency::\_MC\_AclSetPerformative (IntPtr *acl*, MCAclMessage.MC\_FipaPerformative\_e *performative*) [private]
  - 7.48.4.99 **static internal int** LibMC::MCAgency::\_MC\_AclSetSender (IntPtr *acl*, String *name*, String *address*) [private]

**7.48.5.2 MCAgencyOptions\_t LibMC::MCAgency::options** [private]

Definition at line 358 of file MCAgency.cs.

Referenced by ChInitializeOptions(), Initialize(), MCAgency(), SetThreadOff(), SetThreadOn(), SetThreadsAllOff(), and SetThreadsAllOn().

**7.48.5.3 int LibMC::MCAgency::port = -1** [private]

Definition at line 359 of file MCAgency.cs.

Referenced by Initialize().

**7.48.5.4 MCAgencyState LibMC::MCAgency::state = MCAgencyState.NoState** [private]

Definition at line 360 of file MCAgency.cs.

Referenced by End(), HaltAgency(), Initialize(), MCAgency(), and ResumeAgency().

**7.48.5.5 const String LibMC::MCAgency::mcdll** [private]

**Initial value:**

```
"libmc.dll"
```

Definition at line 15 of file MCExports.cs.

**7.48.6 Property Documentation****7.48.6.1 IntPtr LibMC::MCAgency::Agency** [get, set, private]

Definition at line 377 of file MCAgency.cs.

Referenced by AclSend(), AddAgent(), BarrierDelete(), BarrierInit(), ChInitializeOptions(), CondBroadcast(), CondReset(), CondSignal(), CondWait(), End(), FindAgentByID(), FindAgentByName(), HaltAgency(), Initialize(), LoadAgentMigrationMessageFile(), MainLoop(), MutexLock(), MutexUnlock(), RegisterService(), ResetSignal(), ResumeAgency(), RetrieveAgent(), SemaphorePost(), SemaphoreWait(), SendAgentMigrationMessage(), SendAgentMigrationMessageFile(), SetDefaultAgentStatus(), SyncDelete(), SyncInit(), WaitAgent(), WaitRetrieveAgent(), and WaitSignal().

**7.48.6.2 int LibMC::MCAgency::Port** [get, set]

Accessor for the port number of the agency.

Allows the user to set the agency port or get the port number while it is running.

**Note:**

The port must be set before the agency is started. Once the agency is started, the port cannot be changed.

Definition at line 404 of file MCAgency.cs.

Referenced by LoadAgentMigrationMessageFile().

#### 7.48.6.3 MCAgencyState LibMC::MCAgency::State [get]

Accessor for the agency state.

Allows the user to query the state of the agency.

**Note:**

The state cannot be set by the user. It is controlled internally.

Definition at line 424 of file MCAgency.cs.

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

- [/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAgency.cs](#)
- [/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCExports.cs](#)

## 7.49 LibMC::MCAgency::ChOptions\_t Struct Reference

ChOptions structures.

### Data Fields

- [int shelltype](#)
- [String chhome](#)

### 7.49.1 Detailed Description

ChOptions structures.

Allows the user to set the shell type and the home directory.

#### Note:

This struct is pulled directly from the Mobile-C library.

Definition at line 61 of file MCEExports.cs.

### 7.49.2 Field Documentation

#### 7.49.2.1 int LibMC::MCAgency::ChOptions\_t::shelltype

shell type: CH\_REGULARCH or CH\_SAFECH

Definition at line 63 of file MCEExports.cs.

Referenced by LibMC::MCAgency::ChInitializeOptions().

#### 7.49.2.2 String LibMC::MCAgency::ChOptions\_t::chhome

Embedded Ch home directory. if NULL, use Ch home directory for standard/professional edition as Embedded Ch home directory

Definition at line 66 of file MCEExports.cs.

Referenced by LibMC::MCAgency::ChInitializeOptions().

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

- [/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCEExports.cs](#)



## 7.50 LibMC::MCAgency::MCAgency\_t Struct Reference

### Data Fields

- [int client](#)
- [int server](#)
- [String hostName](#)
- [String filename](#)
- [int portno](#)
- [int portnoc](#)
- [IntPtr mc\\_platform](#)
- [int default\\_agentstatus](#)
- [int threads](#)
- [int enable\\_security](#)
- [int\[ \] stack\\_size](#)
- [int last\\_error](#)

### 7.50.1 Detailed Description

Definition at line 23 of file MCEExports.cs.

### 7.50.2 Field Documentation

#### 7.50.2.1 `int LibMC::MCAgency::MCAgency_t::client`

Definition at line 25 of file MCEExports.cs.

#### 7.50.2.2 `int LibMC::MCAgency::MCAgency_t::server`

Definition at line 26 of file MCEExports.cs.

#### 7.50.2.3 `String LibMC::MCAgency::MCAgency_t::hostName`

Local Hostname

Definition at line 27 of file MCEExports.cs.

#### 7.50.2.4 `String LibMC::MCAgency::MCAgency_t::filename`

Definition at line 28 of file MCEExports.cs.

#### 7.50.2.5 `int LibMC::MCAgency::MCAgency_t::portno`

Local port number

Definition at line 29 of file MCEExports.cs.

**7.50.2.6 int LibMC::MCAgency::MCAgency\_t::portnoc**

Definition at line 30 of file MCEExports.cs.

**7.50.2.7 IntPtr LibMC::MCAgency::MCAgency\_t::mc\_platform**

Local MobileC Platform

Definition at line 31 of file MCEExports.cs.

**7.50.2.8 int LibMC::MCAgency::MCAgency\_t::default\_agentstatus**

Agency default agent status

Definition at line 32 of file MCEExports.cs.

**7.50.2.9 int LibMC::MCAgency::MCAgency\_t::threads**

flag which determines which threads to start

Definition at line 33 of file MCEExports.cs.

**7.50.2.10 int LibMC::MCAgency::MCAgency\_t::enable\_security**

Security flag

Definition at line 34 of file MCEExports.cs.

**7.50.2.11 int [ ] LibMC::MCAgency::MCAgency\_t::stack\_size**

Definition at line 37 of file MCEExports.cs.

**7.50.2.12 int LibMC::MCAgency::MCAgency\_t::last\_error**

Definition at line 38 of file MCEExports.cs.

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

- [/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCEExports.cs](#)

## 7.51 LibMC::MCAgency::MCAgencyOptions\_t Struct Reference

### Data Fields

- [int threads](#)
- [int default\\_agent\\_status](#)
- [int modified](#)
- [int enable\\_security](#)
- [int\[ \] stack\\_size](#)

### 7.51.1 Detailed Description

Definition at line 42 of file MCExports.cs.

### 7.51.2 Field Documentation

#### 7.51.2.1 int LibMC::MCAgency::MCAgencyOptions\_t::threads

Threads to start

Definition at line 44 of file MCExports.cs.

#### 7.51.2.2 int LibMC::MCAgency::MCAgencyOptions\_t::default\_agent\_status

Default agent status

Definition at line 45 of file MCExports.cs.

#### 7.51.2.3 int LibMC::MCAgency::MCAgencyOptions\_t::modified

unused

Definition at line 46 of file MCExports.cs.

#### 7.51.2.4 int LibMC::MCAgency::MCAgencyOptions\_t::enable\_security

security enable flag

Definition at line 47 of file MCExports.cs.

#### 7.51.2.5 int [ ] LibMC::MCAgency::MCAgencyOptions\_t::stack\_size

If the stack size is -1, use system def.

Definition at line 50 of file MCExports.cs.

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

- [/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCExports.cs](#)

## 7.52 MCAgencyOptions\_s Struct Reference

User modifiable agency options.

```
#include <libmc.h>
```

### Data Fields

- [int threads](#)
- [int default\\_agent\\_status](#)
- [int modified](#)
- [int enable\\_security](#)
- [int stack\\_size](#) [MC\_THREAD\_ALL]

### 7.52.1 Detailed Description

User modifiable agency options.

Definition at line 235 of file libmc.h.

### 7.52.2 Field Documentation

#### 7.52.2.1 int MCAgencyOptions\_s::threads

Threads to start

Definition at line 236 of file libmc.h.

Referenced by MC\_Initialize(), MC\_InitializeAgencyOptions(), MC\_SetThreadOff(), MC\_SetThreadOn(), MC\_SetThreadsAllOff(), and MC\_SetThreadsAllOn().

#### 7.52.2.2 int MCAgencyOptions\_s::default\_agent\_status

Default agent status

Definition at line 237 of file libmc.h.

Referenced by MC\_Initialize(), and MC\_InitializeAgencyOptions().

#### 7.52.2.3 int MCAgencyOptions\_s::modified

unused

Definition at line 238 of file libmc.h.

Referenced by MC\_InitializeAgencyOptions().

#### 7.52.2.4 int MCAgencyOptions\_s::enable\_security

security enable flag

Definition at line 239 of file libmc.h.

Referenced by MC\_Initialize(), and MC\_InitializeAgencyOptions().

#### 7.52.2.5 int MCAgencyOptions\_s::stack\_size[MC\_THREAD\_ALL]

If the stack size is -1, use system def.

Definition at line 242 of file libmc.h.

Referenced by MC\_Initialize(), and MC\_InitializeAgencyOptions().

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

- </home/dko/projects/mobilec/trunk/src/include/libmc.h>

## 7.53 LibMC::MCAgent Class Reference

Wrapper class for MCAgent\_t structure.

### Public Types

- enum [MC\\_AgentType\\_e](#)  
*Enum for describing the type of an agent.*
- enum [MC\\_AgentStatus\\_e](#)  
*Enum for describing the status of an agent.*

### Public Member Functions

- [MCAgent](#) ()  
*Default constructor.*
- override string [ToString](#) ()  
*Display the agent's fields.*
- int [DeleteAgent](#) ()  
*Deletes an agent.*
- String [GetAgentXMLString](#) ()  
*Gets the agent's XML string.*
- int [PrintAgentCode](#) ()  
*Gets the agent's C code string.*
- String [RetrieveAgentCode](#) ()  
*Gets the agent's C code string.*
- int [TerminateAgent](#) ()  
*Terminates an agent.*
- int [AclPost](#) (MCAclMessage message)  
*Posts an ACL message to the agent.*
- MCAclMessage [AclRetrieve](#) ()  
*Retrieve an ACL message from the agent.*
- MCAclMessage [AclWaitRetrieve](#) ()  
*Wait for and retrieve an ACL message from the agent.*
- int [CallAgentFunc](#) (String funcName, IntPtr retval, IntPtr varg)  
*Calls a function in an agent script.*
- IntPtr [GetAgentExecEngine](#) ()

*Gets an agent's Ch interpreter.*

- `int GetAgentReturnData (int task_num, IntPtr data, IntPtr dim, IntPtr extent)`

*Calls a function in an agent script.*

## Static Public Member Functions

- static implicit `operator IntPtr (MCAgent agent)`
- static implicit `operator MCAgent (IntPtr ip)`

## Properties

- internal `IntPtr Agent` [get, set]
- `int AgentID` [get]  
*Gets the agent's ID number.*
- `String AgentName` [get]  
*Gets the agent's name.*
- `int AgentNumTasks` [get]  
*Gets the agent's number of tasks.*
- `MC_AgentStatus_e AgentStatus` [get, set]  
*Gets or sets the agent's status.*
- `MC_AgentType_e AgentType` [get]  
*Gets the agent's type.*

## Private Member Functions

- internal `MCAgent (IntPtr ip)`
- `void GetAgentFields ()`

## Private Attributes

- `IntPtr agent_p`
- `String name = ""`
- `int id = -1`
- `int numTasks = -1`
- `MC_AgentStatus_e status = MC_AgentStatus_e.MC_NO_STATUS`
- `MC_AgentType_e type = MC_AgentType_e.MC_NONE`

### 7.53.1 Detailed Description

Wrapper class for MCAgent\_t structure.

This class provides an interface to the Mobile-C agent structure. Member functions for the class are generally overloaded versions of the respective functions in the Mobile-C library. The class maintains a pointer to a Mobile-C agent in unmanaged memory. The pointer is not accessible by the user.

Definition at line 60 of file MCAgent.cs.

### 7.53.2 Member Enumeration Documentation

#### 7.53.2.1 enum LibMC::MCAgent::MC\_AgentType\_e

Enum for describing the type of an agent.

**Note:**

This enum is pulled directly from the Mobile-C library.

**Enumerator:**

*MC\_NONE* Default value to describe uninitialized agent.

*MC\_REMOTE\_AGENT* A remote agent.

*MC\_LOCAL\_AGENT* A local agent.

*MC\_RETURN\_AGENT* A returning agent.

Definition at line 74 of file MCAgent.cs.

#### 7.53.2.2 enum LibMC::MCAgent::MC\_AgentStatus\_e

Enum for describing the status of an agent.

**Note:**

This enum is pulled directly from the Mobile-C library.

**Enumerator:**

*MC\_NO\_STATUS* Default value for uninitialized agent

*MC\_WAIT\_CH* Waiting to be started

*MC\_WAIT\_MESSGSEND* Finished, waiting to migrate

*MC\_AGENT\_ACTIVE* Running

*MC\_AGENT\_NEUTRAL* Not running, but do not flush

*MC\_AGENT\_SUSPENDED* Unused

*MC\_WAIT\_FINISHED* Finished, waiting to be flushed

Definition at line 87 of file MCAgent.cs.



### 7.53.3 Constructor & Destructor Documentation

#### 7.53.3.1 LibMC::MCAgent::MCAgent () [inline]

Default constructor.

Creates an empty agent.

Definition at line 103 of file MCAgent.cs.

Referenced by operator MCAgent().

#### 7.53.3.2 internal LibMC::MCAgent::MCAgent (IntPtr ip) [inline, private]

Definition at line 109 of file MCAgent.cs.

References Agent.

### 7.53.4 Member Function Documentation

#### 7.53.4.1 override string LibMC::MCAgent::ToString () [inline]

Display the agent's fields.

Formats and returns a string with all of the agent's properties.

##### Returns:

A string containing a formatted representation of the agent's properties.

##### Note:

The agency port and any other options must be set before calling this function.

Definition at line 125 of file MCAgent.cs.

References AgentID, AgentName, AgentNumTasks, AgentStatus, and AgentType.

#### 7.53.4.2 void LibMC::MCAgent::GetAgentFields () [inline, private]

Definition at line 162 of file MCAgent.cs.

References Agent, name, numTasks, status, and type.

#### 7.53.4.3 static implicit LibMC::MCAgent::operator IntPtr (MCAgent agent) [inline, static]

Definition at line 277 of file MCAgent.cs.

References Agent.

#### 7.53.4.4 static implicit LibMC::MCAgent::operator MCAgent (IntPtr ip) [inline, static]

Definition at line 282 of file MCAgent.cs.

References MCAgent().

**7.53.4.5 int LibMC::MCAgent::DeleteAgent () [inline]**

Deletes an agent.

Deletes an agent from the agency.

**Returns:**

The return value of the underlying MC\_DeleteAgent function call.

Definition at line 299 of file MCAgent.cs.

References Agent.

**7.53.4.6 String LibMC::MCAgent::GetAgentXMLString () [inline]**

Gets the agent's XML string.

Returns the full XML string associated with the agent.

**Returns:**

The return value of the underlying MC\_GetAgentXMLString function call.

Definition at line 312 of file MCAgent.cs.

References Agent.

**7.53.4.7 int LibMC::MCAgent::PrintAgentCode () [inline]**

Gets the agent's C code string.

Prints the C code associated with the agent to stdout.

**Returns:**

The return value of the underlying MC\_PrintAgentCode function call.

Definition at line 325 of file MCAgent.cs.

References Agent.

**7.53.4.8 String LibMC::MCAgent::RetrieveAgentCode () [inline]**

Gets the agent's C code string.

Returns the C code associated with the agent.

**Returns:**

A string containing the agent's C code.

Definition at line 337 of file MCAgent.cs.

References Agent.

**7.53.4.9 int LibMC::MCAgent::TerminateAgent () [inline]**

Terminates an agent.

Terminates an agent regardless of the agent's state.

**Returns:**

The return value of the underlying MC\_TerminateAgent function call.

Definition at line 350 of file MCAgent.cs.

References Agent.

**7.53.4.10 int LibMC::MCAgent::AclPost (MCAclMessage *message*) [inline]**

Posts an ACL message to the agent.

Delivers an ACL message to the agent.

**Parameters:**

*message* The ACL message object to deliver.

**Returns:**

The return value of the underlying MC\_AclPost function call.

**Note:**

The message must be a valid message or this function call will fail.

Definition at line 375 of file MCAgent.cs.

References LibMC::MCAclMessage::AclMsg, and Agent.

**7.53.4.11 MCAclMessage LibMC::MCAgent::AclRetrieve () [inline]**

Retrieve an ACL message from the agent.

Retrieves an ACL message from the agent if one is available.

**Returns:**

The ACL message or a blank ACL message if one was not available.

**Note:**

The message must be a valid message or this function call will fail.

Definition at line 391 of file MCAgent.cs.

References Agent.

**7.53.4.12** `MCACLMessage LibMC::MCAgent::AclWaitRetrieve ()` `[inline]`

Wait for and retrieve an ACL message from the agent.

Retrieves an ACL message from the agent when one becomes available.

**Returns:**

The ACL message or a blank ACL message if the call fails.

**Note:**

This function call blocks.

Definition at line 410 of file MCAgent.cs.

References Agent.

**7.53.4.13** `int LibMC::MCAgent::CallAgentFunc (String funcName, IntPtr retval, IntPtr varg)`  
`[inline]`

Calls a function in an agent script.

Calls a function in an agent's script file. This function requires manual marshaling by the user.

**Parameters:**

*funcName* The name of the function to call

*retval* A pointer to memory for the return value

*varg* A pointer to the argument for the function

**Returns:**

The return value of the underlying MC\_CallAgentFunc function call.

**Note:**

BE VERY CAREFUL! You must marshal your arguments!

**Todo**

Find a better way to handle retval and varg

Definition at line 434 of file MCAgent.cs.

References Agent.

**7.53.4.14** `IntPtr LibMC::MCAgent::GetAgentExecEngine ()` `[inline]`

Gets an agent's Ch interpreter.

Gets a pointer to the agent's Ch interpreter. Will be improved shortly.

**Returns:**

A pointer to the Ch interpreter.

**Note:**

Nothing in the LibMC.NET library can make use of the Ch interpreter yet.

**Todo**

Wrap MC\_GetAgentExecEngine with an object for the void\* pointer return type (Ch interpreter).

Definition at line 451 of file MCAgent.cs.

References Agent.

**7.53.4.15 int LibMC::MCAgent::GetAgentReturnData (int *task\_num*, IntPtr *data*, IntPtr *dim*, IntPtr *extent*) [inline]**

Calls a function in an agent script.

Calls a function in an agent's script file. This function requires manual marshaling by the user.

**Parameters:**

*task\_num* Task number to get data from

*data* A pointer to memory for the data

*dim* A pointer to hold the dimensions of the data

*extent* A pointer to hold the dimensions of the data

**Returns:**

The return value of the underlying MC\_GetAgentReturnData function call.

**Note:**

This function does nothing but throw an exception right now.

**Todo**

Implement GetAgentReturnData

Definition at line 472 of file MCAgent.cs.

**7.53.5 Field Documentation****7.53.5.1 IntPtr LibMC::MCAgent::agent\_p [private]**

Definition at line 62 of file MCAgent.cs.

**7.53.5.2 String LibMC::MCAgent::name = "" [private]**

Definition at line 63 of file MCAgent.cs.

Referenced by GetAgentFields().

**7.53.5.3 int LibMC::MCAgent::id = -1 [private]**

Definition at line 64 of file MCAgent.cs.

**7.53.5.4** `int LibMC::MCAgent::numTasks = -1` [private]

Definition at line 65 of file MCAgent.cs.

Referenced by GetAgentFields().

**7.53.5.5** `MC_AgentStatus_e LibMC::MCAgent::status = MC_AgentStatus_e.MC_NO_STATUS`  
[private]

Definition at line 66 of file MCAgent.cs.

Referenced by GetAgentFields().

**7.53.5.6** `MC_AgentType_e LibMC::MCAgent::type = MC_AgentType_e.MC_NONE`  
[private]

Definition at line 67 of file MCAgent.cs.

Referenced by GetAgentFields().

**7.53.6 Property Documentation****7.53.6.1** `internal IntPtr LibMC::MCAgent::Agent` [get, set, private]

Definition at line 142 of file MCAgent.cs.

Referenced by AclPost(), AclRetrieve(), AclWaitRetrieve(), LibMC::MCAgency::AddAgent(), CallAgent-  
Func(), DeleteAgent(), GetAgentExecEngine(), GetAgentFields(), GetAgentXMLString(), MCAgent(),  
operator IntPtr(), PrintAgentCode(), LibMC::MCAgency::RegisterService(), RetrieveAgentCode(), and  
TerminateAgent().

**7.53.6.2** `int LibMC::MCAgent::AgentID` [get]

Gets the agent's ID number.

Gets the agent's ID number as assigned by Mobile-C if the agent is a valid agent.

**Returns:**

The agent's ID number or -1 for an empty agent.

Definition at line 180 of file MCAgent.cs.

Referenced by ToString().

**7.53.6.3** `String LibMC::MCAgent::AgentName` [get]

Gets the agent's name.

Gets the agent's name as assigned by Mobile-C or the agent script if the agent is a valid agent.

**Returns:**

The agent's name or an empty string for an empty agent.

Definition at line 200 of file MCAgent.cs.

Referenced by ToString().

#### 7.53.6.4 int LibMC::MCAgent::AgentNumTasks [get]

Gets the agent's number of tasks.

Gets the agent's ID number of tasks if the agent is a valid agent.

##### Returns:

The agent's ID number of tasks or -1 for an empty agent.

Definition at line 219 of file MCAgent.cs.

Referenced by ToString().

#### 7.53.6.5 MC\_AgentStatus\_e LibMC::MCAgent::AgentStatus [get, set]

Gets or sets the agent's status.

Gets or sets the agent's status. When setting the status, the status is double-checked after setting it and may not be set depending on the state of the agent and the agency.

##### Returns:

The agent's status or MC\_NO\_STATUS for an empty agent.

Definition at line 241 of file MCAgent.cs.

Referenced by ToString().

#### 7.53.6.6 MC\_AgentType\_e LibMC::MCAgent::AgentType [get]

Gets the agent's type.

Gets the agent's type.

##### Returns:

The agent's type or MC\_NONE for an empty agent.

Definition at line 263 of file MCAgent.cs.

Referenced by ToString().

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

- </home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAgent.cs>

## 7.54 md2\_context Struct Reference

MD2 context structure.

```
#include <md2.h>
```

### Data Fields

- unsigned char [cksum](#) [16]
- unsigned char [state](#) [48]
- unsigned char [buffer](#) [16]
- unsigned char [ipad](#) [64]
- unsigned char [opad](#) [64]
- [int](#) [left](#)

### 7.54.1 Detailed Description

MD2 context structure.

Definition at line 17 of file md2.h.

### 7.54.2 Field Documentation

#### 7.54.2.1 unsigned char md2\_context::cksum[16]

checksum of the data block

Definition at line 19 of file md2.h.

Referenced by [md2\\_finish\(\)](#), and [md2\\_process\(\)](#).

#### 7.54.2.2 unsigned char md2\_context::state[48]

intermediate digest state

Definition at line 20 of file md2.h.

Referenced by [md2\\_finish\(\)](#), and [md2\\_process\(\)](#).

#### 7.54.2.3 unsigned char md2\_context::buffer[16]

data block being processed

Definition at line 21 of file md2.h.

Referenced by [md2\\_finish\(\)](#), [md2\\_process\(\)](#), and [md2\\_update\(\)](#).

#### 7.54.2.4 unsigned char md2\_context::ipad[64]

HMAC: inner padding

Definition at line 22 of file md2.h.

Referenced by [md2\\_hmac\\_starts\(\)](#).



**7.54.2.5 unsigned char md2\_context::opad[64]**

HMAC: outer padding

Definition at line 23 of file md2.h.

Referenced by md2\_hmac\_finish(), and md2\_hmac\_starts().

**7.54.2.6 int md2\_context::left**

amount of data in buffer

Definition at line 24 of file md2.h.

Referenced by md2\_finish(), and md2\_update().

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

- [/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/md2.h](#)

## 7.55 md4\_context Struct Reference

MD4 context structure.

```
#include <md4.h>
```

### Data Fields

- unsigned long [total](#) [2]
- unsigned long [state](#) [4]
- unsigned char [buffer](#) [64]
- unsigned char [ipad](#) [64]
- unsigned char [opad](#) [64]

### 7.55.1 Detailed Description

MD4 context structure.

Definition at line 17 of file md4.h.

### 7.55.2 Field Documentation

#### 7.55.2.1 unsigned long md4\_context::total[2]

number of bytes processed

Definition at line 19 of file md4.h.

Referenced by md4\_finish(), md4\_starts(), and md4\_update().

#### 7.55.2.2 unsigned long md4\_context::state[4]

intermediate digest state

Definition at line 20 of file md4.h.

Referenced by md4\_finish(), md4\_process(), and md4\_starts().

#### 7.55.2.3 unsigned char md4\_context::buffer[64]

data block being processed

Definition at line 21 of file md4.h.

Referenced by md4\_update().

#### 7.55.2.4 unsigned char md4\_context::ipad[64]

HMAC: inner padding

Definition at line 22 of file md4.h.

Referenced by md4\_hmac\_starts().

#### 7.55.2.5 unsigned char md4\_context::opad[64]

HMAC: outer padding

Definition at line 23 of file md4.h.

Referenced by md4\_hmac\_finish(), and md4\_hmac\_starts().

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

- </home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/md4.h>

## 7.56 md5\_context Struct Reference

MD5 context structure.

```
#include <md5.h>
```

### Data Fields

- unsigned long [total](#) [2]
- unsigned long [state](#) [4]
- unsigned char [buffer](#) [64]
- unsigned char [ipad](#) [64]
- unsigned char [opad](#) [64]

### 7.56.1 Detailed Description

MD5 context structure.

Definition at line 17 of file md5.h.

### 7.56.2 Field Documentation

#### 7.56.2.1 unsigned long md5\_context::total[2]

number of bytes processed

Definition at line 19 of file md5.h.

Referenced by md5\_finish(), md5\_starts(), and md5\_update().

#### 7.56.2.2 unsigned long md5\_context::state[4]

intermediate digest state

Definition at line 20 of file md5.h.

Referenced by md5\_finish(), md5\_process(), and md5\_starts().

#### 7.56.2.3 unsigned char md5\_context::buffer[64]

data block being processed

Definition at line 21 of file md5.h.

Referenced by md5\_update().

#### 7.56.2.4 unsigned char md5\_context::ipad[64]

HMAC: inner padding

Definition at line 22 of file md5.h.

Referenced by md5\_hmac\_starts().

#### 7.56.2.5 unsigned char md5\_context::opad[64]

HMAC: outer padding

Definition at line 23 of file md5.h.

Referenced by md5\_hmac\_finish(), and md5\_hmac\_starts().

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

- </home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/md5.h>

## 7.57 message\_s Struct Reference

```
#include <message.h>
```

### Data Fields

- struct sockaddr\_in \* [addr](#)
- [int connect\\_id](#)
- [int message\\_id](#)
- [int isHTTP](#)
- enum [message\\_type\\_e](#) [message\\_type](#)
- enum [http\\_performative\\_e](#) [http\\_type](#)
- [mxml\\_node\\_t](#) \* [xml\\_root](#)
- [mxml\\_node\\_t](#) \* [xml\\_payload](#)
- char \* [message\\_body](#)
- char \* [update\\_name](#)
- [int update\\_num](#)
- char \* [from\\_address](#)
- char \* [to\\_address](#)
- char \* [target](#)
- [int agent\\_xml\\_flag](#)

### 7.57.1 Detailed Description

Definition at line 72 of file message.h.

### 7.57.2 Field Documentation

#### 7.57.2.1 struct sockaddr\_in\* message\_s::addr [read]

Definition at line 74 of file message.h.

Referenced by [message\\_Destroy\(\)](#), [message\\_InitializeFromAgent\(\)](#), [message\\_InitializeFromConnection\(\)](#), [message\\_InitializeFromString\(\)](#), and [message\\_New\(\)](#).

#### 7.57.2.2 int message\_s::connect\_id

Definition at line 77 of file message.h.

Referenced by [message\\_InitializeFromConnection\(\)](#), [message\\_InitializeFromString\(\)](#), and [message\\_New\(\)](#).

#### 7.57.2.3 int message\_s::message\_id

Definition at line 78 of file message.h.

Referenced by [AP\\_QUEUE\\_SEARCH\\_TEMPLATE\(\)](#), [message\\_InitializeFromAgent\(\)](#), [message\\_InitializeFromConnection\(\)](#), [message\\_InitializeFromString\(\)](#), and [message\\_New\(\)](#).

**7.57.2.4 int message\_s::isHTTP**

Definition at line 81 of file message.h.

Referenced by message\_New(), mtp\_http\_ComposeMessage(), and mtp\_http\_CreateMessage().

**7.57.2.5 enum message\_type\_e message\_s::message\_type**

Definition at line 84 of file message.h.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), agent\_Initialize(), MC\_AclSend(), message\_InitializeFromAgent(), message\_InitializeFromString(), and message\_New().

**7.57.2.6 enum http\_performative\_e message\_s::http\_type**

Definition at line 85 of file message.h.

Referenced by message\_New().

**7.57.2.7 mxml\_node\_t\* message\_s::xml\_root**

Definition at line 87 of file message.h.

Referenced by acc\_Thread(), agent\_Initialize(), MC\_LoadAgentFromFile(), message\_Destroy(), message\_InitializeFromAgent(), message\_InitializeFromConnection(), message\_InitializeFromString(), and message\_New().

**7.57.2.8 mxml\_node\_t\* message\_s::xml\_payload**

Definition at line 88 of file message.h.

Referenced by agent\_Initialize(), MC\_LoadAgentFromFile(), and message\_New().

**7.57.2.9 char\* message\_s::message\_body**

Definition at line 91 of file message.h.

Referenced by acc\_Thread(), message\_Destroy(), message\_InitializeFromAgent(), message\_InitializeFromConnection(), message\_InitializeFromString(), message\_New(), message\_Send(), mtp\_http\_ComposeMessage(), and mtp\_http\_CreateMessage().

**7.57.2.10 char\* message\_s::update\_name**

Definition at line 93 of file message.h.

Referenced by message\_Destroy(), message\_InitializeFromAgent(), message\_InitializeFromString(), and message\_New().

**7.57.2.11 int message\_s::update\_num**

Definition at line 94 of file message.h.

Referenced by message\_New().

**7.57.2.12 char\* message\_s::from\_address**

Definition at line 96 of file message.h.

Referenced by acc\_MessageHandlerThread(), AP\_QUEUE\_SEARCH\_TEMPLATE(), message\_Destroy(), message\_InitializeFromAgent(), message\_InitializeFromConnection(), message\_InitializeFromString(), and message\_New().

**7.57.2.13 char\* message\_s::to\_address**

Definition at line 97 of file message.h.

Referenced by acc\_MessageHandlerThread(), AP\_QUEUE\_SEARCH\_TEMPLATE(), MC\_LoadAgentFromFile(), message\_Destroy(), message\_InitializeFromAgent(), message\_InitializeFromConnection(), message\_InitializeFromString(), message\_New(), message\_queue\_SendOutgoing(), message\_Send(), mtp\_http\_ComposeMessage(), and mtp\_http\_CreateMessage().

**7.57.2.14 char\* message\_s::target**

Definition at line 99 of file message.h.

Referenced by MC\_AclSend(), message\_Destroy(), message\_InitializeFromAgent(), message\_InitializeFromConnection(), message\_InitializeFromString(), message\_New(), and mtp\_http\_ComposeMessage().

**7.57.2.15 int message\_s::agent\_xml\_flag**

Definition at line 106 of file message.h.

Referenced by agent\_Initialize(), message\_Destroy(), message\_InitializeFromAgent(), and message\_New().

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

- /home/dko/projects/mobilec/trunk/src/include/[message.h](#)



## 7.58 mpi Struct Reference

MPI structure.

```
#include <bignum.h>
```

### Data Fields

- [int s](#)
- [int n](#)
- [t\\_int \\* p](#)

### 7.58.1 Detailed Description

MPI structure.

Definition at line 56 of file bignum.h.

### 7.58.2 Field Documentation

#### 7.58.2.1 int mpi::s

integer sign

Definition at line 58 of file bignum.h.

Referenced by [mpi\\_add\\_int\(\)](#), [mpi\\_add\\_mpi\(\)](#), [mpi\\_cmp\\_int\(\)](#), [mpi\\_cmp\\_mpi\(\)](#), [mpi\\_copy\(\)](#), [mpi\\_div\\_int\(\)](#), [mpi\\_div\\_mpi\(\)](#), [mpi\\_gcd\(\)](#), [mpi\\_grow\(\)](#), [mpi\\_is\\_prime\(\)](#), [mpi\\_lset\(\)](#), [mpi\\_montred\(\)](#), [mpi\\_mul\\_int\(\)](#), [mpi\\_mul\\_mpi\(\)](#), [mpi\\_read\\_string\(\)](#), [mpi\\_sub\\_int\(\)](#), [mpi\\_sub\\_mpi\(\)](#), and [mpi\\_write\\_string\(\)](#).

#### 7.58.2.2 int mpi::n

total # of limbs

Definition at line 59 of file bignum.h.

Referenced by [mpi\\_add\\_abs\(\)](#), [mpi\\_add\\_int\(\)](#), [mpi\\_cmp\\_abs\(\)](#), [mpi\\_cmp\\_int\(\)](#), [mpi\\_cmp\\_mpi\(\)](#), [mpi\\_copy\(\)](#), [mpi\\_div\\_int\(\)](#), [mpi\\_div\\_mpi\(\)](#), [mpi\\_exp\\_mod\(\)](#), [mpi\\_free\(\)](#), [mpi\\_grow\(\)](#), [mpi\\_is\\_prime\(\)](#), [mpi\\_lsb\(\)](#), [mpi\\_lset\(\)](#), [mpi\\_mod\\_int\(\)](#), [mpi\\_montmul\(\)](#), [mpi\\_montred\(\)](#), [mpi\\_msb\(\)](#), [mpi\\_mul\\_int\(\)](#), [mpi\\_mul\\_mpi\(\)](#), [mpi\\_shift\\_l\(\)](#), [mpi\\_shift\\_r\(\)](#), [mpi\\_sub\\_abs\(\)](#), [mpi\\_sub\\_int\(\)](#), [mpi\\_write\\_string\(\)](#), and [x509\\_cert\\_info\(\)](#).

#### 7.58.2.3 t\_int\* mpi::p

pointer to limbs

Definition at line 60 of file bignum.h.

Referenced by [dhm\\_make\\_params\(\)](#), [dhm\\_make\\_public\(\)](#), [mpi\\_add\\_abs\(\)](#), [mpi\\_add\\_int\(\)](#), [mpi\\_cmp\\_abs\(\)](#), [mpi\\_cmp\\_int\(\)](#), [mpi\\_cmp\\_mpi\(\)](#), [mpi\\_copy\(\)](#), [mpi\\_div\\_int\(\)](#), [mpi\\_div\\_mpi\(\)](#), [mpi\\_exp\\_mod\(\)](#), [mpi\\_free\(\)](#), [mpi\\_gcd\(\)](#), [mpi\\_gen\\_prime\(\)](#), [mpi\\_grow\(\)](#), [mpi\\_inv\\_mod\(\)](#), [mpi\\_is\\_prime\(\)](#), [mpi\\_lsb\(\)](#), [mpi\\_lset\(\)](#), [mpi\\_mod\\_int\(\)](#), [mpi\\_montg\\_init\(\)](#), [mpi\\_montmul\(\)](#), [mpi\\_montred\(\)](#), [mpi\\_msb\(\)](#), [mpi\\_mul\\_int\(\)](#), [mpi\\_mul\\_mpi\(\)](#), [mpi\\_read\\_binary\(\)](#), [mpi\\_read\\_string\(\)](#), [mpi\\_shift\\_l\(\)](#), [mpi\\_shift\\_r\(\)](#), [mpi\\_sub\\_abs\(\)](#), [mpi\\_sub\\_int\(\)](#), [mpi\\_write\\_binary\(\)](#), [mpi\\_write\\_string\(\)](#), and [rsa\\_check\\_pubkey\(\)](#).

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

- </home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/bignum.h>

## 7.59 mtp\_http\_content\_s Struct Reference

```
#include <mtp_http.h>
```

### Data Fields

- char \* [content\\_type](#)
- void \* [data](#)

### 7.59.1 Detailed Description

Definition at line 106 of file mtp\_http.h.

### 7.59.2 Field Documentation

#### 7.59.2.1 char\* mtp\_http\_content\_s::content\_type

Definition at line 108 of file mtp\_http.h.

Referenced by MC\_AclSend(), mtp\_http\_CreateMessage(), mtp\_http\_Destroy(), and mtp\_http\_Parse().

#### 7.59.2.2 void\* mtp\_http\_content\_s::data

Definition at line 109 of file mtp\_http.h.

Referenced by acc\_Thread(), MC\_AclSend(), mtp\_http\_CreateMessage(), mtp\_http\_Destroy(), and mtp\_http\_Parse().

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

- /home/dko/projects/mobilec/trunk/src/include/[mtp\\_http.h](#)

## 7.60 mtp\_http\_s Struct Reference

```
#include <mtp_http.h>
```

### Data Fields

- enum [http\\_status\\_code\\_e](#) `http_status_code`
- enum [http\\_performative\\_e](#) `http_performative`
- char \* [http\\_version](#)
- char \* [host](#)
- char \* [return\\_code](#)
- char \* [target](#)
- char \* [date](#)
- char \* [server](#)
- char \* [accept\\_ranges](#)
- char \* [content\\_length](#)
- char \* [connection](#)
- char \* [content\\_type](#)
- char \* [user\\_agent](#)
- char \* [cache\\_control](#)
- char \* [mime\\_version](#)
- int [response\\_code](#)
- char \* [response\\_string](#)
- int [message\\_parts](#)
- char \* [boundary](#)
- struct [mtp\\_http\\_content\\_s](#) \* [content](#)

### 7.60.1 Detailed Description

Definition at line 112 of file `mtp_http.h`.

### 7.60.2 Field Documentation

#### 7.60.2.1 enum `http_status_code_e` `mtp_http_s::http_status_code`

Definition at line 114 of file `mtp_http.h`.

#### 7.60.2.2 enum `http_performative_e` `mtp_http_s::http_performative`

Definition at line 115 of file `mtp_http.h`.

Referenced by `acc_Thread()`, `http_ParseHeader()`, and `mtp_http_Parse()`.

#### 7.60.2.3 char\* `mtp_http_s::http_version`

Definition at line 116 of file `mtp_http.h`.

Referenced by `mtp_http_Destroy()`.

**7.60.2.4 char\* mtp\_http\_s::host**

Definition at line 117 of file mtp\_http.h.

Referenced by MC\_AclSend(), mtp\_http\_CreateMessage(), and mtp\_http\_Destroy().

**7.60.2.5 char\* mtp\_http\_s::return\_code**

Definition at line 118 of file mtp\_http.h.

Referenced by mtp\_http\_Destroy().

**7.60.2.6 char\* mtp\_http\_s::target**

Definition at line 119 of file mtp\_http.h.

Referenced by acc\_Thread(), http\_ParseHeader(), MC\_AclSend(), mtp\_http\_CreateMessage(), and mtp\_http\_Destroy().

**7.60.2.7 char\* mtp\_http\_s::date**

Definition at line 121 of file mtp\_http.h.

Referenced by mtp\_http\_Destroy().

**7.60.2.8 char\* mtp\_http\_s::server**

Definition at line 122 of file mtp\_http.h.

Referenced by mtp\_http\_Destroy().

**7.60.2.9 char\* mtp\_http\_s::accept\_ranges**

Definition at line 123 of file mtp\_http.h.

Referenced by mtp\_http\_Destroy().

**7.60.2.10 char\* mtp\_http\_s::content\_length**

Definition at line 124 of file mtp\_http.h.

Referenced by mtp\_http\_Destroy().

**7.60.2.11 char\* mtp\_http\_s::connection**

Definition at line 125 of file mtp\_http.h.

Referenced by mtp\_http\_Destroy().

**7.60.2.12 char\* mtp\_http\_s::content\_type**

Definition at line 126 of file mtp\_http.h.

Referenced by `mtp_http_Destroy()`, and `mtp_http_Parse()`.

#### **7.60.2.13 char\* mtp\_http\_s::user\_agent**

Definition at line 127 of file `mtp_http.h`.

Referenced by `mtp_http_Destroy()`.

#### **7.60.2.14 char\* mtp\_http\_s::cache\_control**

Definition at line 129 of file `mtp_http.h`.

#### **7.60.2.15 char\* mtp\_http\_s::mime\_version**

Definition at line 130 of file `mtp_http.h`.

#### **7.60.2.16 int mtp\_http\_s::response\_code**

Definition at line 133 of file `mtp_http.h`.

Referenced by `http_ParseHeader()`.

#### **7.60.2.17 char\* mtp\_http\_s::response\_string**

Definition at line 134 of file `mtp_http.h`.

Referenced by `http_ParseHeader()`.

#### **7.60.2.18 int mtp\_http\_s::message\_parts**

Definition at line 138 of file `mtp_http.h`.

Referenced by `acc_Thread()`, `MC_AclSend()`, `mtp_http_CreateMessage()`, `mtp_http_Destroy()`, and `mtp_http_Parse()`.

#### **7.60.2.19 char\* mtp\_http\_s::boundary**

Definition at line 139 of file `mtp_http.h`.

Referenced by `mtp_http_Destroy()`, and `mtp_http_Parse()`.

#### **7.60.2.20 struct mtp\_http\_content\_s\* mtp\_http\_s::content** [read]

Definition at line 140 of file `mtp_http.h`.

Referenced by `acc_Thread()`, `MC_AclSend()`, `mtp_http_CreateMessage()`, `mtp_http_Destroy()`, `mtp_http_New()`, and `mtp_http_Parse()`.

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

- `/home/dko/projects/mobilec/trunk/src/include/mtp\_http.h`

## 7.61 mxml\_attr\_s Struct Reference

```
#include <mxml.h>
```

### Data Fields

- char \* [name](#)
- char \* [value](#)

### 7.61.1 Detailed Description

Definition at line 86 of file mxml.h.

### 7.61.2 Field Documentation

#### 7.61.2.1 char\* mxml\_attr\_s::name

Definition at line 88 of file mxml.h.

Referenced by `mxml_write_node()`, `mxmlDelete()`, `mxmlElementGetAttr()`, and `mxmlElementSetAttr()`.

#### 7.61.2.2 char\* mxml\_attr\_s::value

Definition at line 89 of file mxml.h.

Referenced by `mxml_write_node()`, `mxmlDelete()`, `mxmlElementGetAttr()`, and `mxmlElementSetAttr()`.

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

- `/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml.h`

## 7.62 mxml\_custom\_s Struct Reference

```
#include <mxml.h>
```

### Data Fields

- void \* [data](#)
- void(\* [destroy](#) )(void \*)

### 7.62.1 Detailed Description

Definition at line 105 of file mxml.h.

### 7.62.2 Field Documentation

#### 7.62.2.1 void\* mxml\_custom\_s::data

Definition at line 107 of file mxml.h.

Referenced by mxmlDelete(), mxmlNewCustom(), and mxmlSetCustom().

#### 7.62.2.2 void(\* mxml\_custom\_s::destroy)(void \*)

Referenced by mxmlDelete(), mxmlNewCustom(), and mxmlSetCustom().

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

- /home/dko/projects/mobilec/trunk/src/mxml-2.2.2/[mxml.h](#)



## 7.63 mxml\_fdbuf\_s Struct Reference

### Data Fields

- [int fd](#)
- unsigned char \* [current](#)
- unsigned char \* [end](#)
- unsigned char [buffer](#) [8192]

### 7.63.1 Detailed Description

Definition at line 83 of file mxml-file.c.

### 7.63.2 Field Documentation

#### 7.63.2.1 int mxml\_fdbuf\_s::fd

Definition at line 85 of file mxml-file.c.

Referenced by [mxml\\_fd\\_read\(\)](#), [mxml\\_fd\\_write\(\)](#), [mxxmlLoadFd\(\)](#), and [mxxmlSaveFd\(\)](#).

#### 7.63.2.2 unsigned char\* mxml\_fdbuf\_s::current

Definition at line 86 of file mxml-file.c.

Referenced by [mxml\\_fd\\_getc\(\)](#), [mxml\\_fd\\_putc\(\)](#), [mxml\\_fd\\_read\(\)](#), [mxml\\_fd\\_write\(\)](#), [mxxmlLoadFd\(\)](#), and [mxxmlSaveFd\(\)](#).

#### 7.63.2.3 unsigned char \* mxml\_fdbuf\_s::end

Definition at line 86 of file mxml-file.c.

Referenced by [mxml\\_fd\\_getc\(\)](#), [mxml\\_fd\\_putc\(\)](#), [mxml\\_fd\\_read\(\)](#), [mxxmlLoadFd\(\)](#), and [mxxmlSaveFd\(\)](#).

#### 7.63.2.4 unsigned char mxml\_fdbuf\_s::buffer[8192]

Definition at line 86 of file mxml-file.c.

Referenced by [mxml\\_fd\\_read\(\)](#), [mxml\\_fd\\_write\(\)](#), [mxxmlLoadFd\(\)](#), and [mxxmlSaveFd\(\)](#).

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

- [/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-file.c](#)

## 7.64 mxml\_index\_s Struct Reference

```
#include <mxml.h>
```

### Data Fields

- char \* [attr](#)
- int [num\\_nodes](#)
- int [alloc\\_nodes](#)
- int [cur\\_node](#)
- [mxml\\_node\\_t](#) \*\* [nodes](#)

### 7.64.1 Detailed Description

Definition at line 133 of file mxml.h.

### 7.64.2 Field Documentation

#### 7.64.2.1 char\* mxml\_index\_s::attr

Definition at line 135 of file mxml.h.

Referenced by [index\\_compare\(\)](#), [index\\_find\(\)](#), [mxmlIndexDelete\(\)](#), [mxmlIndexFind\(\)](#), and [mxmlIndexNew\(\)](#).

#### 7.64.2.2 int mxml\_index\_s::num\_nodes

Definition at line 136 of file mxml.h.

Referenced by [main\(\)](#), [mxmlIndexEnum\(\)](#), [mxmlIndexFind\(\)](#), [mxmlIndexNew\(\)](#), and [mxmlIndexReset\(\)](#).

#### 7.64.2.3 int mxml\_index\_s::alloc\_nodes

Definition at line 137 of file mxml.h.

Referenced by [mxmlIndexDelete\(\)](#), and [mxmlIndexNew\(\)](#).

#### 7.64.2.4 int mxml\_index\_s::cur\_node

Definition at line 138 of file mxml.h.

Referenced by [mxmlIndexEnum\(\)](#), [mxmlIndexFind\(\)](#), and [mxmlIndexReset\(\)](#).

#### 7.64.2.5 mxml\_node\_t\*\* mxml\_index\_s::nodes

Definition at line 139 of file mxml.h.

Referenced by [index\\_sort\(\)](#), [mxmlIndexDelete\(\)](#), [mxmlIndexEnum\(\)](#), [mxmlIndexFind\(\)](#), [mxmlIndexNew\(\)](#), and [mxmlIndexReset\(\)](#).

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

- </home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml.h>

## 7.65 mxml\_node\_s Struct Reference

```
#include <mxml.h>
```

### Data Fields

- [mxml\\_type\\_t](#) type
- struct [mxml\\_node\\_s](#) \* [next](#)
- struct [mxml\\_node\\_s](#) \* [prev](#)
- struct [mxml\\_node\\_s](#) \* [parent](#)
- struct [mxml\\_node\\_s](#) \* [child](#)
- struct [mxml\\_node\\_s](#) \* [last\\_child](#)
- [mxml\\_value\\_t](#) value

### 7.65.1 Detailed Description

Definition at line 122 of file `mxml.h`.

### 7.65.2 Field Documentation

#### 7.65.2.1 mxml\_type\_t mxml\_node\_s::type

Definition at line 124 of file `mxml.h`.

Referenced by `agent_xml_parse__fill_row_data()`, `fipa_envelope_HandleAclRepresentation()`, `fipa_envelope_HandleComments()`, `fipa_envelope_HandleDate()`, `fipa_envelope_HandlePayloadEncoding()`, `fipa_envelope_HandlePayloadLength()`, `fipa_envelope_ParseAddresses()`, `fipa_envelope_ParseAgentIdentifier()`, `main()`, `mxml_new()`, `mxml_write_node()`, `mxmlDelete()`, `mxmlElementGetAttr()`, `mxmlElementSetAttr()`, `mxmlFindElement()`, `mxmlSetCustom()`, `mxmlSetElement()`, `mxmlSetInteger()`, `mxmlSetOpaque()`, `mxmlSetReal()`, `mxmlSetText()`, `mxmlSetTextf()`, `write_element()`, `xml_get_cdata()`, `xml_get_element_name()`, `xml_get_next_element()`, and `xml_get_text()`.

#### 7.65.2.2 struct mxml\_node\_s\* mxml\_node\_s::next [read]

Definition at line 125 of file `mxml.h`.

Referenced by `add_variable()`, `main()`, `mxml_write_node()`, `mxmlAdd()`, `mxmlFindElement()`, `mxmlRemove()`, `mxmlWalkNext()`, `scan_file()`, `sort_node()`, `xml_get_next_element()`, and `xml_get_text()`.

#### 7.65.2.3 struct mxml\_node\_s\* mxml\_node\_s::prev [read]

Definition at line 126 of file `mxml.h`.

Referenced by `mxml_write_node()`, `mxmlAdd()`, `mxmlRemove()`, and `mxmlWalkPrev()`.

#### 7.65.2.4 struct mxml\_node\_s\* mxml\_node\_s::parent [read]

Definition at line 127 of file `mxml.h`.

Referenced by `agent_xml_parse__data()`, `mxml_load_data()`, `mxmlAdd()`, `mxmlRemove()`, `mxmlWalkNext()`, `mxmlWalkPrev()`, `sort_node()`, `whitespace_cb()`, `ws_cb()`, and `xml_find_sibling()`.

**7.65.2.5 struct mxml\_node\_s\* mxml\_node\_s::child** [read]

Definition at line 128 of file mxml.h.

Referenced by add\_variable(), agent\_xml\_parse\_\_fill\_row\_data(), fipa\_envelope\_HandleAclRepresentation(), fipa\_envelope\_HandleComments(), fipa\_envelope\_HandleDate(), fipa\_envelope\_HandlePayloadEncoding(), fipa\_envelope\_HandlePayloadLength(), fipa\_envelope\_ParseAddresses(), fipa\_envelope\_ParseAgentIdentifier(), main(), mxml\_write\_node(), mxmlAdd(), mxmlDelete(), mxmlRemove(), mxmlWalkNext(), scan\_file(), sort\_node(), whitespace\_cb(), write\_documentation(), write\_element(), and xml\_get\_text().

**7.65.2.6 struct mxml\_node\_s\* mxml\_node\_s::last\_child** [read]

Definition at line 129 of file mxml.h.

Referenced by add\_variable(), main(), mxmlAdd(), mxmlRemove(), mxmlWalkPrev(), and scan\_file().

**7.65.2.7 mxml\_value\_t mxml\_node\_s::value**

Definition at line 130 of file mxml.h.

Referenced by add\_variable(), agent\_xml\_parse\_\_fill\_row\_data(), fipa\_envelope\_HandleAclRepresentation(), fipa\_envelope\_HandleComments(), fipa\_envelope\_HandleDate(), fipa\_envelope\_HandlePayloadEncoding(), fipa\_envelope\_HandlePayloadLength(), fipa\_envelope\_ParseAddresses(), fipa\_envelope\_ParseAgentIdentifier(), index\_compare(), index\_find(), main(), mxml\_get\_entity(), mxml\_load\_data(), mxml\_parse\_element(), mxml\_write\_node(), mxmlDelete(), mxmlElementGetAttr(), mxmlElementSetAttr(), mxmlFindElement(), mxmlIndexNew(), mxmlNewCustom(), mxmlNewElement(), mxmlNewInteger(), mxmlNewOpaque(), mxmlNewReal(), mxmlNewText(), mxmlNewTextf(), mxmlSetCustom(), mxmlSetElement(), mxmlSetInteger(), mxmlSetOpaque(), mxmlSetReal(), mxmlSetText(), mxmlSetTextf(), scan\_file(), sort\_node(), type\_cb(), update\_comment(), whitespace\_cb(), write\_element(), ws\_cb(), xml\_get\_element\_name(), and xml\_get\_text().

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

- /home/dko/projects/mobilec/trunk/src/mxml-2.2.2/[mxml.h](#)

## 7.66 mxml\_text\_s Struct Reference

```
#include <mxml.h>
```

### Data Fields

- [int whitespace](#)
- [char \\* string](#)

### 7.66.1 Detailed Description

Definition at line 99 of file mxml.h.

### 7.66.2 Field Documentation

#### 7.66.2.1 int mxml\_text\_s::whitespace

Definition at line 101 of file mxml.h.

Referenced by `add_variable()`, `main()`, `mxml_write_node()`, `mxmlNewText()`, `mxmlNewTextf()`, `mxmlSetText()`, `mxmlSetTextf()`, `scan_file()`, and `write_element()`.

#### 7.66.2.2 char\* mxml\_text\_s::string

Definition at line 102 of file mxml.h.

Referenced by `add_variable()`, `agent_xml_parse__fill_row_data()`, `fipa_envelope_HandleAclRepresentation()`, `fipa_envelope_HandleComments()`, `fipa_envelope_HandleDate()`, `fipa_envelope_HandlePayloadEncoding()`, `fipa_envelope_HandlePayloadLength()`, `fipa_envelope_ParseAddresses()`, `fipa_envelope_ParseAgentIdentifier()`, `main()`, `mxml_write_node()`, `mxmlDelete()`, `mxmlNewText()`, `mxmlNewTextf()`, `mxmlSetText()`, `mxmlSetTextf()`, `scan_file()`, `update_comment()`, `write_element()`, and `xml_get_text()`.

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

- [/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml.h](#)

## 7.67 mxml\_value\_s Struct Reference

```
#include <mxml.h>
```

### Data Fields

- `char * name`
- `int num_attrs`
- `mxml_attr_t * attrs`

### 7.67.1 Detailed Description

Definition at line 92 of file `mxml.h`.

### 7.67.2 Field Documentation

#### 7.67.2.1 `char* mxml_value_s::name`

Definition at line 94 of file `mxml.h`.

Referenced by `index_compare()`, `index_find()`, `main()`, `mxml_get_entity()`, `mxml_load_data()`, `mxml_parse_element()`, `mxml_write_node()`, `mxmlDelete()`, `mxmlElementSetAttr()`, `mxmlFindElement()`, `mxmllIndexNew()`, `mxmlNewElement()`, `mxmlSetElement()`, `scan_file()`, `sort_node()`, `type_cb()`, `update_comment()`, `whitespace_cb()`, `ws_cb()`, `xml_get_element_name()`, and `xml_get_text()`.

#### 7.67.2.2 `int mxml_value_s::num_attrs`

Definition at line 95 of file `mxml.h`.

Referenced by `mxml_write_node()`, `mxmlDelete()`, `mxmlElementGetAttr()`, and `mxmlElementSetAttr()`.

#### 7.67.2.3 `mxml_attr_t* mxml_value_s::attrs`

Definition at line 96 of file `mxml.h`.

Referenced by `mxml_write_node()`, `mxmlDelete()`, `mxmlElementGetAttr()`, and `mxmlElementSetAttr()`.

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

- `/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml.h`

## 7.68 mxml\_value\_u Union Reference

```
#include <mxml.h>
```

### Data Fields

- [mxml\\_element\\_t](#) element
- [int](#) integer
- [char \\*](#) opaque
- [double](#) real
- [mxml\\_text\\_t](#) text
- [mxml\\_custom\\_t](#) custom

### 7.68.1 Detailed Description

Definition at line 112 of file mxml.h.

### 7.68.2 Field Documentation

#### 7.68.2.1 mxml\_element\_t mxml\_value\_u::element

Definition at line 114 of file mxml.h.

Referenced by [index\\_compare\(\)](#), [index\\_find\(\)](#), [main\(\)](#), [mxml\\_get\\_entity\(\)](#), [mxml\\_load\\_data\(\)](#), [mxml\\_parse\\_element\(\)](#), [mxml\\_write\\_node\(\)](#), [mxmlDelete\(\)](#), [mxmlElementGetAttr\(\)](#), [mxmlElementSetAttr\(\)](#), [mxmlFindElement\(\)](#), [mxmlIndexNew\(\)](#), [mxmlNewElement\(\)](#), [mxmlSetElement\(\)](#), [scan\\_file\(\)](#), [sort\\_node\(\)](#), [type\\_cb\(\)](#), [update\\_comment\(\)](#), [whitespace\\_cb\(\)](#), [ws\\_cb\(\)](#), [xml\\_get\\_element\\_name\(\)](#), and [xml\\_get\\_text\(\)](#).

#### 7.68.2.2 int mxml\_value\_u::integer

Definition at line 115 of file mxml.h.

Referenced by [main\(\)](#), [mxml\\_write\\_node\(\)](#), [mxmlNewInteger\(\)](#), and [mxmlSetInteger\(\)](#).

#### 7.68.2.3 char\* mxml\_value\_u::opaque

Definition at line 116 of file mxml.h.

Referenced by [main\(\)](#), [mxml\\_write\\_node\(\)](#), [mxmlDelete\(\)](#), [mxmlNewOpaque\(\)](#), and [mxmlSetOpaque\(\)](#).

#### 7.68.2.4 double mxml\_value\_u::real

Definition at line 117 of file mxml.h.

Referenced by [main\(\)](#), [mxml\\_write\\_node\(\)](#), [mxmlNewReal\(\)](#), and [mxmlSetReal\(\)](#).

#### 7.68.2.5 mxml\_text\_t mxml\_value\_u::text

Definition at line 118 of file mxml.h.



Referenced by `add_variable()`, `agent_xml_parse__fill_row_data()`, `fipa_envelope_HandleAclRepresentation()`, `fipa_envelope_HandleComments()`, `fipa_envelope_HandleDate()`, `fipa_envelope_HandlePayloadEncoding()`, `fipa_envelope_HandlePayloadLength()`, `fipa_envelope_ParseAddresses()`, `fipa_envelope_ParseAgentIdentifier()`, `main()`, `mxml_write_node()`, `mxmlDelete()`, `mxmlNewText()`, `mxmlNewTextf()`, `mxmlSetText()`, `mxmlSetTextf()`, `scan_file()`, `update_comment()`, `write_element()`, and `xml_get_text()`.

#### 7.68.2.6 mxml\_custom\_t mxml\_value\_u::custom

Definition at line 119 of file `mxml.h`.

Referenced by `mxmlDelete()`, `mxmlNewCustom()`, and `mxmlSetCustom()`.

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

- `/home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml.h`

## 7.69 rsa\_context Struct Reference

```
#include <rsa.h>
```

### Data Fields

- [int ver](#)
- [int len](#)
- [mpi N](#)
- [mpi E](#)
- [mpi D](#)
- [mpi P](#)
- [mpi Q](#)
- [mpi DP](#)
- [mpi DQ](#)
- [mpi QP](#)
- [mpi RN](#)
- [mpi RP](#)
- [mpi RQ](#)

### 7.69.1 Detailed Description

Definition at line 55 of file `rsa.h`.

### 7.69.2 Field Documentation

#### 7.69.2.1 `int rsa_context::ver`

should be 0

Definition at line 57 of file `rsa.h`.

Referenced by `x509_parse_key()`.

#### 7.69.2.2 `int rsa_context::len`

size(N) in chars

Definition at line 58 of file `rsa.h`.

Referenced by `main()`, `rsa_gen_key()`, `rsa_pkcs1_decrypt()`, `rsa_pkcs1_encrypt()`, `rsa_pkcs1_sign()`, `rsa_pkcs1_verify()`, `rsa_private()`, `rsa_public()`, `rsa_read_private()`, `rsa_read_public()`, `ssl_parse_certificate_verify()`, `ssl_parse_client_key_exchange()`, `ssl_parse_server_key_exchange()`, `ssl_write_certificate_verify()`, `ssl_write_client_key_exchange()`, `ssl_write_server_key_exchange()`, `x509_add_certs()`, and `x509_parse_key()`.

#### 7.69.2.3 `mpi rsa_context::N`

public modulus

Definition at line 59 of file `rsa.h`.

Referenced by `main()`, `rsa_check_privkey()`, `rsa_check_pubkey()`, `rsa_free()`, `rsa_gen_key()`, `rsa_private()`, `rsa_public()`, `rsa_read_private()`, `rsa_read_public()`, `rsa_write_private()`, `rsa_write_public()`, `x509_add_certs()`, `x509_cert_info()`, and `x509_parse_key()`.

#### 7.69.2.4 `mpi rsa_context::E`

public exponent

Definition at line 60 of file `rsa.h`.

Referenced by `main()`, `rsa_check_privkey()`, `rsa_check_pubkey()`, `rsa_free()`, `rsa_gen_key()`, `rsa_public()`, `rsa_read_private()`, `rsa_read_public()`, `rsa_write_private()`, `rsa_write_public()`, `x509_add_certs()`, and `x509_parse_key()`.

#### 7.69.2.5 `mpi rsa_context::D`

private exponent

Definition at line 61 of file `rsa.h`.

Referenced by `main()`, `rsa_free()`, `rsa_gen_key()`, `rsa_private()`, `rsa_read_private()`, `rsa_write_private()`, and `x509_parse_key()`.

#### 7.69.2.6 `mpi rsa_context::P`

1st prime factor

Definition at line 63 of file `rsa.h`.

Referenced by `main()`, `rsa_check_privkey()`, `rsa_free()`, `rsa_gen_key()`, `rsa_private()`, `rsa_read_private()`, `rsa_write_private()`, and `x509_parse_key()`.

#### 7.69.2.7 `mpi rsa_context::Q`

2nd prime factor

Definition at line 64 of file `rsa.h`.

Referenced by `main()`, `rsa_check_privkey()`, `rsa_free()`, `rsa_gen_key()`, `rsa_private()`, `rsa_read_private()`, `rsa_write_private()`, and `x509_parse_key()`.

#### 7.69.2.8 `mpi rsa_context::DP`

$D \bmod (P - 1)$

Definition at line 65 of file `rsa.h`.

Referenced by `main()`, `rsa_free()`, `rsa_gen_key()`, `rsa_private()`, `rsa_read_private()`, `rsa_write_private()`, and `x509_parse_key()`.

#### 7.69.2.9 `mpi rsa_context::DQ`

$D \bmod (Q - 1)$

Definition at line 66 of file `rsa.h`.

Referenced by `main()`, `rsa_free()`, `rsa_gen_key()`, `rsa_private()`, `rsa_read_private()`, `rsa_write_private()`, and `x509_parse_key()`.

#### **7.69.2.10 `mpi rsa_context::QP`**

inverse of  $Q \% P$

Definition at line 67 of file `rsa.h`.

Referenced by `main()`, `rsa_free()`, `rsa_gen_key()`, `rsa_private()`, `rsa_read_private()`, `rsa_write_private()`, and `x509_parse_key()`.

#### **7.69.2.11 `mpi rsa_context::RN`**

cached  $R^2 \bmod N$

Definition at line 69 of file `rsa.h`.

Referenced by `rsa_free()`, `rsa_private()`, and `rsa_public()`.

#### **7.69.2.12 `mpi rsa_context::RP`**

cached  $R^2 \bmod P$

Definition at line 70 of file `rsa.h`.

Referenced by `rsa_free()`, and `rsa_private()`.

#### **7.69.2.13 `mpi rsa_context::RQ`**

cached  $R^2 \bmod Q$

Definition at line 71 of file `rsa.h`.

Referenced by `rsa_free()`, and `rsa_private()`.

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

- `/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/rsa.h`

## 7.70 LibMC::Properties::Settings Class Reference

### Public Member Functions

- [Settings](#) ()

### Properties

- static [Settings Default](#) [get]

### Private Member Functions

- void [SettingChangingEventHandler](#) (object sender, System.Configuration.SettingChangingEventArgs e)
- void [SettingsSavingEventHandler](#) (object sender, System.ComponentModel.CancelEventArgs e)

### Static Private Attributes

- static [Settings defaultInstance](#) = (([Settings](#))(global::System.Configuration.ApplicationSettingsBase.Synchronized(new [Settings](#)())))

#### 7.70.1 Detailed Description

Definition at line 16 of file Settings.Designer.cs.

#### 7.70.2 Constructor & Destructor Documentation

##### 7.70.2.1 LibMC::Properties::Settings::Settings () [inline]

Definition at line 19 of file Settings.cs.

#### 7.70.3 Member Function Documentation

##### 7.70.3.1 void LibMC::Properties::Settings::SettingChangingEventHandler (object sender, System.Configuration.SettingChangingEventArgs e) [inline, private]

Definition at line 28 of file Settings.cs.

##### 7.70.3.2 void LibMC::Properties::Settings::SettingsSavingEventHandler (object sender, System.ComponentModel.CancelEventArgs e) [inline, private]

Definition at line 32 of file Settings.cs.

## 7.70.4 Field Documentation

**7.70.4.1 Settings LibMC::Properties::Settings::defaultInstance =**  
**((Settings)(global::System.Configuration.ApplicationSettingsBase.Synchronized(new**  
**Settings())))** [static, private]

Definition at line 18 of file Settings.Designer.cs.

## 7.70.5 Property Documentation

**7.70.5.1 Settings LibMC::Properties::Settings::Default** [static, get]

Definition at line 20 of file Settings.Designer.cs.

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

- [/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/Properties/Settings.Designer.cs](#)
- [/home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/Settings.cs](#)

## 7.71 sha1\_context Struct Reference

SHA-1 context structure.

```
#include <sha1.h>
```

### Data Fields

- unsigned long [total](#) [2]
- unsigned long [state](#) [5]
- unsigned char [buffer](#) [64]
- unsigned char [ipad](#) [64]
- unsigned char [opad](#) [64]

### 7.71.1 Detailed Description

SHA-1 context structure.

Definition at line 17 of file sha1.h.

### 7.71.2 Field Documentation

#### 7.71.2.1 unsigned long sha1\_context::total[2]

number of bytes processed

Definition at line 19 of file sha1.h.

Referenced by sha1\_finish(), sha1\_starts(), and sha1\_update().

#### 7.71.2.2 unsigned long sha1\_context::state[5]

intermediate digest state

Definition at line 20 of file sha1.h.

Referenced by sha1\_finish(), sha1\_process(), and sha1\_starts().

#### 7.71.2.3 unsigned char sha1\_context::buffer[64]

data block being processed

Definition at line 21 of file sha1.h.

Referenced by sha1\_update().

#### 7.71.2.4 unsigned char sha1\_context::ipad[64]

HMAC: inner padding

Definition at line 22 of file sha1.h.

Referenced by sha1\_hmac\_starts().

#### 7.71.2.5 unsigned char sha1\_context::opad[64]

HMAC: outer padding

Definition at line 23 of file sha1.h.

Referenced by sha1\_hmac\_finish(), and sha1\_hmac\_starts().

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

- </home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/sha1.h>



## 7.72 sha2\_context Struct Reference

SHA-256 context structure.

```
#include <sha2.h>
```

### Data Fields

- unsigned long [total](#) [2]
- unsigned long [state](#) [8]
- unsigned char [buffer](#) [64]
- unsigned char [ipad](#) [64]
- unsigned char [opad](#) [64]
- [int is224](#)

### 7.72.1 Detailed Description

SHA-256 context structure.

Definition at line 17 of file sha2.h.

### 7.72.2 Field Documentation

#### 7.72.2.1 unsigned long sha2\_context::total[2]

number of bytes processed

Definition at line 19 of file sha2.h.

Referenced by sha2\_finish(), sha2\_starts(), and sha2\_update().

#### 7.72.2.2 unsigned long sha2\_context::state[8]

intermediate digest state

Definition at line 20 of file sha2.h.

Referenced by sha2\_finish(), sha2\_process(), and sha2\_starts().

#### 7.72.2.3 unsigned char sha2\_context::buffer[64]

data block being processed

Definition at line 21 of file sha2.h.

Referenced by sha2\_update().

#### 7.72.2.4 unsigned char sha2\_context::ipad[64]

HMAC: inner padding

Definition at line 22 of file sha2.h.

Referenced by sha2\_hmac\_starts().

**7.72.2.5 unsigned char sha2\_context::opad[64]**

HMAC: outer padding

Definition at line 23 of file sha2.h.

Referenced by sha2\_hmac\_finish(), and sha2\_hmac\_starts().

**7.72.2.6 int sha2\_context::is224**

0 if SHA-256, 1 if SHA-224

Definition at line 24 of file sha2.h.

Referenced by sha2\_finish(), sha2\_hmac\_finish(), and sha2\_starts().

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

- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/[sha2.h](#)

## 7.73 sha4\_context Struct Reference

SHA-512 context structure.

```
#include <sha4.h>
```

### Data Fields

- uint64 [total](#) [2]
- uint64 [state](#) [8]
- unsigned char [buffer](#) [128]
- unsigned char [ipad](#) [64]
- unsigned char [opad](#) [64]
- [int](#) [is384](#)

### 7.73.1 Detailed Description

SHA-512 context structure.

Definition at line 25 of file sha4.h.

### 7.73.2 Field Documentation

#### 7.73.2.1 uint64 sha4\_context::total[2]

number of bytes processed

Definition at line 27 of file sha4.h.

Referenced by sha4\_finish(), sha4\_starts(), and sha4\_update().

#### 7.73.2.2 uint64 sha4\_context::state[8]

intermediate digest state

Definition at line 28 of file sha4.h.

Referenced by sha4\_finish(), sha4\_process(), and sha4\_starts().

#### 7.73.2.3 unsigned char sha4\_context::buffer[128]

data block being processed

Definition at line 29 of file sha4.h.

Referenced by sha4\_update().

#### 7.73.2.4 unsigned char sha4\_context::ipad[64]

HMAC: inner padding

Definition at line 30 of file sha4.h.

Referenced by sha4\_hmac\_starts().

**7.73.2.5 unsigned char sha4\_context::opad[64]**

HMAC: outer padding

Definition at line 31 of file sha4.h.

Referenced by sha4\_hmac\_finish(), and sha4\_hmac\_starts().

**7.73.2.6 int sha4\_context::is384**

0 if SHA-512, 1 if SHA-384

Definition at line 32 of file sha4.h.

Referenced by sha4\_finish(), sha4\_hmac\_finish(), and sha4\_starts().

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

- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/[sha4.h](#)

## 7.74 ssl\_context Struct Reference

```
#include <ssl.h>
```

### Data Fields

- `int` `state`
- `int` `major_ver`
- `int` `minor_ver`
- `unsigned char` `max_ver` [2]
- `unsigned char *` `in_ctr`
- `unsigned char *` `in_hdr`
- `unsigned char *` `in_msg`
- `unsigned char *` `in_offt`
- `int` `read_fd`
- `int` `in_msgtype`
- `int` `in_msglen`
- `int` `in_left`
- `int` `in_hslen`
- `int` `nb_zero`
- `unsigned char *` `out_ctr`
- `unsigned char *` `out_hdr`
- `unsigned char *` `out_msg`
- `int` `write_fd`
- `int` `out_msgtype`
- `int` `out_msglen`
- `int` `out_left`
- `int` `out_uoff`
- `rsa_context *` `own_key`
- `x509_cert *` `own_cert`
- `x509_cert *` `ca_chain`
- `x509_cert *` `peer_cert`
- `char *` `peer_cn`
- `int` `endpoint`
- `int` `authmode`
- `int` `client_auth`
- `int` `verify_result`
- `int` `resumed`
- `int` `sidlen`
- `unsigned char` `sessid` [32]
- `unsigned char *` `sidtable`
- `md5_context` `hs_md5`
- `sha1_context` `hs_sha1`
- `dhm_context` `dhm_ctx`
- `int`(`* rng_f`)(`void *`)
- `void *` `rng_d`
- `int` `pmslen`
- `unsigned char` `premaster` [256]
- `unsigned char` `randbytes` [64]

- unsigned char [master](#) [48]
- [int](#) \* [cipherlist](#)
- [int](#) [cipher](#)
- [int](#) [keylen](#)
- [int](#) [minlen](#)
- [int](#) [ctxlen](#)
- void \* [ctx\\_enc](#)
- void \* [ctx\\_dec](#)
- [int](#) [ivlen](#)
- unsigned char [iv\\_enc](#) [16]
- unsigned char [iv\\_dec](#) [16]
- [int](#) [maclen](#)
- unsigned char [mac\\_enc](#) [32]
- unsigned char [mac\\_dec](#) [32]

### 7.74.1 Detailed Description

Definition at line 132 of file `ssl.h`.

### 7.74.2 Field Documentation

#### 7.74.2.1 `int ssl_context::state`

SSL handshake: current state

Definition at line 134 of file `ssl.h`.

Referenced by `ssl_client_start()`, `ssl_close_notify()`, `ssl_parse_certificate_request()`, `ssl_parse_certificate_verify()`, `ssl_parse_client_hello()`, `ssl_parse_client_key_exchange()`, `ssl_parse_server_hello()`, `ssl_parse_server_hello_done()`, `ssl_parse_server_key_exchange()`, `ssl_server_start()`, `ssl_write_certificate()`, `ssl_write_certificate_request()`, `ssl_write_certificate_verify()`, `ssl_write_client_hello()`, `ssl_write_client_key_exchange()`, `ssl_write_server_hello()`, `ssl_write_server_hello_done()`, and `ssl_write_server_key_exchange()`.

#### 7.74.2.2 `int ssl_context::major_ver`

equal to `SSLV3_MAJOR_VERSION`

Definition at line 139 of file `ssl.h`.

Referenced by `ssl_parse_client_hello()`, `ssl_write_client_hello()`, `ssl_write_record()`, and `ssl_write_server_hello()`.

#### 7.74.2.3 `int ssl_context::minor_ver`

either 0: SSLv3, or 1: TLSv1.0

Definition at line 140 of file `ssl.h`.

Referenced by `ssl_calc_verify()`, `ssl_decrypt_buf()`, `ssl_derive_keys()`, `ssl_encrypt_buf()`, `ssl_parse_client_hello()`, `ssl_parse_client_key_exchange()`, `ssl_parse_server_hello()`, `ssl_write_certificate()`, `ssl_write_client_hello()`, `ssl_write_client_key_exchange()`, `ssl_write_record()`, and `ssl_write_server_hello()`.

**7.74.2.4 unsigned char ssl\_context::max\_ver[2]**

max. version supported by client

Definition at line 141 of file ssl.h.

Referenced by ssl\_parse\_client\_hello(), ssl\_parse\_client\_key\_exchange(), ssl\_write\_client\_hello(), and ssl\_write\_client\_key\_exchange().

**7.74.2.5 unsigned char\* ssl\_context::in\_ctr**

64-bit incoming message counter

Definition at line 146 of file ssl.h.

Referenced by ssl\_decrypt\_buf(), ssl\_free(), and ssl\_init().

**7.74.2.6 unsigned char\* ssl\_context::in\_hdr**

5-byte record header (in\_ctr+8)

Definition at line 147 of file ssl.h.

Referenced by ssl\_decrypt\_buf(), ssl\_init(), and ssl\_parse\_client\_hello().

**7.74.2.7 unsigned char\* ssl\_context::in\_msg**

the message payload (in\_hdr+5)

Definition at line 148 of file ssl.h.

Referenced by ssl\_decrypt\_buf(), ssl\_init(), ssl\_parse\_certificate\_request(), ssl\_parse\_certificate\_verify(), ssl\_parse\_client\_hello(), ssl\_parse\_client\_key\_exchange(), ssl\_parse\_server\_hello(), ssl\_parse\_server\_hello\_done(), ssl\_parse\_server\_key\_exchange(), and ssl\_read().

**7.74.2.8 unsigned char\* ssl\_context::in\_offt**

read offset in message payload

Definition at line 149 of file ssl.h.

Referenced by ssl\_read().

**7.74.2.9 int ssl\_context::read\_fd**

descriptor for read operations

Definition at line 151 of file ssl.h.

Referenced by ssl\_parse\_client\_hello().

**7.74.2.10 int ssl\_context::in\_msgtype**

record header: message type

Definition at line 152 of file ssl.h.

Referenced by `ssl_decrypt_buf()`, `ssl_parse_certificate_request()`, `ssl_parse_certificate_verify()`, `ssl_parse_client_key_exchange()`, `ssl_parse_server_hello()`, `ssl_parse_server_hello_done()`, `ssl_parse_server_key_exchange()`, and `ssl_read()`.

#### **7.74.2.11 int ssl\_context::in\_msglen**

record header: message length

Definition at line 153 of file `ssl.h`.

Referenced by `ssl_decrypt_buf()`, and `ssl_read()`.

#### **7.74.2.12 int ssl\_context::in\_left**

amount of (tcp) data read so far

Definition at line 155 of file `ssl.h`.

Referenced by `ssl_parse_client_hello()`.

#### **7.74.2.13 int ssl\_context::in\_hslen**

current handshake message length

Definition at line 156 of file `ssl.h`.

Referenced by `ssl_parse_certificate_verify()`, `ssl_parse_client_key_exchange()`, `ssl_parse_server_hello()`, `ssl_parse_server_hello_done()`, and `ssl_parse_server_key_exchange()`.

#### **7.74.2.14 int ssl\_context::nb\_zero**

# of 0-length encrypted messages

Definition at line 157 of file `ssl.h`.

Referenced by `ssl_decrypt_buf()`.

#### **7.74.2.15 unsigned char\* ssl\_context::out\_ctr**

64-bit outgoing message counter

Definition at line 162 of file `ssl.h`.

Referenced by `ssl_encrypt_buf()`, `ssl_free()`, and `ssl_init()`.

#### **7.74.2.16 unsigned char\* ssl\_context::out\_hdr**

5-byte record header (`out_ctr+8`)

Definition at line 163 of file `ssl.h`.

Referenced by `ssl_flush_output()`, `ssl_init()`, and `ssl_write_record()`.



**7.74.2.17 unsigned char\* ssl\_context::out\_msg**

the message payload (out\_hdr+5)

Definition at line 164 of file ssl.h.

Referenced by ssl\_close\_notify(), ssl\_encrypt\_buf(), ssl\_init(), ssl\_write(), ssl\_write\_certificate(), ssl\_write\_certificate\_request(), ssl\_write\_certificate\_verify(), ssl\_write\_client\_hello(), ssl\_write\_client\_key\_exchange(), ssl\_write\_record(), ssl\_write\_server\_hello(), ssl\_write\_server\_hello\_done(), and ssl\_write\_server\_key\_exchange().

**7.74.2.18 int ssl\_context::write\_fd**

descriptor for write operations

Definition at line 166 of file ssl.h.

Referenced by ssl\_flush\_output(), and ssl\_write\_record().

**7.74.2.19 int ssl\_context::out\_msgtype**

record header: message type

Definition at line 167 of file ssl.h.

Referenced by ssl\_close\_notify(), ssl\_encrypt\_buf(), ssl\_write(), ssl\_write\_certificate(), ssl\_write\_certificate\_request(), ssl\_write\_certificate\_verify(), ssl\_write\_client\_hello(), ssl\_write\_client\_key\_exchange(), ssl\_write\_record(), ssl\_write\_server\_hello(), ssl\_write\_server\_hello\_done(), and ssl\_write\_server\_key\_exchange().

**7.74.2.20 int ssl\_context::out\_msglen**

record header: message length

Definition at line 168 of file ssl.h.

Referenced by ssl\_close\_notify(), ssl\_encrypt\_buf(), ssl\_flush\_output(), ssl\_write(), ssl\_write\_certificate(), ssl\_write\_certificate\_request(), ssl\_write\_certificate\_verify(), ssl\_write\_client\_hello(), ssl\_write\_client\_key\_exchange(), ssl\_write\_record(), ssl\_write\_server\_hello(), ssl\_write\_server\_hello\_done(), and ssl\_write\_server\_key\_exchange().

**7.74.2.21 int ssl\_context::out\_left**

amount of data not yet written

Definition at line 170 of file ssl.h.

Referenced by ssl\_flush\_output(), and ssl\_write\_record().

**7.74.2.22 int ssl\_context::out\_uoff**

offset in user-supplied buffer

Definition at line 171 of file ssl.h.

Referenced by ssl\_write().

**7.74.2.23 rsa\_context\* ssl\_context::own\_key**

own RSA private key

Definition at line 176 of file ssl.h.

Referenced by ssl\_parse\_certificate\_request(), ssl\_parse\_client\_key\_exchange(), ssl\_write\_certificate\_verify(), and ssl\_write\_server\_key\_exchange().

**7.74.2.24 x509\_cert\* ssl\_context::own\_cert**

own X.509 certificate

Definition at line 177 of file ssl.h.

Referenced by ssl\_parse\_certificate\_request(), and ssl\_write\_certificate().

**7.74.2.25 x509\_cert\* ssl\_context::ca\_chain**

own trusted CA chain

Definition at line 178 of file ssl.h.

Referenced by ssl\_write\_certificate\_request().

**7.74.2.26 x509\_cert\* ssl\_context::peer\_cert**

peer X.509 cert chain

Definition at line 179 of file ssl.h.

Referenced by ssl\_free(), ssl\_parse\_certificate\_verify(), ssl\_parse\_server\_key\_exchange(), and ssl\_write\_client\_key\_exchange().

**7.74.2.27 char\* ssl\_context::peer\_cn**

expected peer CN

Definition at line 180 of file ssl.h.

**7.74.2.28 int ssl\_context::endpoint**

0: client, 1: server

Definition at line 182 of file ssl.h.

Referenced by ssl\_derive\_keys(), ssl\_handshake(), ssl\_set\_endpoint(), and ssl\_write\_certificate().

**7.74.2.29 int ssl\_context::authmode**

verification mode

Definition at line 183 of file ssl.h.

Referenced by ssl\_parse\_certificate\_verify(), and ssl\_write\_certificate\_request().

**7.74.2.30 int ssl\_context::client\_auth**

flag for client auth.

Definition at line 184 of file ssl.h.

Referenced by ssl\_parse\_certificate\_request(), ssl\_parse\_server\_hello\_done(), ssl\_write\_certificate(), and ssl\_write\_certificate\_verify().

**7.74.2.31 int ssl\_context::verify\_result**

verification result

Definition at line 185 of file ssl.h.

Referenced by ssl\_get\_verify\_result().

**7.74.2.32 int ssl\_context::resumed**

session resuming flag

Definition at line 190 of file ssl.h.

Referenced by ssl\_derive\_keys(), ssl\_parse\_server\_hello(), and ssl\_write\_server\_hello().

**7.74.2.33 int ssl\_context::sidlen**

session id length

Definition at line 191 of file ssl.h.

Referenced by ssl\_get\_session(), ssl\_init(), ssl\_parse\_client\_hello(), ssl\_parse\_server\_hello(), ssl\_set\_session(), ssl\_write\_client\_hello(), and ssl\_write\_server\_hello().

**7.74.2.34 unsigned char ssl\_context::sessid[32]**

session id

Definition at line 192 of file ssl.h.

Referenced by ssl\_get\_session(), ssl\_init(), ssl\_parse\_client\_hello(), ssl\_parse\_server\_hello(), ssl\_set\_session(), ssl\_write\_client\_hello(), and ssl\_write\_server\_hello().

**7.74.2.35 unsigned char\* ssl\_context::sidtable**

table of session IDs

Definition at line 193 of file ssl.h.

Referenced by ssl\_get\_session(), and ssl\_set\_session().

**7.74.2.36 md5\_context ssl\_context::hs\_md5**

MD5( Handshake msgs )

Definition at line 198 of file ssl.h.

Referenced by `ssl_calc_verify()`, `ssl_parse_client_hello()`, `ssl_write_client_hello()`, and `ssl_write_record()`.

#### **7.74.2.37 sha1\_context ssl\_context::hs\_sha1**

SHA1( Handshake msgs )

Definition at line 199 of file `ssl.h`.

Referenced by `ssl_calc_verify()`, `ssl_parse_client_hello()`, `ssl_write_client_hello()`, and `ssl_write_record()`.

#### **7.74.2.38 dhm\_context ssl\_context::dhm\_ctx**

DHM key exchange

Definition at line 200 of file `ssl.h`.

Referenced by `ssl_free()`, `ssl_parse_client_key_exchange()`, `ssl_parse_server_key_exchange()`, `ssl_write_client_key_exchange()`, and `ssl_write_server_key_exchange()`.

#### **7.74.2.39 int(\* ssl\_context::rng\_f)(void \*)**

RNG function

Referenced by `ssl_write_client_hello()`, `ssl_write_client_key_exchange()`, `ssl_write_server_hello()`, and `ssl_write_server_key_exchange()`.

#### **7.74.2.40 void\* ssl\_context::rng\_d**

RNG data

Definition at line 203 of file `ssl.h`.

Referenced by `ssl_write_client_hello()`, `ssl_write_client_key_exchange()`, `ssl_write_server_hello()`, and `ssl_write_server_key_exchange()`.

#### **7.74.2.41 int ssl\_context::pmslen**

premaster length

Definition at line 205 of file `ssl.h`.

Referenced by `ssl_derive_keys()`, `ssl_parse_client_key_exchange()`, and `ssl_write_client_key_exchange()`.

#### **7.74.2.42 unsigned char ssl\_context::premaster[256]**

premaster secret

Definition at line 206 of file `ssl.h`.

Referenced by `ssl_derive_keys()`, `ssl_parse_client_key_exchange()`, and `ssl_write_client_key_exchange()`.

**7.74.2.43 unsigned char ssl\_context::randbytes[64]**

random bytes

Definition at line 207 of file ssl.h.

Referenced by ssl\_derive\_keys(), ssl\_parse\_client\_hello(), ssl\_parse\_server\_hello(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_client\_hello(), ssl\_write\_server\_hello(), and ssl\_write\_server\_key\_exchange().

**7.74.2.44 unsigned char ssl\_context::master[48]**

master secret

Definition at line 208 of file ssl.h.

Referenced by ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_get\_session(), ssl\_init(), and ssl\_set\_session().

**7.74.2.45 int\* ssl\_context::cipherlist**

accepted ciphersuites

Definition at line 210 of file ssl.h.

Referenced by ssl\_parse\_client\_hello(), ssl\_parse\_server\_hello(), and ssl\_write\_client\_hello().

**7.74.2.46 int ssl\_context::cipher**

current chosen cipher

Definition at line 211 of file ssl.h.

Referenced by ssl\_derive\_keys(), ssl\_get\_session(), ssl\_parse\_client\_hello(), ssl\_parse\_client\_key\_exchange(), ssl\_parse\_server\_hello(), ssl\_parse\_server\_key\_exchange(), ssl\_set\_session(), ssl\_write\_client\_key\_exchange(), ssl\_write\_server\_hello(), and ssl\_write\_server\_key\_exchange().

**7.74.2.47 int ssl\_context::keylen**

symmetric key length

Definition at line 212 of file ssl.h.

Referenced by ssl\_derive\_keys().

**7.74.2.48 int ssl\_context::minlen**

min. ciphertext length

Definition at line 213 of file ssl.h.

Referenced by ssl\_decrypt\_buf(), and ssl\_derive\_keys().

**7.74.2.49 int ssl\_context::ctxlen**

cipher context length

Definition at line 215 of file ssl.h.

Referenced by `ssl_derive_keys()`, and `ssl_free()`.

#### **7.74.2.50 void\* ssl\_context::ctx\_enc**

encryption context

Definition at line 216 of file ssl.h.

Referenced by `ssl_derive_keys()`, `ssl_encrypt_buf()`, and `ssl_free()`.

#### **7.74.2.51 void\* ssl\_context::ctx\_dec**

decryption context

Definition at line 217 of file ssl.h.

Referenced by `ssl_decrypt_buf()`, `ssl_derive_keys()`, and `ssl_free()`.

#### **7.74.2.52 int ssl\_context::ivlen**

IV length

Definition at line 219 of file ssl.h.

Referenced by `ssl_decrypt_buf()`, `ssl_derive_keys()`, and `ssl_encrypt_buf()`.

#### **7.74.2.53 unsigned char ssl\_context::iv\_enc[16]**

IV (encryption)

Definition at line 220 of file ssl.h.

Referenced by `ssl_derive_keys()`, and `ssl_encrypt_buf()`.

#### **7.74.2.54 unsigned char ssl\_context::iv\_dec[16]**

IV (decryption)

Definition at line 221 of file ssl.h.

Referenced by `ssl_decrypt_buf()`, and `ssl_derive_keys()`.

#### **7.74.2.55 int ssl\_context::maclen**

MAC length

Definition at line 223 of file ssl.h.

Referenced by `ssl_decrypt_buf()`, `ssl_derive_keys()`, and `ssl_encrypt_buf()`.

#### **7.74.2.56 unsigned char ssl\_context::mac\_enc[32]**

MAC (encryption)

Definition at line 224 of file ssl.h.

Referenced by `ssl_derive_keys()`, and `ssl_encrypt_buf()`.

#### 7.74.2.57 unsigned char ssl\_context::mac\_dec[32]

MAC (decryption)

Definition at line 225 of file ssl.h.

Referenced by `ssl_decrypt_buf()`, and `ssl_derive_keys()`.

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

- `/home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/ssl.h`

## 7.75 syncList\_s Struct Reference

```
#include <sync_list.h>
```

### Data Fields

- `RWLOCK_T * lock`
- `MUTEX_T * giant_lock`
- `list_p list`
- `int size`

### 7.75.1 Detailed Description

Definition at line 26 of file `sync_list.h`.

### 7.75.2 Field Documentation

#### 7.75.2.1 `RWLOCK_T* syncList_s::lock`

Definition at line 27 of file `sync_list.h`.

Referenced by `syncListAddNode()`, `syncListDelete()`, `syncListFind()`, `syncListInit()`, and `syncListRemove()`.

#### 7.75.2.2 `MUTEX_T* syncList_s::giant_lock`

Definition at line 28 of file `sync_list.h`.

Referenced by `MC_SyncDelete()`, and `MC_SyncInit()`.

#### 7.75.2.3 `list_p syncList_s::list`

Definition at line 30 of file `sync_list.h`.

Referenced by `syncListAddNode()`, `syncListDelete()`, `syncListFind()`, and `syncListRemove()`.

#### 7.75.2.4 `int syncList_s::size`

Definition at line 31 of file `sync_list.h`.

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

- `/home/dko/projects/mobilec/trunk/src/mc_sync/sync_list.h`



## 7.76 syncListNode\_s Struct Reference

```
#include <sync_list.h>
```

### Data Fields

- `MUTEX_T * lock`
- `COND_T * cond`
- `SEMAPHORE_T * sem`
- `int id`
- `int signalled`

### 7.76.1 Detailed Description

Definition at line 16 of file `sync_list.h`.

### 7.76.2 Field Documentation

#### 7.76.2.1 `MUTEX_T* syncListNode_s::lock`

Definition at line 17 of file `sync_list.h`.

Referenced by `MC_CondBroadcast()`, `MC_CondReset()`, `MC_CondSignal()`, `MC_CondWait()`, `MC_MutexLock()`, `MC_MutexUnlock()`, `MC_SyncDelete()`, `syncListNodeDestroy()`, `syncListNodeInit()`, and `syncListNodeNew()`.

#### 7.76.2.2 `COND_T* syncListNode_s::cond`

Definition at line 18 of file `sync_list.h`.

Referenced by `MC_CondBroadcast()`, `MC_CondSignal()`, `MC_CondWait()`, `syncListNodeDestroy()`, `syncListNodeInit()`, and `syncListNodeNew()`.

#### 7.76.2.3 `SEMAPHORE_T* syncListNode_s::sem`

Definition at line 19 of file `sync_list.h`.

Referenced by `MC_SemaphorePost()`, `MC_SemaphoreWait()`, `syncListNodeDestroy()`, `syncListNodeInit()`, and `syncListNodeNew()`.

#### 7.76.2.4 `int syncListNode_s::id`

Definition at line 20 of file `sync_list.h`.

Referenced by `MC_SyncInit()`, `syncListAddNode()`, `syncListDelete()`, and `syncListRemove()`.

#### 7.76.2.5 `int syncListNode_s::signalled`

Definition at line 21 of file `sync_list.h`.

Referenced by MC\_CondBroadcast(), MC\_CondReset(), MC\_CondSignal(), MC\_CondWait(), and syncListNodeNew().

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

- [/home/dko/projects/mobilec/trunk/src/mc\\_sync/sync\\_list.h](/home/dko/projects/mobilec/trunk/src/mc_sync/sync_list.h)

## Chapter 8

# Mobile-C File Documentation

### 8.1 /home/dko/projects/mobilec/trunk/src/acc.c File Reference

```
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#include <netdb.h>
#include <sys/un.h>
#include <unistd.h>
#include <sys/time.h>
#include <pthread.h>
#include <stdlib.h>
#include "include/acc.h"
#include "include/connection.h"
#include "include/data_structures.h"
#include "include/macros.h"
#include "include/mc_error.h"
#include "include/mc_platform.h"
#include "include/message.h"
#include "include/mtp_http.h"
#include "include/xml_parser.h"
#include "include/fipa_acl_envelope.h"
```

#### Defines

- #define [BACKLOG](#) 10

## Functions

- `acc_p acc_Initialize` (struct `mc_platform_s *mc_platform`)
- `int acc_Destroy` (`acc_p acc`)
- `void * acc_MessageHandlerThread` (`void *arg`)
- `void * acc_Thread` (`void *arg`)
- `void acc_Start` (`mc_platform_p mc_platform`)
- `void * listen_Thread` (`void *arg`)

### 8.1.1 Define Documentation

#### 8.1.1.1 #define BACKLOG 10

Definition at line 63 of file `acc.c`.

Referenced by `listen_Thread()`.

### 8.1.2 Function Documentation

#### 8.1.2.1 int acc\_Destroy (acc\_p acc)

Definition at line 82 of file `acc.c`.

References `MC_SUCCESS`.

Referenced by `mc_platform_Destroy()`.

#### 8.1.2.2 acc\_p acc\_Initialize (struct mc\_platform\_s \* mc\_platform)

Definition at line 66 of file `acc.c`.

References `COND_INIT`, `COND_T`, `MUTEX_INIT`, and `MUTEX_T`.

Referenced by `mc_platform_Initialize()`.

#### 8.1.2.3 void\* acc\_MessageHandlerThread (void \* arg)

Definition at line 94 of file `acc.c`.

References `agent_Initialize()`, `mc_platform_s::agent_queue`, `agent_s::agent_status`, `AGENT_UPDATE`, `mc_platform_s::ams`, `mc_platform_s::asm_message_queue`, `CANCEL`, `COND_BROADCAST`, `COND_-SIGNAL`, `COND_WAIT`, `agent_s::datastate`, `ENCRYPTED_DATA`, `ENCRYPTION_INITIALIZE`, `FIPA_-ACL`, `message_s::from_address`, `mc_platform_s::giant`, `mc_platform_s::giant_cond`, `mc_platform_s::giant_lock`, `agent_s::lock`, `MC_AGENT_NEUTRAL`, `mc_platform`, `MC_RECV_AGENT`, `MC_-RECV_MESSAGE`, `MC_RECV_RETURN`, `mc_platform_s::MC_signal`, `mc_platform_s::MC_signal_-cond`, `mc_platform_s::MC_signal_lock`, `message_Destroy()`, `mc_platform_s::message_queue`, `message_-Send()`, `message_s::message_type`, `MOBILE_AGENT`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `N_-UNDRSTD`, `agent_s::name`, `agent_datastate_s::persistent`, `QUER_IF`, `QUER_REF`, `mc_platform_s::quit`, `mc_platform_s::quit_lock`, `RELAY`, `REQUEST`, `REQUEST_ENCRYPTION_INITIALIZE`, `RETURN_-MSG`, `SUBSCRIBE`, `message_s::to_address`, and `WARN`.

Referenced by `acc_Start()`.

#### 8.1.2.4 void acc\_Start (mc\_platform\_p mc\_platform)

Definition at line 540 of file acc.c.

References mc\_platform\_s::acc, acc\_MessageHandlerThread(), acc\_Thread(), listen\_Thread(), MC\_THREAD\_ACC, mc\_platform\_s::stack\_size, and THREAD\_CREATE.

Referenced by mc\_platform\_Initialize().

#### 8.1.2.5 void\* acc\_Thread (void \* arg)

Definition at line 328 of file acc.c.

References agent\_mailbox\_Post(), mc\_platform\_s::agent\_queue, AGENT\_UPDATE, mc\_platform\_s::asm\_message\_queue, CANCEL, COND\_BROADCAST, COND\_WAIT, connection\_Destroy(), mc\_platform\_s::connection\_queue, mtp\_http\_s::content, mtp\_http\_content\_s::data, ENCRYPTED\_DATA, ENCRYPTION\_INITIALIZE, FIPA\_ACL, fipa\_acl\_envelope\_Destroy(), fipa\_acl\_envelope\_New(), fipa\_acl\_message\_Destroy(), fipa\_acl\_message\_New(), fipa\_acl\_Parse(), fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_envelope\_Parse(), fipa\_message\_string\_Destroy(), fipa\_message\_string\_New(), mc\_platform\_s::giant, mc\_platform\_s::giant\_cond, mc\_platform\_s::giant\_lock, mtp\_http\_s::http\_performative, HTTP\_POST, HTTP\_PUT, agent\_s::mailbox, mc\_platform, MC\_RECV\_CONNECTION, mc\_platform\_s::MC\_signal, mc\_platform\_s::MC\_signal\_cond, mc\_platform\_s::MC\_signal\_lock, fipa\_message\_string\_s::message, message\_s::message\_body, message\_Destroy(), message\_New(), mtp\_http\_s::message\_parts, mc\_platform\_s::message\_queue, message\_s::message\_type, message\_xml\_parse(), MOBILE\_AGENT, mtp\_http\_Destroy(), mtp\_http\_InitializeFromConnection(), mtp\_http\_New(), MUTEX\_LOCK, MUTEX\_UNLOCK, MXML\_NO\_CALLBACK, mxxmlLoadString(), N\_UNDRSTD, fipa\_agent\_identifier\_s::name, fipa\_agent\_identifier\_set\_s::num, fipa\_acl\_envelope\_s::num\_params, fipa\_acl\_envelope\_s::params, fipa\_message\_string\_s::parse, QUER\_IF, QUER\_REF, mc\_platform\_s::quit, mc\_platform\_s::quit\_lock, RELAY, REQUEST, REQUEST\_ENCRYPTION\_INITIALIZE, RETURN\_MSG, SUBSCRIBE, mtp\_http\_s::target, fipa\_acl\_Param\_s::to, WARN, and message\_s::xml\_root.

Referenced by acc\_Start().

#### 8.1.2.6 void\* listen\_Thread (void \* arg)

Definition at line 583 of file acc.c.

References mc\_platform\_s::acc, BACKLOG, CHECK\_NULL, COND\_BROADCAST, mc\_platform\_s::connection\_queue, mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, and mc\_platform\_s::port.

Referenced by acc\_Start().

## 8.2 /home/dko/projects/mobilec/trunk/src/agent.c File Reference

```
#include <unistd.h>
#include <embedch.h>
#include "include/libmc.h"
#include "include/agent.h"
#include "include/mc_platform.h"
#include "include/message.h"
#include "include/agent_lib.h"
#include "include/interpreter_variable_data.h"
#include "include/xml_parser.h"
```

### Functions

- [int agent\\_AddPersistentVariable](#) ([agent\\_p](#) agent, [int](#) task\_num, const char \*var\_name)
- [agent\\_p agent\\_Copy](#) (const [agent\\_p](#) agent)
- [agent\\_p agent\\_New](#) (void)
- [agent\\_p agent\\_Initialize](#) (struct [mc\\_platform\\_s](#) \*mc\_platform, [message\\_p](#) message, [int](#) id)
- [int agent\\_Destroy](#) ([agent\\_p](#) agent)
- void [agent\\_RunChScript](#) ([agent\\_p](#) agent, [mc\\_platform\\_p](#) mc\_platform)
- void \* [agent\\_RunChScriptThread](#) (void \*ChAgent)

### 8.2.1 Function Documentation

#### 8.2.1.1 [int agent\\_AddPersistentVariable](#) ([agent\\_p](#) agent, [int](#) task\_num, const char \* var\_name)

Definition at line 49 of file agent.c.

References [agent\\_s::agent\\_interp](#), [agent\\_task\\_s::agent\\_variable\\_list](#), [interpreter\\_variable\\_data\\_s::array\\_dim](#), [interpreter\\_variable\\_data\\_s::array\\_extent](#), [CH\\_DATATYPE\\_SIZE](#), [CHECK\\_NULL](#), [interpreter\\_variable\\_data\\_s::data](#), [interpreter\\_variable\\_data\\_s::data\\_type](#), [agent\\_s::datastate](#), [MC\\_ERR](#), [agent\\_s::name](#), [interpreter\\_variable\\_data\\_s::name](#), [interpreter\\_variable\\_data\\_s::size](#), [size](#), [agent\\_datastate\\_s::task\\_progress](#), and [agent\\_datastate\\_s::tasks](#).

#### 8.2.1.2 [agent\\_p agent\\_Copy](#) (const [agent\\_p](#) agent)

Definition at line 131 of file agent.c.

References [agent\\_datastate\\_Copy\(\)](#), [agent\\_s::agent\\_interp](#), [agent\\_s::agent\\_persistent](#), [agent\\_s::agent\\_status](#), [agent\\_s::agent\\_thread](#), [agent\\_s::agent\\_type](#), [agent\\_s::arrival\\_time](#), [agent\\_s::datastate](#), [agent\\_s::home](#), [agent\\_s::home\\_port](#), [agent\\_s::id](#), [agent\\_s::lock](#), [MUTEX\\_INIT](#), [MUTEX\\_LOCK](#), [MUTEX\\_T](#), [agent\\_s::name](#), [agent\\_s::orphan](#), [agent\\_s::owner](#), [agent\\_s::return\\_data](#), and [agent\\_s::run\\_lock](#).

Referenced by [MC\\_CopyAgent\(\)](#).

#### 8.2.1.3 [int agent\\_Destroy](#) ([agent\\_p](#) agent)

Definition at line 322 of file agent.c.

References agent\_datastate\_Destroy(), agent\_s::agent\_interp, agent\_mailbox\_Destroy(), agent\_s::agent\_status, agent\_s::agent\_thread, agent\_s::datastate, agent\_s::home, agent\_s::lock, agent\_s::mailbox, MC\_AGENT\_NEUTRAL, MC\_SUCCESS, MUTEX\_DESTROY, MUTEX\_LOCK, agent\_s::name, agent\_s::owner, and agent\_s::run\_lock.

Referenced by agent\_Initialize().

#### 8.2.1.4 agent\_p agent\_Initialize (struct mc\_platform\_s \* mc\_platform, message\_p message, int id)

Definition at line 216 of file agent.c.

References agent\_datastate\_New(), agent\_Destroy(), agent\_mailbox\_New(), agent\_s::agent\_pipe\_active, agent\_s::agent\_pipe\_ready\_to\_read, agent\_s::agent\_ready\_to\_send, agent\_s::agent\_script\_ready, agent\_s::agent\_status, agent\_s::agent\_thread, agent\_s::agent\_thread\_id, agent\_s::agent\_type, message\_s::agent\_xml\_flag, agent\_xml\_parse(), agent\_s::arrival\_time, agent\_s::datastate, mc\_platform\_s::default\_agentstatus, mc\_platform\_s::err, agent\_s::id, agent\_s::lock, agent\_s::mailbox, MC\_ERR\_PARSE, agent\_s::mc\_platform, MC\_REMOTE\_AGENT, MC\_RETURN\_AGENT, MC\_WAIT\_CH, message\_s::message\_type, MOBILE\_AGENT, MUTEX\_DESTROY, MUTEX\_INIT, MUTEX\_T, agent\_s::orphan, RETURN\_MSG, agent\_s::run\_lock, agent\_datastate\_s::xml\_agent\_root, message\_s::xml\_payload, message\_s::xml\_root, and agent\_datastate\_s::xml\_root.

Referenced by acc\_MessageHandlerThread().

#### 8.2.1.5 agent\_p agent\_New (void)

Definition at line 193 of file agent.c.

References agent\_s::lock, MUTEX\_INIT, MUTEX\_NEW, and agent\_s::run\_lock.

Referenced by MC\_ComposeAgent().

#### 8.2.1.6 void agent\_RunChScript (agent\_p agent, mc\_platform\_p mc\_platform)

Definition at line 356 of file agent.c.

References agent\_RunChScriptThread(), agent\_s::agent\_status, agent\_s::agent\_thread, MC\_AGENT\_ACTIVE, agent\_s::mc\_platform, MC\_THREAD\_AGENT, mc\_platform\_s::stack\_size, THREAD\_CREATE, and THREAD\_T.

Referenced by ams\_ManageAgentList().

#### 8.2.1.7 void\* agent\_RunChScriptThread (void \* ChAgent)

Definition at line 397 of file agent.c.

References agent\_s::agent\_interp, mc\_platform\_s::ams, COND\_SIGNAL, agent\_s::datastate, agent\_s::id, mc\_platform\_s::interp\_options, interpreter\_variable\_data\_Destroy(), interpreter\_variable\_data\_Initialize(), interpreter\_variable\_data\_InitializeFromAgent(), MC\_AclAddReceiver\_chdl(), MC\_AclAddReplyTo\_chdl(), MC\_AclDestroy\_chdl(), MC\_AclNew\_chdl(), MC\_AclPost\_chdl(), MC\_AclReply\_chdl(), MC\_AclRetrieve\_chdl(), MC\_AclSend\_chdl(), MC\_AclSetContent\_chdl(), MC\_AclSetPerformative\_chdl(), MC\_AclSetSender\_chdl(), MC\_AclWaitRetrieve\_chdl(), MC\_AddAgent\_chdl(), MC\_AGENT\_NEUTRAL, MC\_AgentVariableRetrieve\_chdl(), MC\_AgentVariableSave\_chdl(), MC\_Barrier\_chdl(), MC\_BarrierDelete\_chdl(), MC\_BarrierInit\_chdl(), MC\_CallAgentFunc\_chdl(), MC\_ComposeAgent\_chdl(), MC\_CondBroadcast\_chdl(), MC\_CondReset\_chdl(), MC\_CondSignal\_chdl(), MC\_CondWait\_chdl(), MC\_DeleteAgent\_chdl(), MC\_

DeregisterService\_chdl(), MC\_DestroyServiceSearchResult\_chdl(), MC\_End\_chdl(), MC\_EXEC\_AGENT, MC\_FindAgentByID\_chdl(), MC\_FindAgentByName\_chdl(), MC\_GetAgentID\_chdl(), MC\_GetAgentName\_chdl(), MC\_GetAgentStatus\_chdl(), MC\_GetAgentXMLString\_chdl(), MC\_GetTimeOfDay\_chdl(), MC\_HaltAgency\_chdl(), MC\_MutexLock\_chdl(), MC\_MutexUnlock\_chdl(), agent\_s::mc\_platform, mc\_platform, MC\_PrintAgentCode\_chdl(), MC\_RegisterService\_chdl(), MC\_ResumeAgency\_chdl(), MC\_RetrieveAgent\_chdl(), MC\_RetrieveAgentCode\_chdl(), MC\_RETURN\_AGENT, MC\_SearchForService\_chdl(), MC\_SemaphorePost\_chdl(), MC\_SemaphoreWait\_chdl(), MC\_SendAgentMigrationMessage\_chdl(), MC\_SendAgentMigrationMessageFile\_chdl(), MC\_SendSteerCommand\_chdl(), MC\_SetAgentStatus\_chdl(), MC\_SetDefaultAgentStatus\_chdl(), mc\_platform\_s::MC\_signal, mc\_platform\_s::MC\_signal\_cond, mc\_platform\_s::MC\_signal\_lock, MC\_SyncDelete\_chdl(), MC\_SyncInit\_chdl(), MC\_TerminateAgent\_chdl(), MC\_WAIT\_FINISHED, MC\_WAIT\_MESSGSEND, MUTEX\_LOCK, MUTEX\_UNLOCK, agent\_s::name, SIGNAL, and agent\_datastate\_s::task\_progress.

Referenced by agent\_RunChScript().



## 8.3 /home/dko/projects/mobilec/trunk/src/agent\_datastate.c File Reference

```
#include <mxml.h>
#include "include/agent_datastate.h"
#include "include/macros.h"
#include "include/mc_error.h"
```

### Functions

- [agent\\_datastate\\_p agent\\_datastate\\_Copy](#) (const [agent\\_datastate\\_p](#) *datastate*)
- [agent\\_datastate\\_p agent\\_datastate\\_New](#) (void)
- [int agent\\_datastate\\_Destroy](#) ([agent\\_datastate\\_p](#) *agent\_datastate*)

### 8.3.1 Function Documentation

#### 8.3.1.1 [agent\\_datastate\\_p agent\\_datastate\\_Copy](#) (const [agent\\_datastate\\_p](#) *datastate*)

Definition at line 41 of file `agent_datastate.c`.

References `agent_datastate_s::agent_code`, `agent_datastate_s::agent_code_ids`, `agent_datastate_s::agent_codes`, `agent_datastate_New()`, `agent_task_Copy()`, `agent_datastate_s::init_agent_status`, `agent_datastate_s::number_of_tasks`, `agent_datastate_s::persistent`, `agent_datastate_s::return_data`, `agent_datastate_s::task_progress`, and `agent_datastate_s::tasks`.

Referenced by `agent_Copy()`.

#### 8.3.1.2 [int agent\\_datastate\\_Destroy](#) ([agent\\_datastate\\_p](#) *agent\_datastate*)

Definition at line 129 of file `agent_datastate.c`.

References `agent_datastate_s::agent_code_ids`, `agent_datastate_s::agent_codes`, `agent_task_Destroy()`, `MC_SUCCESS`, `mxmlDelete()`, `agent_datastate_s::number_of_tasks`, `agent_datastate_s::tasks`, and `agent_datastate_s::xml_root`.

Referenced by `agent_Destroy()`.

#### 8.3.1.3 [agent\\_datastate\\_p agent\\_datastate\\_New](#) (void)

Definition at line 109 of file `agent_datastate.c`.

References `agent_datastate_s::agent_code`, `CHECK_NULL`, `agent_datastate_s::init_agent_status`, `agent_datastate_s::number_of_tasks`, `agent_datastate_s::persistent`, `agent_datastate_s::return_data`, `agent_datastate_s::task_progress`, `agent_datastate_s::tasks`, `agent_datastate_s::xml_agent_root`, and `agent_datastate_s::xml_root`.

Referenced by `agent_datastate_Copy()`, `agent_Initialize()`, and `MC_ComposeAgent()`.

## 8.4 /home/dko/projects/mobilec/trunk/src/agent\_mailbox.c File Reference

```
#include "include/data_structures.h"
```

### Functions

- [agent\\_mailbox\\_p agent\\_mailbox\\_New](#) (void)
- [agent\\_mailbox\\_p agent\\_mailbox\\_Copy](#) (agent\_mailbox\_p src)
- [int agent\\_mailbox\\_Destroy](#) (agent\_mailbox\_t \*mailbox)
- [int agent\\_mailbox\\_Post](#) (agent\_mailbox\_p mailbox, fipa\_acl\_message\_t \*message)
- [fipa\\_acl\\_message\\_t \\* agent\\_mailbox\\_Retrieve](#) (agent\_mailbox\_p mailbox)
- [fipa\\_acl\\_message\\_t \\* agent\\_mailbox\\_WaitRetrieve](#) (agent\_mailbox\_p mailbox)

### 8.4.1 Function Documentation

#### 8.4.1.1 agent\_mailbox\_p agent\_mailbox\_Copy (agent\_mailbox\_p src)

Definition at line 15 of file agent\_mailbox.c.

References agent\_mailbox\_New(), and agent\_mailbox\_s::mail\_queue.

#### 8.4.1.2 int agent\_mailbox\_Destroy (agent\_mailbox\_t \* mailbox)

Definition at line 23 of file agent\_mailbox.c.

References agent\_mailbox\_s::mail\_queue.

Referenced by agent\_Destroy().

#### 8.4.1.3 agent\_mailbox\_p agent\_mailbox\_New (void)

Definition at line 6 of file agent\_mailbox.c.

References agent\_mailbox\_s::mail\_queue.

Referenced by agent\_Initialize(), and agent\_mailbox\_Copy().

#### 8.4.1.4 int agent\_mailbox\_Post (agent\_mailbox\_p mailbox, fipa\_acl\_message\_t \* message)

Definition at line 32 of file agent\_mailbox.c.

References agent\_mailbox\_s::mail\_queue.

Referenced by acc\_Thread(), and MC\_AclPost().

#### 8.4.1.5 fipa\_acl\_message\_t\* agent\_mailbox\_Retrieve (agent\_mailbox\_p mailbox)

Definition at line 38 of file agent\_mailbox.c.

References agent\_mailbox\_s::mail\_queue.

Referenced by agent\_mailbox\_WaitRetrieve(), and MC\_AclRetrieve().

**8.4.1.6 fipa\_acl\_message\_t\* agent\_mailbox\_WaitRetrieve (agent\_mailbox\_p *mailbox*)**

Definition at line 43 of file agent\_mailbox.c.

References agent\_mailbox\_Retrieve(), COND\_WAIT, agent\_mailbox\_s::mail\_queue, MUTEX\_LOCK, and MUTEX\_UNLOCK.

Referenced by MC\_AclWaitRetrieve().

## 8.5 /home/dko/projects/mobilec/trunk/src/agent\_return\_data.c File Reference

```
#include "include/interpreter_variable_data.h"
#include "include/agent.h"
```

### Functions

- [interpreter\\_variable\\_data\\_p interpreter\\_variable\\_data\\_New](#) (void)
- [interpreter\\_variable\\_data\\_p interpreter\\_variable\\_data\\_InitializeFromAgent](#) (agent\_p agent)
- [interpreter\\_variable\\_data\\_p interpreter\\_variable\\_data\\_Initialize](#) (agent\_p agent, const char \*varname)
- [int interpreter\\_variable\\_data\\_Destroy](#) (interpreter\_variable\_data\_p agent\_variable\_data)
- [interpreter\\_variable\\_data\\_p interpreter\\_variable\\_data\\_Copy](#) (interpreter\_variable\_data\_p src)

### 8.5.1 Function Documentation

#### 8.5.1.1 [interpreter\\_variable\\_data\\_p interpreter\\_variable\\_data\\_Copy](#) (interpreter\_variable\_data\_p src)

Definition at line 229 of file agent\_return\_data.c.

References [interpreter\\_variable\\_data\\_s::array\\_dim](#), [interpreter\\_variable\\_data\\_s::array\\_extent](#), [interpreter\\_variable\\_data\\_s::data](#), [interpreter\\_variable\\_data\\_s::data\\_type](#), [interpreter\\_variable\\_data\\_New\(\)](#), [interpreter\\_variable\\_data\\_s::name](#), and [interpreter\\_variable\\_data\\_s::size](#).

#### 8.5.1.2 [int interpreter\\_variable\\_data\\_Destroy](#) (interpreter\_variable\_data\_p agent\_variable\_data)

Definition at line 210 of file agent\_return\_data.c.

References [interpreter\\_variable\\_data\\_s::array\\_extent](#), [interpreter\\_variable\\_data\\_s::data](#), [MC\\_SUCCESS](#), and [interpreter\\_variable\\_data\\_s::name](#).

Referenced by [agent\\_RunChScriptThread\(\)](#), and [agent\\_task\\_Destroy\(\)](#).

#### 8.5.1.3 [interpreter\\_variable\\_data\\_p interpreter\\_variable\\_data\\_Initialize](#) (agent\_p agent, const char \* varname)

Definition at line 139 of file agent\_return\_data.c.

References [agent\\_s::agent\\_interp](#), [interpreter\\_variable\\_data\\_s::array\\_dim](#), [interpreter\\_variable\\_data\\_s::array\\_extent](#), [CH\\_DATATYPE\\_SIZE](#), [CHECK\\_NULL](#), [interpreter\\_variable\\_data\\_s::data](#), [interpreter\\_variable\\_data\\_s::data\\_type](#), [MUTEX\\_LOCK](#), [MUTEX\\_UNLOCK](#), [interpreter\\_variable\\_data\\_s::name](#), [agent\\_s::run\\_lock](#), [interpreter\\_variable\\_data\\_s::size](#), and [size](#).

Referenced by [agent\\_RunChScriptThread\(\)](#).

#### 8.5.1.4 [interpreter\\_variable\\_data\\_p interpreter\\_variable\\_data\\_InitializeFromAgent](#) (agent\_p agent)

Definition at line 55 of file agent\_return\_data.c.

References agent\_s::agent\_interp, interpreter\_variable\_data\_s::array\_dim, interpreter\_variable\_data\_s::array\_extent, CH\_DATATYPE\_SIZE, CHECK\_NULL, interpreter\_variable\_data\_s::data, interpreter\_variable\_data\_s::data\_type, agent\_s::datastate, interpreter\_variable\_data\_s::name, interpreter\_variable\_data\_s::size, size, agent\_datastate\_s::task\_progress, agent\_datastate\_s::tasks, and agent\_task\_s::var\_name.

Referenced by agent\_RunChScriptThread().

#### 8.5.1.5 interpreter\_variable\_data\_p interpreter\_variable\_data\_New (void)

Definition at line 40 of file agent\_return\_data.c.

References interpreter\_variable\_data\_s::array\_dim, interpreter\_variable\_data\_s::array\_extent, CHECK\_NULL, interpreter\_variable\_data\_s::data, interpreter\_variable\_data\_s::data\_type, interpreter\_variable\_data\_s::name, and interpreter\_variable\_data\_s::size.

Referenced by agent\_xml\_parse\_\_data(), and interpreter\_variable\_data\_Copy().

## 8.6 /home/dko/projects/mobilec/trunk/src/agent\_task.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "include/agent_task.h"
#include "include/mc_error.h"
```

### Functions

- [agent\\_task\\_p agent\\_task\\_New](#) (void)
- [agent\\_task\\_p agent\\_task\\_Copy](#) ([agent\\_task\\_p](#) task)
- [int agent\\_task\\_Destroy](#) ([agent\\_task\\_p](#) agent\_task)

### 8.6.1 Function Documentation

#### 8.6.1.1 [agent\\_task\\_p agent\\_task\\_Copy](#) ([agent\\_task\\_p](#) task)

Definition at line 60 of file `agent_task.c`.

References `agent_task_s::agent_return_data`, `agent_task_s::code_id`, `agent_task_s::init_agent_status`, `agent_task_s::number_of_elements`, `agent_task_s::persistent`, `agent_task_s::server_name`, `agent_task_s::size_of_element_array`, `agent_task_s::task_completed`, and `agent_task_s::var_name`.

Referenced by `agent_datastate_Copy()`.

#### 8.6.1.2 [int agent\\_task\\_Destroy](#) ([agent\\_task\\_p](#) agent\_task)

Definition at line 102 of file `agent_task.c`.

References `agent_task_s::agent_return_data`, `agent_task_s::agent_variable_list`, `agent_task_s::code_id`, `interpreter_variable_data_Destroy()`, `MC_SUCCESS`, `agent_task_s::saved_variables`, `agent_task_s::server_name`, and `agent_task_s::var_name`.

Referenced by `agent_datastate_Destroy()`.

#### 8.6.1.3 [agent\\_task\\_p agent\\_task\\_New](#) (void)

Definition at line 42 of file `agent_task.c`.

References `agent_task_s::agent_variable_list`, `agent_task_s::num_saved_variables`, and `agent_task_s::saved_variables`.

Referenced by `agent_xml_parse__tasks()`, and `MC_ComposeAgent()`.

## 8.7 /home/dko/projects/mobilec/trunk/src/ams.c File Reference

```
#include "include/ams.h"
#include "include/agent.h"
#include "include/data_structures.h"
#include "include/mc_platform.h"
```

### Functions

- [int ams\\_Destroy](#) (ams\_p ams)
- ams\_p [ams\\_Initialize](#) (mc\_platform\_p mc\_platform)
- void [ams\\_Print](#) (ams\_p ams)
- [int ams\\_ManageAgentList](#) (ams\_p ams)
- void [ams\\_Start](#) (mc\_platform\_p mc\_platform)
- void \* [ams\\_Thread](#) (void \*arg)

### 8.7.1 Function Documentation

#### 8.7.1.1 int ams\_Destroy (ams\_p ams)

Definition at line 41 of file ams.c.

References COND\_DESTROY, MC\_SUCCESS, and MUTEX\_DESTROY.

Referenced by mc\_platform\_Destroy().

#### 8.7.1.2 ams\_p ams\_Initialize (mc\_platform\_p mc\_platform)

Definition at line 52 of file ams.c.

References CHECK\_NULL, COND\_INIT, COND\_T, MUTEX\_INIT, and MUTEX\_T.

Referenced by mc\_platform\_Initialize().

#### 8.7.1.3 int ams\_ManageAgentList (ams\_p ams)

Definition at line 111 of file ams.c.

References [agent\\_RunChScript\(\)](#), [agent\\_s::agent\\_status](#), [ListSearch\(\)](#), [agent\\_s::lock](#), MC\_AGENT\_ACTIVE, MC\_AGENT\_NEUTRAL, MC\_WAIT\_CH, MC\_WAIT\_FINISHED, MC\_WAIT\_MESSGSEND, [message\\_Destroy\(\)](#), [message\\_InitializeFromAgent\(\)](#), [message\\_New\(\)](#), MUTEX\_LOCK, MUTEX\_UNLOCK, and [agent\\_s::orphan](#).

Referenced by [ams\\_Thread\(\)](#).

#### 8.7.1.4 void ams\_Print (ams\_p ams)

Definition at line 79 of file ams.c.

References [agent\\_s::agent\\_status](#), [agent\\_s::connect\\_id](#), [agent\\_s::id](#), [ListSearch\(\)](#), MUTEX\_LOCK, and MUTEX\_UNLOCK.

**8.7.1.5 void `ams_Start` (`mc_platform_p` *mc\_platform*)**

Definition at line 195 of file `ams.c`.

References `mc_platform_s::ams`, `ams_Thread()`, `MC_THREAD_AMS`, `mc_platform_s::stack_size`, and `THREAD_CREATE`.

Referenced by `mc_platform_Initialize()`.

**8.7.1.6 void\* `ams_Thread` (`void *` *arg*)**

Definition at line 225 of file `ams.c`.

References `mc_platform_s::ams`, `ams_ManageAgentList()`, `COND_BROADCAST`, `COND_WAIT`, `mc_platform`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `mc_platform_s::quit`, and `mc_platform_s::quit_lock`.

Referenced by `ams_Start()`.



## 8.8 /home/dko/projects/mobilec/trunk/src/barrier.c File Reference

```
#include "include/barrier.h"
#include "include/mc_error.h"
```

### Functions

- [barrier\\_node\\_p barrier\\_node\\_Initialize](#) ([int id](#), [int num\\_registered](#))
- [int barrier\\_node\\_Destroy](#) ([barrier\\_node\\_p node](#))
- [int barrier\\_queue\\_Add](#) ([barrier\\_queue\\_p list](#), [barrier\\_node\\_p node](#))
- [int barrier\\_queue\\_Delete](#) ([int id](#), [barrier\\_queue\\_p list](#))
- [int barrier\\_queue\\_Destroy](#) ([barrier\\_queue\\_p queue](#))
- [barrier\\_node\\_p barrier\\_queue\\_Get](#) ([barrier\\_queue\\_p list](#), [int id](#))
- [barrier\\_queue\\_p barrier\\_queue\\_New](#) ([void](#))
- [barrier\\_node\\_p barrier\\_queue\\_Pop](#) ([barrier\\_queue\\_p queue](#))

### 8.8.1 Function Documentation

#### 8.8.1.1 [int barrier\\_node\\_Destroy](#) ([barrier\\_node\\_p node](#))

Definition at line 65 of file barrier.c.

References [barrier\\_node\\_s::cond](#), [COND\\_DESTROY](#), [barrier\\_node\\_s::lock](#), [MC\\_SUCCESS](#), and [MUTEX\\_DESTROY](#).

Referenced by [barrier\\_queue\\_Delete\(\)](#), and [barrier\\_queue\\_Destroy\(\)](#).

#### 8.8.1.2 [barrier\\_node\\_p barrier\\_node\\_Initialize](#) ([int id](#), [int num\\_registered](#))

Definition at line 40 of file barrier.c.

References [CHECK\\_NULL](#), [barrier\\_node\\_s::cond](#), [COND\\_INIT](#), [COND\\_T](#), [barrier\\_node\\_s::id](#), [barrier\\_node\\_s::lock](#), [MUTEX\\_INIT](#), [MUTEX\\_T](#), [node](#), [barrier\\_node\\_s::num\\_registered](#), and [barrier\\_node\\_s::num\\_waiting](#).

Referenced by [MC\\_BarrierInit\(\)](#).

#### 8.8.1.3 [int barrier\\_queue\\_Add](#) ([barrier\\_queue\\_p list](#), [barrier\\_node\\_p node](#))

Definition at line 82 of file barrier.c.

References [DATA](#), [barrier\\_node\\_s::id](#), [barrier\\_queue\\_s::list](#), [ListAdd\(\)](#), [list\\_s::listhead](#), [barrier\\_queue\\_s::lock](#), [MC\\_SUCCESS](#), [MC\\_WARN\\_DUPLICATE](#), [listNode\\_s::next](#), [listNode\\_s::node\\_data](#), [RWLOCK\\_WRLock](#), [RWLOCK\\_WRunLock](#), and [barrier\\_queue\\_s::size](#).

Referenced by [MC\\_BarrierInit\(\)](#).

#### 8.8.1.4 [int barrier\\_queue\\_Delete](#) ([int id](#), [barrier\\_queue\\_p list](#))

Definition at line 106 of file barrier.c.

References `barrier_node_Destroy()`, `barrier_node_s::id`, `barrier_queue_s::list`, `ListDelete()`, `ListSearch()`, `barrier_queue_s::lock`, `MC_ERR_NOT_FOUND`, `MC_SUCCESS`, `RWLOCK_WRLOCK`, `RWLOCK_WRUNLOCK`, `barrier_queue_s::size`, and `list_s::size`.

Referenced by `MC_BarrierDelete()`.

#### **8.8.1.5 `int barrier_queue_Destroy (barrier_queue_p queue)`**

Definition at line 126 of file `barrier.c`.

References `barrier_node_Destroy()`, `barrier_queue_Pop()`, `barrier_queue_s::list`, `ListTerminate()`, `barrier_queue_s::lock`, `MC_SUCCESS`, `node`, and `RWLOCK_DESTROY`.

Referenced by `mc_platform_Destroy()`.

#### **8.8.1.6 `barrier_node_p barrier_queue_Get (barrier_queue_p list, int id)`**

Definition at line 140 of file `barrier.c`.

References `barrier_queue_s::list`, `list_s::listhead`, `barrier_queue_s::lock`, `listNode_s::next`, `listNode_s::node_data`, `RWLOCK_RDLOCK`, and `RWLOCK_RDUNLOCK`.

Referenced by `MC_Barrier()`, and `MC_BarrierInit()`.

#### **8.8.1.7 `barrier_queue_p barrier_queue_New (void)`**

Definition at line 157 of file `barrier.c`.

References `CHECK_NULL`, `barrier_queue_s::list`, `ListInitialize()`, `barrier_queue_s::lock`, `RWLOCK_INIT`, and `RWLOCK_T`.

Referenced by `mc_platform_Initialize()`.

#### **8.8.1.8 `barrier_node_p barrier_queue_Pop (barrier_queue_p queue)`**

Definition at line 171 of file `barrier.c`.

References `barrier_queue_s::list`, `ListPop()`, and `node`.

Referenced by `barrier_queue_Destroy()`.

## 8.9 /home/dko/projects/mobilec/trunk/src/cmd\_prompt.c File Reference

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include "include/cmd_prompt.h"
#include "include/commands.h"
#include "config.h"
```

### Functions

- [cmd\\_prompt\\_p cmd\\_prompt\\_Initialize \(mc\\_platform\\_p mc\\_platform\)](#)
- [int cmd\\_prompt\\_Destroy \(cmd\\_prompt\\_p cmd\\_prompt\)](#)
- [void cmd\\_prompt\\_Start \(mc\\_platform\\_p mc\\_platform\)](#)
- [void \\* cmd\\_prompt\\_Thread \(void \\*arg\)](#)
- [int split\\_string \(char \\*\\*\\*args, const char \\*buf\)](#)
- [int process\\_command \(command\\_t \\*cmd\)](#)
- [int exec\\_command \(command\\_t cmd, mc\\_platform\\_p global\)](#)
- [int dealloc\\_command \(command\\_t \\*cmd\)](#)
- [int handler\\_QUIT \(void \\*arg, mc\\_platform\\_p global\)](#)
- [int handler\\_HELP \(void \\*arg, mc\\_platform\\_p global\)](#)
- [int handler\\_SEND \(void \\*arg, mc\\_platform\\_p global\)](#)
- [int handler\\_PRINT\\_CONNECTLIST \(void \\*arg, mc\\_platform\\_p global\)](#)
- [int handler\\_PRINTLIST\\_MESSAGE \(void \\*arg, mc\\_platform\\_p global\)](#)
- [int handler\\_PRINTLIST\\_AGENTS \(void \\*arg, mc\\_platform\\_p global\)](#)

### 8.9.1 Function Documentation

#### 8.9.1.1 [int cmd\\_prompt\\_Destroy \(cmd\\_prompt\\_p \*cmd\\_prompt\*\)](#)

Definition at line 63 of file `cmd_prompt.c`.

References `MC_SUCCESS`.

Referenced by `mc_platform_Destroy()`.

#### 8.9.1.2 [cmd\\_prompt\\_p cmd\\_prompt\\_Initialize \(mc\\_platform\\_p \*mc\\_platform\*\)](#)

Definition at line 55 of file `cmd_prompt.c`.

Referenced by `mc_platform_Initialize()`.

**8.9.1.3 void cmd\_prompt\_Start (mc\_platform\_p mc\_platform)**

Definition at line 70 of file cmd\_prompt.c.

References mc\_platform\_s::cmd\_prompt, cmd\_prompt\_Thread(), MC\_THREAD\_CP, mc\_platform\_s::stack\_size, cmd\_prompt\_s::thread, and THREAD\_CREATE.

Referenced by mc\_platform\_Initialize().

**8.9.1.4 void\* cmd\_prompt\_Thread (void \* arg)**

Definition at line 101 of file cmd\_prompt.c.

References command\_s::args, dealloc\_command(), exec\_command(), command\_s::index, mc\_platform, command\_s::num\_args, process\_command(), and split\_string().

Referenced by cmd\_prompt\_Start().

**8.9.1.5 int dealloc\_command (command\_t \* cmd)**

Definition at line 228 of file cmd\_prompt.c.

Referenced by cmd\_prompt\_Thread().

**8.9.1.6 int exec\_command (command\_t cmd, mc\_platform\_p global)**

Definition at line 212 of file cmd\_prompt.c.

Referenced by cmd\_prompt\_Thread().

**8.9.1.7 int handler\_HELP (void \* arg, mc\_platform\_p global)**

Definition at line 250 of file cmd\_prompt.c.

**8.9.1.8 int handler\_PRINT\_CONNECTLIST (void \* arg, mc\_platform\_p global)**

Definition at line 303 of file cmd\_prompt.c.

**8.9.1.9 int handler\_PRINTLIST\_AGENTS (void \* arg, mc\_platform\_p global)**

Definition at line 315 of file cmd\_prompt.c.

**8.9.1.10 int handler\_PRINTLIST\_MESSAGE (void \* arg, mc\_platform\_p global)**

Definition at line 309 of file cmd\_prompt.c.

**8.9.1.11 int handler\_QUIT (void \* arg, mc\_platform\_p global)**

Definition at line 243 of file cmd\_prompt.c.

References mc\_platform\_Destroy().

**8.9.1.12 int handler\_SEND (void \* *arg*, mc\_platform\_p *global*)**

Definition at line 288 of file cmd\_prompt.c.

**8.9.1.13 int process\_command (command\_t \* *cmd*)**

Definition at line 194 of file cmd\_prompt.c.

References command\_s::args, command\_cmds, command\_s::index, and command\_s::num\_args.

Referenced by cmd\_prompt\_Thread().

**8.9.1.14 int split\_string (char \*\*\* *args*, const char \* *buf*)**

Definition at line 146 of file cmd\_prompt.c.

References int.

Referenced by cmd\_prompt\_Thread().

## 8.10 /home/dko/projects/mobilec/trunk/src/connection.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "include/connection.h"
#include "include/mc_error.h"
```

### Functions

- [int connection\\_Destroy](#) ([connection\\_p](#) connection)
- [connection\\_p connection\\_New](#) (void)
- [connection\\_p connection\\_Copy](#) ([connection\\_p](#) connection)

#### 8.10.1 Function Documentation

##### 8.10.1.1 [connection\\_p connection\\_Copy](#) ([connection\\_p](#) *connection*)

Definition at line 72 of file connection.c.

References [connection\\_s::addr](#), [connection\\_s::clientfd](#), [connection\\_s::connect\\_id](#), [connection\\_New\(\)](#), [connection\\_s::remote\\_hostname](#), and [connection\\_s::serverfd](#).

##### 8.10.1.2 [int connection\\_Destroy](#) ([connection\\_p](#) *connection*)

Definition at line 42 of file connection.c.

References [connection\\_s::clientfd](#), [MC\\_SUCCESS](#), and [connection\\_s::remote\\_hostname](#).

Referenced by [acc\\_Thread\(\)](#).

##### 8.10.1.3 [connection\\_p connection\\_New](#) (void)

Definition at line 61 of file connection.c.

Referenced by [connection\\_Copy\(\)](#).

## 8.11 /home/dko/projects/mobilec/trunk/src/data\_structures.c File Reference

```
#include "include/ap_queue_template.h"
#include "include/data_structures.h"
#include "include/interpreter_variable_data.h"
#include "include/agent.h"
```

### Functions

- [AP\\_QUEUE\\_SEARCH\\_TEMPLATE](#) (AP\_QUEUE\_REMOVE\_TEMPLATE(connection\_queue, [Search](#), AP\_QUEUE\_REMOVE\_TEMPLATE(connection, AP\_QUEUE\_REMOVE\_TEMPLATE(int, (node → connect\_id==key)))
- [AP\\_QUEUE\\_SEARCH\\_TEMPLATE](#) (AP\_QUEUE\_REMOVE\_TEMPLATE(agent\_variable\_list, AP\_QUEUE\_REMOVE\_TEMPLATE([Search](#), AP\_QUEUE\_REMOVE\_TEMPLATE(interpreter\_variable\_data, char \*, (!strcmp(node → name, key))))
- [AP\\_QUEUE\\_STD\\_DEFN\\_TEMPLATE](#) (AP\_QUEUE\_SEARCH\_TEMPLATE(agent\_queue, AP\_QUEUE\_SEARCH\_TEMPLATE(agent)
- [AP\\_QUEUE\\_STD\\_DEFN\\_TEMPLATE](#) (fipa\_acl\_message\_p mail\_queue\_SearchReceivers(mail\_queue\_p mail\_queue, fipa\_acl\_message)

### 8.11.1 Function Documentation

**8.11.1.1** [AP\\_QUEUE\\_SEARCH\\_TEMPLATE](#) (AP\_QUEUE\_REMOVE\_TEMPLATE(*agent\_variable\_list*, AP\_QUEUE\_REMOVE\_TEMPLATE(*Search*, AP\_QUEUE\_REMOVE\_TEMPLATE(*interpreter\_variable\_data*, char \*, (!strcmp(*node* → *name*, *key*))))

Definition at line 79 of file data\_structures.c.

References [message\\_s::from\\_address](#), [ListSearch\(\)](#), [message\\_s::message\\_id](#), [MUTEX\\_LOCK](#), [MUTEX\\_UNLOCK](#), [node](#), and [message\\_s::to\\_address](#).

**8.11.1.2** [AP\\_QUEUE\\_SEARCH\\_TEMPLATE](#) (AP\_QUEUE\_REMOVE\_TEMPLATE(*connection\_queue*, *Search*, AP\_QUEUE\_REMOVE\_TEMPLATE(*connection*, AP\_QUEUE\_REMOVE\_TEMPLATE(*int*, (*node* → *connect\_id*==key)))

Definition at line 40 of file data\_structures.c.

References [connection\\_s::connect\\_id](#), [ListSearch\(\)](#), [MUTEX\\_LOCK](#), [MUTEX\\_UNLOCK](#), [node](#), and [connection\\_s::remote\\_hostname](#).

**8.11.1.3** [AP\\_QUEUE\\_STD\\_DEFN\\_TEMPLATE](#) (*fipa\_acl\_message\_p* mail\_queue\_SearchReceivers(*mail\_queue\_p* mail\_queue, *fipa\_acl\_message*)

Definition at line 193 of file data\_structures.c.

References [fipa\\_agent\\_identifier\\_set\\_s::fipa\\_agent\\_identifiers](#), [MUTEX\\_LOCK](#), [MUTEX\\_UNLOCK](#), [fipa\\_agent\\_identifier\\_s::name](#), [listNode\\_s::next](#), [node](#), [listNode\\_s::node\\_data](#), and [fipa\\_acl\\_message\\_s::receiver](#).

#### 8.11.1.4 AP\_QUEUE\_STD\_DEFN\_TEMPLATE (AP\_QUEUE\_SEARCH\_TEMPLATE( *agent\_queue*, AP\_QUEUE\_SEARCH\_TEMPLATE(*agent*)

Definition at line 140 of file data\_structures.c.

References `agent_s::id`, `ListSearch()`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `agent_s::name`, and `node`.



## 8.12 /home/dko/projects/mobilec/trunk/src/df.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include "include/mc_platform.h"
#include "include/df.h"
#include "include/df_request.x.h"
```

### Defines

- #define [REQUEST](#)(name, string, description)

### Functions

- [int df\\_Add](#) (struct df\_s \*df, struct df\_node\_s \*[node](#))
- [int df\\_AddRequest](#) (struct df\_s \*df, struct df\_request\_list\_node\_s \*[node](#))
- [int df\\_Destroy](#) (df\_p df)
- [df\\_p df\\_Initialize](#) ([mc\\_platform\\_p](#) mc\_platform)
- [int df\\_ProcessRequest](#) (struct [mc\\_platform\\_s](#) \*global)
- [int df\\_SearchForService](#) (df\_p df, const char \*searchstring, char \*\*\*agent\_names, char \*\*\*service\_names, int \*\*agent\_ids, int \*num\_entries)
- [void df\\_Start](#) ([mc\\_platform\\_p](#) mc\_platform)
- [int df\\_request\\_list\\_node\\_Destroy](#) (df\_request\_list\_node\_p [node](#))
- [df\\_request\\_list\\_node\\_p df\\_request\\_list\\_node\\_New](#) (void)
- [int df\\_request\\_list\\_Destroy](#) (df\_request\_list\_p df\_request\_list)
- [df\\_request\\_list\\_p df\\_request\\_list\\_New](#) (void)
- [df\\_request\\_list\\_node\\_p df\\_request\\_list\\_Pop](#) (df\_request\_list\_p requests)
- [df\\_request\\_search\\_p df\\_request\\_search\\_New](#) (void)
- [int df\\_request\\_search\\_Destroy](#) (df\_request\_search\_p [node](#))
- [int df\\_node\\_Destroy](#) (df\_node\_p df\_node)
- [void \\* df\\_Thread](#) (void \*arg)
- [int request\\_handler\\_REGISTER](#) (struct [mc\\_platform\\_s](#) \*global, void \*data)
- [int request\\_handler\\_SEARCH](#) (struct [mc\\_platform\\_s](#) \*global, void \*data)
- [int request\\_handler\\_SUBSCRIBE](#) (struct [mc\\_platform\\_s](#) \*global, void \*data)
- [int request\\_handler\\_DEREGISTER](#) (struct [mc\\_platform\\_s](#) \*global, void \*data)

### 8.12.1 Define Documentation

#### 8.12.1.1 #define [REQUEST](#)(name, string, description)

##### Value:

```
if ( !strcmp(request->command, string ) ) { \
    return_code = MC_SUCCESS; \
    handler_code = request_handler_##name( \
        global, \
        request->data ); \
    request_code = REQUEST_##name; \
} else
```

Referenced by `acc_MessageHandlerThread()`, and `acc_Thread()`.

## 8.12.2 Function Documentation

### 8.12.2.1 `int df_Add (struct df_s * df, struct df_node_s * node)`

Definition at line 47 of file `df.c`.

References `ListAdd()`, `MC_SUCCESS`, and `SIGNAL`.

Referenced by `request_handler_REGISTER()`.

### 8.12.2.2 `int df_AddRequest (struct df_s * df, struct df_request_list_node_s * node)`

Definition at line 63 of file `df.c`.

References `ListAdd()`, and `SIGNAL`.

Referenced by `MC_DeregisterService()`, `MC_RegisterService()`, and `MC_SearchForService()`.

### 8.12.2.3 `int df_Destroy (df_p df)`

Definition at line 80 of file `df.c`.

References `COND_DESTROY`, `df_node_Destroy()`, `df_request_list_Destroy()`, `ListPop()`, `ListTerminate()`, `MC_SUCCESS`, `MUTEX_DESTROY`, and `MUTEX_LOCK`.

Referenced by `mc_platform_Destroy()`.

### 8.12.2.4 `df_p df_Initialize (mc_platform_p mc_platform)`

Definition at line 98 of file `df.c`.

References `COND_INIT`, `COND_T`, `df_request_list_New()`, `ListInitialize()`, `MUTEX_INIT`, and `MUTEX_T`.

Referenced by `mc_platform_Initialize()`.

### 8.12.2.5 `int df_node_Destroy (df_node_p df_node)`

Definition at line 408 of file `df.c`.

References `MC_SUCCESS`, and `MUTEX_LOCK`.

Referenced by `df_Destroy()`.

### 8.12.2.6 `int df_ProcessRequest (struct mc_platform_s * global)`

Definition at line 129 of file `df.c`.

References `mc_platform_s::df`, `df_request_list_Pop()`, `MC_ERR_EMPTY`, and `MC_ERR_INVALID`.

Referenced by `df_Thread()`.

**8.12.2.7 int df\_request\_list\_Destroy (df\_request\_list\_p *df\_request\_list*)**

Definition at line 317 of file df.c.

References df\_request\_list\_node\_Destroy(), ListPop(), ListTerminate(), MC\_SUCCESS, and node.

Referenced by df\_Destroy().

**8.12.2.8 df\_request\_list\_p df\_request\_list\_New (void)**

Definition at line 339 of file df.c.

References CHECK\_NULL, COND\_INIT, COND\_T, ListInitialize(), MUTEX\_INIT, and MUTEX\_T.

Referenced by df\_Initialize().

**8.12.2.9 int df\_request\_list\_node\_Destroy (df\_request\_list\_node\_p *node*)**

Definition at line 288 of file df.c.

References COND\_DESTROY, MC\_SUCCESS, and MUTEX\_DESTROY.

Referenced by df\_request\_list\_Destroy(), and MC\_SearchForService().

**8.12.2.10 df\_request\_list\_node\_p df\_request\_list\_node\_New (void)**

Definition at line 299 of file df.c.

References CHECK\_NULL, COND\_INIT, COND\_T, MUTEX\_INIT, MUTEX\_T, and node.

Referenced by MC\_DeregisterService(), MC\_RegisterService(), and MC\_SearchForService().

**8.12.2.11 df\_request\_list\_node\_p df\_request\_list\_Pop (df\_request\_list\_p *requests*)**

Definition at line 364 of file df.c.

References ListPop(), MUTEX\_LOCK, MUTEX\_UNLOCK, and node.

Referenced by df\_ProcessRequest().

**8.12.2.12 int df\_request\_search\_Destroy (df\_request\_search\_p *node*)**

Definition at line 395 of file df.c.

References COND\_DESTROY, MC\_SUCCESS, and MUTEX\_DESTROY.

Referenced by MC\_SearchForService().

**8.12.2.13 df\_request\_search\_p df\_request\_search\_New (void)**

Definition at line 380 of file df.c.

References CHECK\_NULL, COND\_INIT, COND\_T, MUTEX\_INIT, MUTEX\_T, and search.

Referenced by MC\_SearchForService().

**8.12.2.14** `int df_SearchForService(df_p df, const char * searchstring, char *** agent_names, char *** service_names, int ** agent_ids, int * num_entries)`

Definition at line 173 of file df.c.

References MC\_ERR\_NOT\_FOUND, MC\_SUCCESS, MUTEX\_LOCK, MUTEX\_UNLOCK, listNode\_s::next, and listNode\_s::node\_data.

**8.12.2.15** `void df_Start(mc_platform_p mc_platform)`

Definition at line 258 of file df.c.

References mc\_platform\_s::df, df\_Thread(), MC\_THREAD\_DF, mc\_platform\_s::stack\_size, and THREAD\_CREATE.

Referenced by mc\_platform\_Initialize().

**8.12.2.16** `void* df_Thread(void * arg)`

Definition at line 422 of file df.c.

References COND\_BROADCAST, COND\_WAIT, mc\_platform\_s::df, df\_ProcessRequest(), MC\_SUCCESS, MUTEX\_LOCK, MUTEX\_UNLOCK, mc\_platform\_s::quit, and mc\_platform\_s::quit\_lock.

Referenced by df\_Start().

**8.12.2.17** `int request_handler_DEREGISTER(struct mc_platform_s * global, void * data)`

Definition at line 515 of file df.c.

**8.12.2.18** `int request_handler_REGISTER(struct mc_platform_s * global, void * data)`

Definition at line 485 of file df.c.

References mc\_platform\_s::df, and df\_Add().

**8.12.2.19** `int request_handler_SEARCH(struct mc_platform_s * global, void * data)`

Definition at line 491 of file df.c.

**8.12.2.20** `int request_handler_SUBSCRIBE(struct mc_platform_s * global, void * data)`

Definition at line 510 of file df.c.

## 8.13 /home/dko/projects/mobilec/trunk/src/dynstring.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "include/dynstring.h"
```

### Functions

- [dynstring\\_t \\* dynstring\\_New](#) (void)
- [int dynstring\\_Append](#) ([dynstring\\_t](#) \*msg, char \*str)
- [int dynstring\\_Destroy](#) ([dynstring\\_t](#) \*dynstring)

### 8.13.1 Function Documentation

#### 8.13.1.1 int dynstring\_Append (dynstring\_t \* msg, char \* str)

Definition at line 24 of file dynstring.c.

References COMPOSE\_BLOCKSIZE, dynstring\_s::len, dynstring\_s::message, and dynstring\_s::size.

Referenced by fipa\_acl\_Compose(), fipa\_agent\_identifier\_Compose(), fipa\_agent\_identifier\_set\_Compose(), fipa\_DateTime\_Compose(), fipa\_envelope\_Compose\_\_from(), fipa\_number\_Compose(), fipa\_performative\_Compose(), fipa\_string\_Compose(), fipa\_url\_Compose(), fipa\_url\_sequence\_Compose(), fipa\_word\_Compose(), and mtp\_http\_CreateMessage().

#### 8.13.1.2 int dynstring\_Destroy (dynstring\_t \* *dynstring*)

Definition at line 49 of file dynstring.c.

References dynstring\_s::message.

Referenced by fipa\_envelope\_Compose\_\_from(), MC\_AclSend(), and mtp\_http\_CreateMessage().

#### 8.13.1.3 dynstring\_t\* dynstring\_New (void)

Definition at line 8 of file dynstring.c.

References COMPOSE\_BLOCKSIZE, dynstring\_s::len, dynstring\_s::message, and dynstring\_s::size.

Referenced by fipa\_acl\_Compose(), fipa\_envelope\_Compose\_\_from(), and mtp\_http\_CreateMessage().

## 8.14 /home/dko/projects/mobilec/trunk/src/fipa\_acl.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <strings.h>
#include "include/fipa_acl.h"
#include "include/mc_error.h"
#include "include/macros.h"
```

### Defines

- #define [FREEMEM](#)(x) if (x != NULL) free(x)

### Functions

- [fipa\\_acl\\_message\\_t \\* fipa\\_acl\\_message\\_New](#) (void)
- [int fipa\\_acl\\_message\\_Destroy](#) ([fipa\\_acl\\_message\\_t](#) \*message)
- [fipa\\_acl\\_message\\_t \\* fipa\\_acl\\_message\\_Copy](#) ([fipa\\_acl\\_message\\_t](#) \*src)
- [fipa\\_message\\_string\\_t \\* fipa\\_message\\_string\\_New](#) (void)
- [int fipa\\_message\\_string\\_Destroy](#) ([fipa\\_message\\_string\\_t](#) \*message)
- [fipa\\_message\\_string\\_t \\* fipa\\_message\\_string\\_Copy](#) ([fipa\\_message\\_string\\_t](#) \*src)
- [fipa\\_url\\_sequence\\_t \\* fipa\\_url\\_sequence\\_New](#) (void)
- [int fipa\\_url\\_sequence\\_Destroy](#) ([fipa\\_url\\_sequence\\_t](#) \*sequence)
- [fipa\\_url\\_sequence\\_t \\* fipa\\_url\\_sequence\\_Copy](#) ([fipa\\_url\\_sequence\\_t](#) \*src)
- [fipa\\_agent\\_identifier\\_set\\_t \\* fipa\\_agent\\_identifier\\_set\\_New](#) (void)
- [int fipa\\_agent\\_identifier\\_set\\_Destroy](#) ([fipa\\_agent\\_identifier\\_set\\_t](#) \*idset)
- [fipa\\_agent\\_identifier\\_set\\_t \\* fipa\\_agent\\_identifier\\_set\\_Copy](#) ([fipa\\_agent\\_identifier\\_set\\_t](#) \*src)
- [fipa\\_agent\\_identifier\\_t \\* fipa\\_agent\\_identifier\\_New](#) (void)
- [int fipa\\_agent\\_identifier\\_Destroy](#) ([fipa\\_agent\\_identifier\\_t](#) \*id)
- [fipa\\_agent\\_identifier\\_t \\* fipa\\_agent\\_identifier\\_Copy](#) ([fipa\\_agent\\_identifier\\_t](#) \*src)
- [fipa\\_expression\\_t \\* fipa\\_expression\\_New](#) (void)
- [int fipa\\_expression\\_Destroy](#) ([fipa\\_expression\\_t](#) \*expr)
- [fipa\\_expression\\_t \\* fipa\\_expression\\_Copy](#) ([fipa\\_expression\\_t](#) \*src)
- [fipa\\_word\\_t \\* fipa\\_word\\_New](#) (void)
- [int fipa\\_word\\_Destroy](#) ([fipa\\_word\\_t](#) \*word)
- [fipa\\_word\\_t \\* fipa\\_word\\_Copy](#) ([fipa\\_word\\_t](#) \*src)
- [fipa\\_string\\_t \\* fipa\\_string\\_New](#) (void)
- [int fipa\\_string\\_Destroy](#) ([fipa\\_string\\_t](#) \*str)
- [fipa\\_string\\_t \\* fipa\\_string\\_Copy](#) ([fipa\\_string\\_t](#) \*src)
- [fipa\\_DateTime\\_t \\* fipa\\_DateTime\\_New](#) (void)
- [int fipa\\_DateTime\\_Destroy](#) ([fipa\\_DateTime\\_t](#) \*dt)
- [fipa\\_DateTime\\_t \\* fipa\\_DateTime\\_Copy](#) ([fipa\\_DateTime\\_t](#) \*src)
- [fipa\\_url\\_t \\* fipa\\_url\\_New](#) (void)
- [int fipa\\_url\\_Destroy](#) ([fipa\\_url\\_t](#) \*url)
- [fipa\\_url\\_t \\* fipa\\_url\\_Copy](#) ([fipa\\_url\\_t](#) \*src)
- [fipa\\_number\\_t \\* fipa\\_number\\_New](#) (void)

- `int fipa_number_Destroy (fipa_number_t *number)`
- `fipa_number_t * fipa_number_Copy (fipa_number_t *src)`
- `int fipa_acl_Parse (fipa_acl_message_p acl, fipa_message_string_p message)`
- `int fipa_message_parameter_Parse (fipa_acl_message_p acl, fipa_message_string_p message)`
- `int fipa_message_type_Parse (enum fipa_performative_e *performative, fipa_message_string_p message)`
- `int fipa_GetAtom (fipa_message_string_p message, char expected_atom)`
- `int fipa_word_Parse (fipa_word_t **word, fipa_message_string_p message)`
- `int fipa_CheckNextToken (const fipa_message_string_p message, const char *token)`
- `int fipa_expression_Parse (fipa_expression_t **expression, fipa_message_string_p message)`
- `int fipa_GetNextWord (char **word, const fipa_message_string_p message)`
- `int fipa_GetWholeToken (char **word, fipa_message_string_p message)`
- `int fipa_datetime_Parse (fipa_DateTime_p *datetime, fipa_message_string_p message)`
- `int fipa_string_Parse (fipa_string_p *fipa_string, fipa_message_string_p message)`
- `int fipa_agent_identifier_Parse (fipa_agent_identifier_p *aid, fipa_message_string_p message)`
- `int fipa_url_sequence_Parse (fipa_url_sequence_p *urls, fipa_message_string_p message)`
- `int fipa_url_Parse (fipa_url_p *url, fipa_message_string_p message)`
- `int fipa_agent_identifier_set_Parse (fipa_agent_identifier_set_p *agent_ids, fipa_message_string_p message)`
- `int fipa_acl_Compose (dynstring_t **msg, fipa_acl_message_t *acl)`
- `int fipa_performative_Compose (dynstring_t *msg, enum fipa_performative_e performative)`
- `int fipa_url_sequence_Compose (dynstring_t *msg, fipa_url_sequence_t *urls)`
- `int fipa_agent_identifier_set_Compose (dynstring_t *msg, fipa_agent_identifier_set_t *ids)`
- `int fipa_agent_identifier_Compose (dynstring_t *msg, fipa_agent_identifier_t *id)`
- `int fipa_expression_Compose (dynstring_t *msg, fipa_expression_t *expr)`
- `int fipa_word_Compose (dynstring_t *msg, fipa_word_t *word)`
- `int fipa_string_Compose (dynstring_t *msg, fipa_string_t *string)`
- `int fipa_DateTime_Compose (dynstring_t *msg, fipa_DateTime_t *date)`
- `int fipa_url_Compose (dynstring_t *msg, fipa_url_t *url)`
- `int fipa_number_Compose (dynstring_t *msg, fipa_number_t *number)`
- `struct fipa_acl_message_s * fipa_Reply (struct fipa_acl_message_s *acl)`

## 8.14.1 Define Documentation

### 8.14.1.1 #define FREEMEM(x) if (x != NULL) free(x)

Definition at line 48 of file `fipa_acl.c`.

Referenced by `fipa_expression_Destroy()`.

## 8.14.2 Function Documentation

### 8.14.2.1 int fipa\_acl\_Compose (dynstring\_t \*\*msg, fipa\_acl\_message\_t \*acl)

Definition at line 1185 of file `fipa_acl.c`.

References `fipa_acl_message_s::content`, `fipa_acl_message_s::conversation_id`, `dynstring_Append()`, `dynstring_New()`, `fipa_acl_message_s::encoding`, `fipa_agent_identifier_Compose()`, `fipa_agent_identifier_set_Compose()`, `fipa_DateTime_Compose()`, `fipa_expression_Compose()`, `fipa_performative_Compose()`, `fipa_string_Compose()`, `fipa_word_Compose()`, `fipa_acl_message_s::in_reply_to`, `fipa_acl_message_s::language`, `fipa_acl_message_s::ontology`, `fipa_acl_message_s::performative`, `fipa_acl_message_s::protocol`, `fipa_acl_message_s::receiver`, `fipa_acl_message_s::reply_by`, `fipa_acl_message_s::reply_to`, `fipa_acl_message_s::reply_with`, and `fipa_acl_message_s::sender`.

Referenced by MC\_AclSend().

#### 8.14.2.2 `fipa_acl_message_t* fipa_acl_message_Copy (fipa_acl_message_t * src)`

Definition at line 81 of file fipa\_acl.c.

References `fipa_acl_message_s::content`, `fipa_acl_message_s::conversation_id`, `fipa_acl_message_s::encoding`, `fipa_acl_message_New()`, `fipa_agent_identifier_Copy()`, `fipa_agent_identifier_set_Copy()`, `fipa_DateTime_Copy()`, `fipa_expression_Copy()`, `fipa_string_Copy()`, `fipa_word_Copy()`, `fipa_acl_message_s::in_reply_to`, `fipa_acl_message_s::language`, `fipa_acl_message_s::ontology`, `fipa_acl_message_s::performative`, `fipa_acl_message_s::protocol`, `fipa_acl_message_s::receiver`, `fipa_acl_message_s::reply_by`, `fipa_acl_message_s::reply_to`, `fipa_acl_message_s::reply_with`, and `fipa_acl_message_s::sender`.

#### 8.14.2.3 `int fipa_acl_message_Destroy (fipa_acl_message_t * message)`

Definition at line 60 of file fipa\_acl.c.

References `fipa_acl_message_s::content`, `fipa_acl_message_s::conversation_id`, `fipa_acl_message_s::encoding`, `fipa_agent_identifier_Destroy()`, `fipa_agent_identifier_set_Destroy()`, `fipa_DateTime_Destroy()`, `fipa_expression_Destroy()`, `fipa_string_Destroy()`, `fipa_word_Destroy()`, `fipa_acl_message_s::in_reply_to`, `fipa_acl_message_s::language`, `fipa_acl_message_s::ontology`, `fipa_acl_message_s::protocol`, `fipa_acl_message_s::receiver`, `fipa_acl_message_s::reply_by`, `fipa_acl_message_s::reply_to`, `fipa_acl_message_s::reply_with`, and `fipa_acl_message_s::sender`.

Referenced by `acc_Thread()`, and `MC_AclDestroy()`.

#### 8.14.2.4 `fipa_acl_message_t* fipa_acl_message_New (void)`

Definition at line 52 of file fipa\_acl.c.

Referenced by `acc_Thread()`, `fipa_acl_message_Copy()`, `fipa_Reply()`, and `MC_AclNew()`.

#### 8.14.2.5 `int fipa_acl_Parse (fipa_acl_message_p acl, fipa_message_string_p message)`

Definition at line 458 of file fipa\_acl.c.

References `fipa_GetAtom()`, `fipa_message_parameter_Parse()`, `fipa_message_type_Parse()`, `MC_ERR_PARSE`, and `fipa_acl_message_s::performative`.

Referenced by `acc_Thread()`.

#### 8.14.2.6 `int fipa_agent_identifier_Compose (dynstring_t * msg, fipa_agent_identifier_t * id)`

Definition at line 1355 of file fipa\_acl.c.

References `fipa_agent_identifier_s::addresses`, `dynstring_Append()`, `fipa_agent_identifier_set_Compose()`, `fipa_url_sequence_Compose()`, `fipa_agent_identifier_s::name`, `fipa_agent_identifier_set_s::num`, `fipa_url_sequence_s::num`, and `fipa_agent_identifier_s::resolvers`.

Referenced by `fipa_acl_Compose()`, and `fipa_agent_identifier_set_Compose()`.



**8.14.2.7 fipa\_agent\_identifier\_t\* fipa\_agent\_identifier\_Copy (fipa\_agent\_identifier\_t \* src)**

Definition at line 229 of file fipa\_acl.c.

References fipa\_agent\_identifier\_s::addresses, fipa\_agent\_identifier\_New(), fipa\_agent\_identifier\_set\_Copy(), fipa\_url\_sequence\_Copy(), fipa\_agent\_identifier\_s::name, and fipa\_agent\_identifier\_s::resolvers.

Referenced by fipa\_acl\_message\_Copy(), fipa\_acl\_Param\_Copy(), fipa\_agent\_identifier\_set\_Copy(), and fipa\_Reply().

**8.14.2.8 int fipa\_agent\_identifier\_Destroy (fipa\_agent\_identifier\_t \* id)**

Definition at line 217 of file fipa\_acl.c.

References fipa\_agent\_identifier\_s::addresses, fipa\_agent\_identifier\_set\_Destroy(), fipa\_url\_sequence\_Destroy(), fipa\_agent\_identifier\_s::name, and fipa\_agent\_identifier\_s::resolvers.

Referenced by fipa\_acl\_message\_Destroy(), fipa\_acl\_Param\_Destroy(), fipa\_agent\_identifier\_set\_Destroy(), and MC\_AclSetSender().

**8.14.2.9 fipa\_agent\_identifier\_t\* fipa\_agent\_identifier\_New (void)**

Definition at line 209 of file fipa\_acl.c.

Referenced by fipa\_agent\_identifier\_Copy(), fipa\_envelope\_ParseAgentIdentifier(), MC\_AclAddReceiver(), MC\_AclAddReplyTo(), and MC\_AclSetSender().

**8.14.2.10 int fipa\_agent\_identifier\_Parse (fipa\_agent\_identifier\_p \* aid, fipa\_message\_string\_p message)**

Definition at line 985 of file fipa\_acl.c.

References CHECK\_NULL, fipa\_word\_s::content, fipa\_agent\_identifier\_set\_Parse(), fipa\_GetAtom(), fipa\_url\_sequence\_Parse(), fipa\_word\_Destroy(), fipa\_word\_Parse(), MC\_ERR\_PARSE, MC\_SUCCESS, and fipa\_message\_string\_s::parse.

Referenced by fipa\_agent\_identifier\_set\_Parse(), and fipa\_message\_parameter\_Parse().

**8.14.2.11 int fipa\_agent\_identifier\_set\_Compose (dynstring\_t \* msg, fipa\_agent\_identifier\_set\_t \* ids)**

Definition at line 1342 of file fipa\_acl.c.

References dynstring\_Append(), fipa\_agent\_identifier\_Compose(), fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, and fipa\_agent\_identifier\_set\_s::num.

Referenced by fipa\_acl\_Compose(), and fipa\_agent\_identifier\_Compose().

**8.14.2.12 fipa\_agent\_identifier\_set\_t\* fipa\_agent\_identifier\_set\_Copy (fipa\_agent\_identifier\_set\_t \* src)**

Definition at line 188 of file fipa\_acl.c.

References fipa\_agent\_identifier\_Copy(), fipa\_agent\_identifier\_set\_New(), fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_agent\_identifier\_set\_s::num, and fipa\_agent\_identifier\_set\_s::retain\_order.

Referenced by `fipa_acl_message_Copy()`, `fipa_acl_Param_Copy()`, `fipa_agent_identifier_Copy()`, and `fipa_Reply()`.

#### 8.14.2.13 `int fipa_agent_identifier_set_Destroy (fipa_agent_identifier_set_t * idset)`

Definition at line 176 of file `fipa_acl.c`.

References `fipa_agent_identifier_Destroy()`, `fipa_agent_identifier_set_s::fipa_agent_identifiers`, and `fipa_agent_identifier_set_s::num`.

Referenced by `fipa_acl_message_Destroy()`, `fipa_acl_Param_Destroy()`, and `fipa_agent_identifier_Destroy()`.

#### 8.14.2.14 `fipa_agent_identifier_set_t* fipa_agent_identifier_set_New (void)`

Definition at line 168 of file `fipa_acl.c`.

Referenced by `fipa_agent_identifier_set_Copy()`, `fipa_envelope_HandleIntendedReceiver()`, `fipa_envelope_HandleTo()`, `fipa_Reply()`, `MC_AclAddReceiver()`, and `MC_AclAddReplyTo()`.

#### 8.14.2.15 `int fipa_agent_identifier_set_Parse (fipa_agent_identifier_set_p * agent_ids, fipa_message_string_p message)`

Definition at line 1144 of file `fipa_acl.c`.

References `fipa_word_s::content`, `fipa_agent_identifier_Parse()`, `fipa_GetAtom()`, `fipa_word_Parse()`, `MC_ERR_PARSE`, `MC_SUCCESS`, and `fipa_agent_identifier_set_s::retain_order`.

Referenced by `fipa_agent_identifier_Parse()`, and `fipa_message_parameter_Parse()`.

#### 8.14.2.16 `int fipa_CheckNextToken (const fipa_message_string_p message, const char * token)`

Definition at line 647 of file `fipa_acl.c`.

References `fipa_message_string_s::parse`.

Referenced by `fipa_expression_Parse()`.

#### 8.14.2.17 `int fipa_DateTime_Compose (dynstring_t * msg, fipa_DateTime_t * date)`

Definition at line 1428 of file `fipa_acl.c`.

References `fipa_DateTime_s::day`, `dynstring_Append()`, `fipa_DateTime_s::hour`, `fipa_DateTime_s::millisecond`, `fipa_DateTime_s::minute`, `fipa_DateTime_s::month`, `fipa_DateTime_s::second`, `fipa_DateTime_s::sign`, and `fipa_DateTime_s::year`.

Referenced by `fipa_acl_Compose()`, and `fipa_expression_Compose()`.

#### 8.14.2.18 `fipa_DateTime_t* fipa_DateTime_Copy (fipa_DateTime_t * src)`

Definition at line 392 of file `fipa_acl.c`.

References `fipa_DateTime_New()`.

Referenced by `fipa_acl_envelope_Received_Copy()`, `fipa_acl_message_Copy()`, `fipa_acl_Param_Copy()`, and `fipa_expression_Copy()`.

#### 8.14.2.19 `int fipa_DateTime_Destroy (fipa_DateTime_t * dt)`

Definition at line 385 of file `fipa_acl.c`.

Referenced by `fipa_acl_envelope_Received_Destroy()`, `fipa_acl_message_Destroy()`, `fipa_acl_Param_Destroy()`, and `fipa_expression_Destroy()`.

#### 8.14.2.20 `fipa_DateTime_t* fipa_DateTime_New (void)`

Definition at line 377 of file `fipa_acl.c`.

Referenced by `fipa_DateTime_Copy()`.

#### 8.14.2.21 `int fipa_datetime_Parse (fipa_DateTime_p * datetime, fipa_message_string_p message)`

Definition at line 802 of file `fipa_acl.c`.

References `fipa_GetWholeToken()`, `MC_ERR_PARSE`, `MC_SUCCESS`, `fipa_message_string_s::parse`, and `fipa_DateTime_s::sign`.

Referenced by `fipa_envelope_HandleDate()`, `fipa_envelope_HandleReceived()`, `fipa_expression_Parse()`, and `fipa_message_parameter_Parse()`.

#### 8.14.2.22 `int fipa_expression_Compose (dynstring_t * msg, fipa_expression_t * expr)`

Definition at line 1381 of file `fipa_acl.c`.

References `fipa_expression_s::content`, `fipa_expression_s::content_u::datetime`, `fipa_expression_s::content_u::expression`, `fipa_DateTime_Compose()`, `FIPA_EXPR_DATETIME`, `FIPA_EXPR_EXPRESSION`, `FIPA_EXPR_NUMBER`, `FIPA_EXPR_STRING`, `FIPA_EXPR_WORD`, `fipa_expression_Compose()`, `fipa_number_Compose()`, `fipa_string_Compose()`, `fipa_word_Compose()`, `MC_ERR_PARSE`, `fipa_expression_s::content_u::number`, `fipa_expression_s::content_u::string`, `fipa_expression_s::type`, and `fipa_expression_s::content_u::word`.

Referenced by `fipa_acl_Compose()`, and `fipa_expression_Compose()`.

#### 8.14.2.23 `fipa_expression_t* fipa_expression_Copy (fipa_expression_t * src)`

Definition at line 280 of file `fipa_acl.c`.

References `fipa_expression_s::content`, `fipa_expression_s::content_u::datetime`, `fipa_expression_s::content_u::expression`, `fipa_DateTime_Copy()`, `FIPA_EXPR_DATETIME`, `FIPA_EXPR_EXPRESSION`, `FIPA_EXPR_NUMBER`, `FIPA_EXPR_STRING`, `FIPA_EXPR_WORD`, `fipa_expression_Copy()`, `fipa_expression_Destroy()`, `fipa_expression_New()`, `fipa_number_Copy()`, `fipa_string_Copy()`, `fipa_word_Copy()`, `fipa_expression_s::content_u::number`, `fipa_expression_s::content_u::string`, `fipa_expression_s::type`, and `fipa_expression_s::content_u::word`.

Referenced by `fipa_acl_message_Copy()`, and `fipa_expression_Copy()`.

**8.14.2.24 int fipa\_expression\_Destroy (fipa\_expression\_t \* *expr*)**

Definition at line 249 of file fipa\_acl.c.

References fipa\_expression\_s::content, fipa\_expression\_s::content\_u::datetime, fipa\_expression\_s::content\_u::expression, fipa\_DateTime\_Destroy(), FIPA\_EXPR\_DATETIME, FIPA\_EXPR\_EXPRESSION, FIPA\_EXPR\_NUMBER, FIPA\_EXPR\_STRING, FIPA\_EXPR\_WORD, fipa\_expression\_Destroy(), fipa\_number\_Destroy(), fipa\_string\_Destroy(), fipa\_word\_Destroy(), FREEMEM, fipa\_expression\_s::content\_u::number, fipa\_expression\_s::content\_u::string, fipa\_expression\_s::type, and fipa\_expression\_s::content\_u::word.

Referenced by fipa\_acl\_message\_Destroy(), fipa\_expression\_Copy(), and fipa\_expression\_Destroy().

**8.14.2.25 fipa\_expression\_t\* fipa\_expression\_New (void)**

Definition at line 241 of file fipa\_acl.c.

Referenced by fipa\_expression\_Copy().

**8.14.2.26 int fipa\_expression\_Parse (fipa\_expression\_t \*\* *expression*, fipa\_message\_string\_p *message*)**

Definition at line 666 of file fipa\_acl.c.

References fipa\_CheckNextToken(), fipa\_datetime\_Parse(), FIPA\_EXPR\_DATETIME, FIPA\_EXPR\_EXPRESSION, FIPA\_EXPR\_STRING, FIPA\_EXPR\_WORD, fipa\_expression\_Parse(), fipa\_GetAtom(), fipa\_string\_Parse(), fipa\_word\_Parse(), MC\_ERR\_PARSE, MC\_SUCCESS, and fipa\_expression\_s::type.

Referenced by fipa\_expression\_Parse(), and fipa\_message\_parameter\_Parse().

**8.14.2.27 int fipa\_GetAtom (fipa\_message\_string\_p *message*, char *expected\_atom*)**

Definition at line 582 of file fipa\_acl.c.

References MC\_ERR\_PARSE, MC\_SUCCESS, and fipa\_message\_string\_s::parse.

Referenced by fipa\_acl\_Parse(), fipa\_agent\_identifier\_Parse(), fipa\_agent\_identifier\_set\_Parse(), fipa\_expression\_Parse(), fipa\_message\_parameter\_Parse(), fipa\_string\_Parse(), and fipa\_url\_sequence\_Parse().

**8.14.2.28 int fipa\_GetNextWord (char \*\* *word*, const fipa\_message\_string\_p *message*)**

Definition at line 717 of file fipa\_acl.c.

References ERR, MC\_SUCCESS, and fipa\_message\_string\_s::parse.

**8.14.2.29 int fipa\_GetWholeToken (char \*\* *word*, fipa\_message\_string\_p *message*)**

Definition at line 765 of file fipa\_acl.c.

References MC\_SUCCESS, and fipa\_message\_string\_s::parse.

Referenced by fipa\_datetime\_Parse().

**8.14.2.30 int fipa\_message\_parameter\_Parse (fipa\_acl\_message\_p *acl*, fipa\_message\_string\_p *message*)**

Definition at line 478 of file fipa\_acl.c.

References fipa\_acl\_message\_s::content, fipa\_word\_s::content, fipa\_acl\_message\_s::conversation\_id, fipa\_acl\_message\_s::encoding, fipa\_agent\_identifier\_Parse(), fipa\_agent\_identifier\_set\_Parse(), fipa\_datetime\_Parse(), fipa\_expression\_Parse(), fipa\_GetAtom(), fipa\_string\_Parse(), fipa\_word\_Destroy(), fipa\_word\_Parse(), fipa\_acl\_message\_s::in\_reply\_to, fipa\_acl\_message\_s::language, MC\_ERR\_PARSE, fipa\_acl\_message\_s::ontology, fipa\_acl\_message\_s::protocol, fipa\_acl\_message\_s::receiver, fipa\_acl\_message\_s::reply\_by, fipa\_acl\_message\_s::reply\_to, fipa\_acl\_message\_s::reply\_with, and fipa\_acl\_message\_s::sender.

Referenced by fipa\_acl\_Parse().

**8.14.2.31 fipa\_message\_string\_t\* fipa\_message\_string\_Copy (fipa\_message\_string\_t \* *src*)**

Definition at line 122 of file fipa\_acl.c.

References fipa\_message\_string\_s::message, and fipa\_message\_string\_s::parse.

**8.14.2.32 int fipa\_message\_string\_Destroy (fipa\_message\_string\_t \* *message*)**

Definition at line 112 of file fipa\_acl.c.

References fipa\_message\_string\_s::message.

Referenced by acc\_Thread().

**8.14.2.33 fipa\_message\_string\_t\* fipa\_message\_string\_New (void)**

Definition at line 104 of file fipa\_acl.c.

Referenced by acc\_Thread().

**8.14.2.34 int fipa\_message\_type\_Parse (enum fipa\_performative\_e \* *performative*, fipa\_message\_string\_p *message*)**

Definition at line 520 of file fipa\_acl.c.

References fipa\_word\_s::content, FIPA\_ACCEPT\_PROPOSAL, FIPA\_AGREE, FIPA\_CALL\_FOR\_PROPOSAL, FIPA\_CANCEL, FIPA\_CONFIRM, FIPA\_DISCONFIRM, FIPA\_FAILURE, FIPA\_INFORM, FIPA\_INFORM\_IF, FIPA\_INFORM\_REF, FIPA\_NOT\_UNDERSTOOD, FIPA\_PROPOGATE, FIPA\_PROPOSE, FIPA\_PROXY, FIPA\_QUERY\_IF, FIPA\_QUERY\_REF, FIPA\_REFUSE, FIPA\_REJECT\_PROPOSAL, FIPA\_REQUEST, FIPA\_REQUEST\_WHEN, FIPA\_REQUEST\_WHENEVER, FIPA\_SUBSCRIBE, fipa\_word\_Destroy(), fipa\_word\_Parse(), and MC\_ERR\_PARSE.

Referenced by fipa\_acl\_Parse().

**8.14.2.35 int fipa\_number\_Compose (dynstring\_t \* *msg*, fipa\_number\_t \* *number*)**

Definition at line 1456 of file fipa\_acl.c.

References dynstring\_Append(), and fipa\_number\_s::str.

Referenced by `fipa_expression_Compose()`.

#### **8.14.2.36** `fipa_number_t* fipa_number_Copy (fipa_number_t * src)`

Definition at line 448 of file `fipa_acl.c`.

References `fipa_number_New()`, and `fipa_number_s::str`.

Referenced by `fipa_expression_Copy()`.

#### **8.14.2.37** `int fipa_number_Destroy (fipa_number_t * number)`

Definition at line 438 of file `fipa_acl.c`.

References `fipa_number_s::str`.

Referenced by `fipa_expression_Destroy()`.

#### **8.14.2.38** `fipa_number_t* fipa_number_New (void)`

Definition at line 430 of file `fipa_acl.c`.

Referenced by `fipa_number_Copy()`.

#### **8.14.2.39** `int fipa_performative_Compose (dynstring_t * msg, enum fipa_performative_e performative)`

Definition at line 1254 of file `fipa_acl.c`.

References `dynstring_Append()`, `FIPA_ACCEPT_PROPOSAL`, `FIPA_AGREE`, `FIPA_CALL_FOR_PROPOSAL`, `FIPA_CANCEL`, `FIPA_CONFIRM`, `FIPA_DISCONFIRM`, `FIPA_FAILURE`, `FIPA_INFORM`, `FIPA_INFORM_IF`, `FIPA_INFORM_REF`, `FIPA_NOT_UNDERSTOOD`, `FIPA_PROPOGATE`, `FIPA_PROPOSE`, `FIPA_PROXY`, `FIPA_QUERY_IF`, `FIPA_QUERY_REF`, `FIPA_REFUSE`, `FIPA_REJECT_PROPOSAL`, `FIPA_REQUEST`, `FIPA_REQUEST_WHEN`, `FIPA_REQUEST_WHENEVER`, `FIPA_SUBSCRIBE`, and `MC_ERR_PARSE`.

Referenced by `fipa_acl_Compose()`.

#### **8.14.2.40** `struct fipa_acl_message_s* fipa_Reply (struct fipa_acl_message_s * acl)` [read]

Definition at line 1464 of file `fipa_acl.c`.

References `fipa_acl_message_New()`, `fipa_agent_identifier_Copy()`, `fipa_agent_identifier_set_Copy()`, `fipa_agent_identifier_set_New()`, `fipa_agent_identifier_set_s::fipa_agent_identifiers`, `fipa_agent_identifier_set_s::num`, `fipa_acl_message_s::receiver`, `fipa_acl_message_s::reply_to`, `fipa_agent_identifier_set_s::retain_order`, and `fipa_acl_message_s::sender`.

Referenced by `MC_AclReply()`.

#### **8.14.2.41** `int fipa_string_Compose (dynstring_t * msg, fipa_string_t * string)`

Definition at line 1419 of file `fipa_acl.c`.

References `fipa_string_s::content`, and `dynstring_Append()`.

Referenced by fipa\_acl\_Compose(), and fipa\_expression\_Compose().

#### 8.14.2.42 fipa\_string\_t\* fipa\_string\_Copy (fipa\_string\_t \* src)

Definition at line 367 of file fipa\_acl.c.

References fipa\_string\_s::content, and fipa\_string\_New().

Referenced by fipa\_acl\_message\_Copy(), and fipa\_expression\_Copy().

#### 8.14.2.43 int fipa\_string\_Destroy (fipa\_string\_t \* str)

Definition at line 357 of file fipa\_acl.c.

References fipa\_string\_s::content.

Referenced by fipa\_acl\_message\_Destroy(), fipa\_expression\_Destroy(), and MC\_AclSetContent().

#### 8.14.2.44 fipa\_string\_t\* fipa\_string\_New (void)

Definition at line 349 of file fipa\_acl.c.

Referenced by fipa\_string\_Copy(), and MC\_AclSetContent().

#### 8.14.2.45 int fipa\_string\_Parse (fipa\_string\_p \* fipa\_string, fipa\_message\_string\_p message)

Definition at line 940 of file fipa\_acl.c.

References fipa\_string\_s::content, fipa\_GetAtom(), MC\_ERR\_PARSE, MC\_SUCCESS, and fipa\_message\_string\_s::parse.

Referenced by fipa\_expression\_Parse(), and fipa\_message\_parameter\_Parse().

#### 8.14.2.46 int fipa\_url\_Compose (dynstring\_t \* msg, fipa\_url\_t \* url)

Definition at line 1448 of file fipa\_acl.c.

References dynstring\_Append(), and fipa\_url\_s::str.

Referenced by fipa\_url\_sequence\_Compose().

#### 8.14.2.47 fipa\_url\_t\* fipa\_url\_Copy (fipa\_url\_t \* src)

Definition at line 420 of file fipa\_acl.c.

References fipa\_url\_New(), and fipa\_url\_s::str.

Referenced by fipa\_acl\_envelope\_Received\_Copy(), and fipa\_url\_sequence\_Copy().

#### 8.14.2.48 int fipa\_url\_Destroy (fipa\_url\_t \* url)

Definition at line 410 of file fipa\_acl.c.

References fipa\_url\_s::str.

Referenced by fipa\_acl\_envelope\_Received\_Destroy(), and fipa\_url\_sequence\_Destroy().

**8.14.2.49 fipa\_url\_t\* fipa\_url\_New (void)**

Definition at line 402 of file fipa\_acl.c.

Referenced by fipa\_envelope\_HandleReceived(), fipa\_envelope\_ParseAddresses(), fipa\_url\_Copy(), MC\_AclAddReceiver(), MC\_AclAddReplyTo(), and MC\_AclSetSender().

**8.14.2.50 int fipa\_url\_Parse (fipa\_url\_p \* url, fipa\_message\_string\_p message)**

Definition at line 1124 of file fipa\_acl.c.

References fipa\_word\_s::content, fipa\_word\_Destroy(), and fipa\_word\_Parse().

Referenced by fipa\_url\_sequence\_Parse().

**8.14.2.51 int fipa\_url\_sequence\_Compose (dynstring\_t \* msg, fipa\_url\_sequence\_t \* urls)**

Definition at line 1329 of file fipa\_acl.c.

References dynstring\_Append(), fipa\_url\_Compose(), fipa\_url\_sequence\_s::num, and fipa\_url\_sequence\_s::urls.

Referenced by fipa\_agent\_identifier\_Compose().

**8.14.2.52 fipa\_url\_sequence\_t\* fipa\_url\_sequence\_Copy (fipa\_url\_sequence\_t \* src)**

Definition at line 152 of file fipa\_acl.c.

References fipa\_url\_Copy(), fipa\_url\_sequence\_New(), fipa\_url\_sequence\_s::num, and fipa\_url\_sequence\_s::urls.

Referenced by fipa\_agent\_identifier\_Copy().

**8.14.2.53 int fipa\_url\_sequence\_Destroy (fipa\_url\_sequence\_t \* sequence)**

Definition at line 140 of file fipa\_acl.c.

References fipa\_url\_Destroy(), fipa\_url\_sequence\_s::num, and fipa\_url\_sequence\_s::urls.

Referenced by fipa\_agent\_identifier\_Destroy().

**8.14.2.54 fipa\_url\_sequence\_t\* fipa\_url\_sequence\_New (void)**

Definition at line 132 of file fipa\_acl.c.

Referenced by fipa\_url\_sequence\_Copy(), fipa\_url\_sequence\_Parse(), MC\_AclAddReceiver(), MC\_AclAddReplyTo(), and MC\_AclSetSender().

**8.14.2.55 int fipa\_url\_sequence\_Parse (fipa\_url\_sequence\_p \* urls, fipa\_message\_string\_p message)**

Definition at line 1092 of file fipa\_acl.c.

References fipa\_word\_s::content, fipa\_GetAtom(), fipa\_url\_Parse(), fipa\_url\_sequence\_New(), fipa\_word\_Destroy(), fipa\_word\_Parse(), and MC\_ERR\_PARSE.



Referenced by `fipa_agent_identifier_Parse()`.

#### 8.14.2.56 `int fipa_word_Compose (dynstring_t * msg, fipa_word_t * word)`

Definition at line 1411 of file `fipa_acl.c`.

References `fipa_word_s::content`, and `dynstring_Append()`.

Referenced by `fipa_acl_Compose()`, and `fipa_expression_Compose()`.

#### 8.14.2.57 `fipa_word_t* fipa_word_Copy (fipa_word_t * src)`

Definition at line 339 of file `fipa_acl.c`.

References `fipa_word_s::content`, and `fipa_word_New()`.

Referenced by `fipa_acl_message_Copy()`, and `fipa_expression_Copy()`.

#### 8.14.2.58 `int fipa_word_Destroy (fipa_word_t * word)`

Definition at line 329 of file `fipa_acl.c`.

References `fipa_word_s::content`.

Referenced by `fipa_acl_message_Destroy()`, `fipa_agent_identifier_Parse()`, `fipa_expression_Destroy()`, `fipa_message_parameter_Parse()`, `fipa_message_type_Parse()`, `fipa_url_Parse()`, and `fipa_url_sequence_Parse()`.

#### 8.14.2.59 `fipa_word_t* fipa_word_New (void)`

Definition at line 321 of file `fipa_acl.c`.

Referenced by `fipa_word_Copy()`.

#### 8.14.2.60 `int fipa_word_Parse (fipa_word_t ** word, fipa_message_string_p message)`

Definition at line 605 of file `fipa_acl.c`.

References `CHECK_NULL`, `MC_ERR_PARSE`, `MC_SUCCESS`, and `fipa_message_string_s::parse`.

Referenced by `fipa_agent_identifier_Parse()`, `fipa_agent_identifier_set_Parse()`, `fipa_expression_Parse()`, `fipa_message_parameter_Parse()`, `fipa_message_type_Parse()`, `fipa_url_Parse()`, and `fipa_url_sequence_Parse()`.

## 8.15 /home/dko/projects/mobilec/trunk/src/fipa\_envelope.c File Reference

```
#include <stdio.h>
#include <mxml.h>
#include <time.h>
#include "include/fipa_acl_envelope.h"
#include "include/mc_error.h"
#include "include/mc_platform.h"
```

### Functions

- [fipa\\_acl\\_envelope\\_Received\\_t \\* fipa\\_acl\\_envelope\\_Received\\_New](#) (void)
- [int fipa\\_acl\\_envelope\\_Received\\_Destroy](#) (fipa\_acl\_envelope\_Received\_t \*received)
- [fipa\\_acl\\_envelope\\_Received\\_t \\* fipa\\_acl\\_envelope\\_Received\\_Copy](#) (fipa\_acl\_envelope\_Received\_t \*received)
- [fipa\\_acl\\_Param\\_t \\* fipa\\_acl\\_Param\\_New](#) (void)
- [int fipa\\_acl\\_Param\\_Destroy](#) (fipa\_acl\_Param\_t \*param)
- [fipa\\_acl\\_Param\\_t \\* fipa\\_acl\\_Param\\_Copy](#) (fipa\_acl\_Param\_t \*param)
- [fipa\\_acl\\_envelope\\_t \\* fipa\\_acl\\_envelope\\_New](#) (void)
- [int fipa\\_acl\\_envelope\\_Destroy](#) (fipa\_acl\_envelope\_t \*envelope)
- [fipa\\_acl\\_envelope\\_t \\* fipa\\_acl\\_envelope\\_Copy](#) (fipa\_acl\_envelope\_t \*envelope)
- [int fipa\\_envelope\\_Parse](#) (struct fipa\_acl\_envelope\_s \*envelope, const char \*message)
- [int fipa\\_envelope\\_HandleEnvelope](#) (struct fipa\_acl\_envelope\_s \*envelope, mxml\_node\_t \*node)
- [int fipa\\_envelope\\_HandleParams](#) (struct fipa\_acl\_envelope\_s \*envelope, mxml\_node\_t \*node)
- [int fipa\\_envelope\\_HandleTo](#) (struct fipa\_acl\_envelope\_s \*envelope, mxml\_node\_t \*param\_node, int cur\_param)
- [int fipa\\_envelope\\_HandleFrom](#) (struct fipa\_acl\_envelope\_s \*envelope, mxml\_node\_t \*param\_node, int cur\_param)
- [int fipa\\_envelope\\_HandleComments](#) (struct fipa\_acl\_envelope\_s \*envelope, mxml\_node\_t \*param\_node, int cur\_param)
- [int fipa\\_envelope\\_HandleAclRepresentation](#) (struct fipa\_acl\_envelope\_s \*envelope, mxml\_node\_t \*param\_node, int cur\_param)
- [int fipa\\_envelope\\_HandlePayloadLength](#) (struct fipa\_acl\_envelope\_s \*envelope, mxml\_node\_t \*param\_node, int cur\_param)
- [int fipa\\_envelope\\_HandlePayloadEncoding](#) (struct fipa\_acl\_envelope\_s \*envelope, mxml\_node\_t \*param\_node, int cur\_param)
- [int fipa\\_envelope\\_HandleDate](#) (struct fipa\_acl\_envelope\_s \*envelope, mxml\_node\_t \*param\_node, int cur\_param)
- [int fipa\\_envelope\\_HandleIntendedReceiver](#) (struct fipa\_acl\_envelope\_s \*envelope, mxml\_node\_t \*param\_node, int cur\_param)
- [int fipa\\_envelope\\_HandleReceived](#) (struct fipa\_acl\_envelope\_s \*envelope, mxml\_node\_t \*param\_node, int cur\_param)
- [int fipa\\_envelope\\_ParseAgentIdentifier](#) (struct fipa\_agent\_identifier\_s \*\*aid, mxml\_node\_t \*agent\_identifier\_node)
- [int fipa\\_envelope\\_ParseAddresses](#) (struct fipa\_agent\_identifier\_s \*aid, mxml\_node\_t \*addresses\_node)

- `int fipa_envelope_ParseResolvers` (struct `fipa_agent_identifier_s` \*aid, `mxml_node_t` \*resolvers\_node)
- `char * fipa_envelope_Compose` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__envelope` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__params` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__to` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__from` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__acl_representation` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__payload_encoding` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__date` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__intended_receiver` (`fipa_acl_message_t` \*fipa\_acl)

## 8.15.1 Function Documentation

### 8.15.1.1 `fipa_acl_envelope_t* fipa_acl_envelope_Copy` (`fipa_acl_envelope_t` \* *envelope*)

Definition at line 145 of file `fipa_envelope.c`.

References `fipa_acl_envelope_New()`, `fipa_acl_Param_Copy()`, `fipa_acl_envelope_s::num_params`, and `fipa_acl_envelope_s::params`.

### 8.15.1.2 `int fipa_acl_envelope_Destroy` (`fipa_acl_envelope_t` \* *envelope*)

Definition at line 132 of file `fipa_envelope.c`.

References `fipa_acl_Param_Destroy()`, `fipa_acl_envelope_s::num_params`, and `fipa_acl_envelope_s::params`.

Referenced by `acc_Thread()`.

### 8.15.1.3 `fipa_acl_envelope_t* fipa_acl_envelope_New` (void)

Definition at line 124 of file `fipa_envelope.c`.

Referenced by `acc_Thread()`, and `fipa_acl_envelope_Copy()`.

### 8.15.1.4 `fipa_acl_envelope_Received_t* fipa_acl_envelope_Received_Copy` (`fipa_acl_envelope_Received_t` \* *received*)

Definition at line 64 of file `fipa_envelope.c`.

References `fipa_acl_envelope_Received_New()`, `fipa_DateTime_Copy()`, `fipa_url_Copy()`, `fipa_acl_envelope_Received_s::received_by`, `fipa_acl_envelope_Received_s::received_date`, `fipa_acl_envelope_Received_s::received_from`, `fipa_acl_envelope_Received_s::received_id`, and `fipa_acl_envelope_Received_s::received_via`.

Referenced by `fipa_acl_Param_Copy()`.

### 8.15.1.5 `int fipa_acl_envelope_Received_Destroy` (`fipa_acl_envelope_Received_t` \* *received*)

Definition at line 52 of file `fipa_envelope.c`.

References `fipa_DateTime_Destroy()`, `fipa_url_Destroy()`, `fipa_acl_envelope_Received_s::received_by`, `fipa_acl_envelope_Received_s::received_date`, `fipa_acl_envelope_Received_s::received_from`, `fipa_acl_envelope_Received_s::received_id`, and `fipa_acl_envelope_Received_s::received_via`.

Referenced by `fipa_acl_Param_Destroy()`.

#### 8.15.1.6 `fipa_acl_envelope_Received_t* fipa_acl_envelope_Received_New (void)`

Definition at line 43 of file `fipa_envelope.c`.

Referenced by `fipa_acl_envelope_Received_Copy()`, and `fipa_envelope_HandleReceived()`.

#### 8.15.1.7 `fipa_acl_Param_t* fipa_acl_Param_Copy (fipa_acl_Param_t * param)`

Definition at line 105 of file `fipa_envelope.c`.

References `fipa_acl_Param_s::acl_representation`, `fipa_acl_Param_s::comments`, `fipa_acl_Param_s::date`, `fipa_acl_envelope_Received_Copy()`, `fipa_acl_Param_New()`, `fipa_agent_identifier_Copy()`, `fipa_agent_identifier_set_Copy()`, `fipa_DateTime_Copy()`, `fipa_acl_Param_s::from`, `fipa_acl_Param_s::intended_receiver`, `fipa_acl_Param_s::payload_encoding`, `fipa_acl_Param_s::payload_length`, `fipa_acl_Param_s::received`, and `fipa_acl_Param_s::to`.

Referenced by `fipa_acl_envelope_Copy()`.

#### 8.15.1.8 `int fipa_acl_Param_Destroy (fipa_acl_Param_t * param)`

Definition at line 88 of file `fipa_envelope.c`.

References `fipa_acl_Param_s::acl_representation`, `fipa_acl_Param_s::comments`, `fipa_acl_Param_s::date`, `fipa_acl_envelope_Received_Destroy()`, `fipa_agent_identifier_Destroy()`, `fipa_agent_identifier_set_Destroy()`, `fipa_DateTime_Destroy()`, `fipa_acl_Param_s::from`, `fipa_acl_Param_s::intended_receiver`, `fipa_acl_Param_s::payload_encoding`, `fipa_acl_Param_s::payload_length`, `fipa_acl_Param_s::received`, and `fipa_acl_Param_s::to`.

Referenced by `fipa_acl_envelope_Destroy()`.

#### 8.15.1.9 `fipa_acl_Param_t* fipa_acl_Param_New (void)`

Definition at line 80 of file `fipa_envelope.c`.

Referenced by `fipa_acl_Param_Copy()`, and `fipa_envelope_HandleTo()`.

#### 8.15.1.10 `char* fipa_envelope_Compose (fipa_acl_message_t * fipa_acl)`

Definition at line 864 of file `fipa_envelope.c`.

References `fipa_envelope_Compose__envelope()`, `MXML_ADD_AFTER`, `MXML_ADD_TO_PARENT`, `MXML_NO_CALLBACK`, `mxmAdd()`, `mxmDelete()`, `mxmLoadString()`, `mxmSaveAllocString()`, and `node`.

Referenced by `MC_AclSend()`.

**8.15.1.11 mxmml\_node\_t\* fipa\_envelope\_Compose\_\_acl\_representation (fipa\_acl\_message\_t \* *fipa\_acl*)**

Definition at line 1081 of file fipa\_envelope.c.

References mxmmlNewElement(), mxmmlNewText(), and node.

Referenced by fipa\_envelope\_Compose\_\_params().

**8.15.1.12 mxmml\_node\_t\* fipa\_envelope\_Compose\_\_date (fipa\_acl\_message\_t \* *fipa\_acl*)**

Definition at line 1109 of file fipa\_envelope.c.

References mxmmlNewElement(), mxmmlNewText(), and node.

Referenced by fipa\_envelope\_Compose\_\_params().

**8.15.1.13 mxmml\_node\_t\* fipa\_envelope\_Compose\_\_envelope (fipa\_acl\_message\_t \* *fipa\_acl*)**

Definition at line 889 of file fipa\_envelope.c.

References fipa\_envelope\_Compose\_\_params(), MXML\_ADD\_AFTER, MXML\_ADD\_TO\_PARENT, mxmmlAdd(), mxmmlNewElement(), and node.

Referenced by fipa\_envelope\_Compose().

**8.15.1.14 mxmml\_node\_t\* fipa\_envelope\_Compose\_\_from (fipa\_acl\_message\_t \* *fipa\_acl*)**

Definition at line 1016 of file fipa\_envelope.c.

References fipa\_agent\_identifier\_s::addresses, dynstring\_Append(), dynstring\_Destroy(), dynstring\_New(), g\_mc\_platform, mc\_platform\_s::hostname, dynstring\_s::message, mxmmlNewElement(), mxmmlNewText(), fipa\_agent\_identifier\_s::name, fipa\_url\_sequence\_s::num, mc\_platform\_s::port, and fipa\_acl\_message\_s::sender.

Referenced by fipa\_envelope\_Compose\_\_params().

**8.15.1.15 mxmml\_node\_t\* fipa\_envelope\_Compose\_\_intended\_receiver (fipa\_acl\_message\_t \* *fipa\_acl*)**

Definition at line 1138 of file fipa\_envelope.c.

References fipa\_agent\_identifier\_s::addresses, fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, mxmmlNewElement(), mxmmlNewText(), fipa\_agent\_identifier\_s::name, node, fipa\_url\_sequence\_s::num, fipa\_acl\_message\_s::receiver, fipa\_url\_s::str, and fipa\_url\_sequence\_s::urls.

Referenced by fipa\_envelope\_Compose\_\_params().

**8.15.1.16 mxmml\_node\_t\* fipa\_envelope\_Compose\_\_params (fipa\_acl\_message\_t \* *fipa\_acl*)**

Definition at line 909 of file fipa\_envelope.c.

References fipa\_envelope\_Compose\_\_acl\_representation(), fipa\_envelope\_Compose\_\_date(), fipa\_envelope\_Compose\_\_from(), fipa\_envelope\_Compose\_\_intended\_receiver(), fipa\_envelope\_Compose\_\_payload\_encoding(), fipa\_envelope\_Compose\_\_to(), MXML\_ADD\_AFTER, MXML\_ADD\_TO\_PARENT, mxmmlAdd(), mxmmlElementSetAttr(), mxmmlNewElement(), and node.

Referenced by `fipa_envelope_Compose__envelope()`.

#### 8.15.1.17 `mxml_node_t* fipa_envelope_Compose__payload_encoding (fipa_acl_message_t * fipa_acl)`

Definition at line 1095 of file `fipa_envelope.c`.

References `mxmlNewElement()`, `mxmlNewText()`, and `node`.

Referenced by `fipa_envelope_Compose__params()`.

#### 8.15.1.18 `mxml_node_t* fipa_envelope_Compose__to (fipa_acl_message_t * fipa_acl)`

Definition at line 970 of file `fipa_envelope.c`.

References `fipa_agent_identifier_s::addresses`, `fipa_agent_identifier_set_s::fipa_agent_identifiers`, `mxmlNewElement()`, `mxmlNewText()`, `fipa_agent_identifier_s::name`, `node`, `fipa_url_sequence_s::num`, `fipa_agent_identifier_set_s::num`, `fipa_acl_message_s::receiver`, `fipa_url_s::str`, and `fipa_url_sequence_s::urls`.

Referenced by `fipa_envelope_Compose__params()`.

#### 8.15.1.19 `int fipa_envelope_HandleAclRepresentation (struct fipa_acl_envelope_s * envelope, mxml_node_t * param_node, int cur_param)`

Definition at line 376 of file `fipa_envelope.c`.

References `fipa_acl_Param_s::acl_representation`, `mxml_node_s::child`, `MC_ERR_PARSE`, `MXML_DESCEND_FIRST`, `MXML_TEXT`, `mxmlFindElement()`, `fipa_acl_envelope_s::params`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::type`, and `mxml_node_s::value`.

Referenced by `fipa_envelope_HandleParams()`.

#### 8.15.1.20 `int fipa_envelope_HandleComments (struct fipa_acl_envelope_s * envelope, mxml_node_t * param_node, int cur_param)`

Definition at line 352 of file `fipa_envelope.c`.

References `mxml_node_s::child`, `fipa_acl_Param_s::comments`, `MC_ERR_PARSE`, `MXML_DESCEND_FIRST`, `MXML_TEXT`, `mxmlFindElement()`, `fipa_acl_envelope_s::params`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::type`, and `mxml_node_s::value`.

Referenced by `fipa_envelope_HandleParams()`.

#### 8.15.1.21 `int fipa_envelope_HandleDate (struct fipa_acl_envelope_s * envelope, mxml_node_t * param_node, int cur_param)`

Definition at line 447 of file `fipa_envelope.c`.

References `mxml_node_s::child`, `fipa_acl_Param_s::date`, `fipa_datetime_Parse()`, `MC_ERR_PARSE`, `fipa_message_string_s::message`, `MXML_DESCEND_FIRST`, `MXML_TEXT`, `mxmlFindElement()`, `fipa_acl_envelope_s::params`, `fipa_message_string_s::parse`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::type`, and `mxml_node_s::value`.

Referenced by `fipa_envelope_HandleParams()`.

**8.15.1.22** `int fipa_envelope_HandleEnvelope (struct fipa_acl_envelope_s * envelope, mxml_node_t * node)`

Definition at line 173 of file fipa\_envelope.c.

References `fipa_envelope_HandleParams()`, `MC_ERR_PARSE`, `MXML_DESCEND_FIRST`, and `mxmlFindElement()`.

Referenced by `fipa_envelope_Parse()`.

**8.15.1.23** `int fipa_envelope_HandleFrom (struct fipa_acl_envelope_s * envelope, mxml_node_t * param_node, int cur_param)`

Definition at line 323 of file fipa\_envelope.c.

References `fipa_envelope_ParseAgentIdentifier()`, `fipa_acl_Param_s::from`, `MC_ERR_PARSE`, `MXML_DESCEND_FIRST`, `mxmlFindElement()`, and `fipa_acl_envelope_s::params`.

Referenced by `fipa_envelope_HandleParams()`.

**8.15.1.24** `int fipa_envelope_HandleIntendedReceiver (struct fipa_acl_envelope_s * envelope, mxml_node_t * param_node, int cur_param)`

Definition at line 477 of file fipa\_envelope.c.

References `fipa_agent_identifier_set_New()`, `fipa_agent_identifier_set_s::fipa_agent_identifiers`, `fipa_envelope_ParseAgentIdentifier()`, `fipa_acl_Param_s::intended_receiver`, `MC_ERR_PARSE`, `MXML_DESCEND_FIRST`, `MXML_NO_DESCEND`, `mxmlFindElement()`, `fipa_agent_identifier_set_s::num`, `fipa_acl_envelope_s::params`, and `fipa_agent_identifier_set_s::retain_order`.

Referenced by `fipa_envelope_HandleParams()`.

**8.15.1.25** `int fipa_envelope_HandleParams (struct fipa_acl_envelope_s * envelope, mxml_node_t * node)`

Definition at line 193 of file fipa\_envelope.c.

References `fipa_envelope_HandleAclRepresentation()`, `fipa_envelope_HandleComments()`, `fipa_envelope_HandleDate()`, `fipa_envelope_HandleFrom()`, `fipa_envelope_HandleIntendedReceiver()`, `fipa_envelope_HandlePayloadEncoding()`, `fipa_envelope_HandlePayloadLength()`, `fipa_envelope_HandleReceived()`, `fipa_envelope_HandleTo()`, `MC_ERR_PARSE`, `MXML_DESCEND_FIRST`, `mxmlFindElement()`, `fipa_acl_envelope_s::num_params`, and `fipa_acl_envelope_s::params`.

Referenced by `fipa_envelope_HandleEnvelope()`.

**8.15.1.26** `int fipa_envelope_HandlePayloadEncoding (struct fipa_acl_envelope_s * envelope, mxml_node_t * param_node, int cur_param)`

Definition at line 423 of file fipa\_envelope.c.

References `mxml_node_s::child`, `MC_ERR_PARSE`, `MXML_DESCEND_FIRST`, `MXML_TEXT`, `mxmlFindElement()`, `fipa_acl_envelope_s::params`, `fipa_acl_Param_s::payload_encoding`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::type`, and `mxml_node_s::value`.

Referenced by `fipa_envelope_HandleParams()`.

### 8.15.1.27 **int fipa\_envelope\_HandlePayloadLength** (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *param\_node*, int *cur\_param*)

Definition at line 400 of file fipa\_envelope.c.

References mxml\_node\_s::child, MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_TEXT, mxmlFindElement(), fipa\_acl\_envelope\_s::params, fipa\_acl\_Param\_s::payload\_length, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, and mxml\_node\_s::value.

Referenced by fipa\_envelope\_HandleParams().

### 8.15.1.28 **int fipa\_envelope\_HandleReceived** (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *param\_node*, int *cur\_param*)

Definition at line 555 of file fipa\_envelope.c.

References fipa\_acl\_envelope\_Received\_New(), fipa\_datetime\_Parse(), fipa\_url\_New(), MC\_ERR\_PARSE, fipa\_message\_string\_s::message, MXML\_DESCEND\_FIRST, mxmlElementGetAttr(), mxmlFindElement(), node, fipa\_acl\_envelope\_s::params, fipa\_message\_string\_s::parse, fipa\_acl\_Param\_s::received, fipa\_acl\_envelope\_Received\_s::received\_by, fipa\_acl\_envelope\_Received\_s::received\_date, fipa\_acl\_envelope\_Received\_s::received\_from, fipa\_acl\_envelope\_Received\_s::received\_id, fipa\_acl\_envelope\_Received\_s::received\_via, and fipa\_url\_s::str.

Referenced by fipa\_envelope\_HandleParams().

### 8.15.1.29 **int fipa\_envelope\_HandleTo** (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *param\_node*, int *cur\_param*)

Definition at line 245 of file fipa\_envelope.c.

References fipa\_acl\_Param\_New(), fipa\_agent\_identifier\_set\_New(), fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_envelope\_ParseAgentIdentifier(), MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_NO\_DESCEND, mxmlFindElement(), fipa\_agent\_identifier\_set\_s::num, fipa\_acl\_envelope\_s::params, fipa\_agent\_identifier\_set\_s::retain\_order, and fipa\_acl\_Param\_s::to.

Referenced by fipa\_envelope\_HandleParams().

### 8.15.1.30 **int fipa\_envelope\_Parse** (struct fipa\_acl\_envelope\_s \* *envelope*, const char \* *message*)

Definition at line 160 of file fipa\_envelope.c.

References fipa\_envelope\_HandleEnvelope(), MXML\_NO\_CALLBACK, mxmlDelete(), and mxmlLoadString().

Referenced by acc\_Thread().

### 8.15.1.31 **int fipa\_envelope\_ParseAddresses** (struct fipa\_agent\_identifier\_s \* *aid*, mxml\_node\_t \* *addresses\_node*)

Definition at line 732 of file fipa\_envelope.c.

References fipa\_agent\_identifier\_s::addresses, mxml\_node\_s::child, fipa\_url\_New(), MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_NO\_DESCEND, MXML\_TEXT, mxmlFindElement(), fipa\_url\_sequence\_s::num, fipa\_url\_s::str, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, fipa\_url\_sequence\_s::urls, and mxml\_node\_s::value.



Referenced by fipa\_envelope\_ParseAgentIdentifier().

**8.15.1.32 int fipa\_envelope\_ParseAgentIdentifier (struct fipa\_agent\_identifier\_s \*\* *aid*,  
mxml\_node\_t \* *agent\_identifier\_node*)**

Definition at line 682 of file fipa\_envelope.c.

References mxml\_node\_s::child, fipa\_agent\_identifier\_New(), fipa\_envelope\_ParseAddresses(), fipa\_envelope\_ParseResolvers(), MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_TEXT, mxmlFindElement(), mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, and mxml\_node\_s::value.

Referenced by fipa\_envelope\_HandleFrom(), fipa\_envelope\_HandleIntendedReceiver(), fipa\_envelope\_HandleTo(), and fipa\_envelope\_ParseResolvers().

**8.15.1.33 int fipa\_envelope\_ParseResolvers (struct fipa\_agent\_identifier\_s \* *aid*, mxml\_node\_t \*  
*resolvers\_node*)**

Definition at line 799 of file fipa\_envelope.c.

References fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_envelope\_ParseAgentIdentifier(), MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_NO\_DESCEND, mxmlFindElement(), fipa\_agent\_identifier\_set\_s::num, fipa\_agent\_identifier\_s::resolvers, and fipa\_agent\_identifier\_set\_s::retain\_order.

Referenced by fipa\_envelope\_ParseAgentIdentifier().

## 8.16 /home/dko/projects/mobilec/trunk/src/include/acc.h File Reference

```
#include <sys/socket.h>
#include "macros.h"
```

### Functions

- **STRUCT** (acc, struct [mc\\_platform\\_s](#) \*[mc\\_platform](#); [THREAD\\_T](#) [thread](#); [THREAD\\_T](#) [message\\_handler\\_thread](#); [THREAD\\_T](#) [listen\\_thread](#); [int](#) [waiting](#); [MUTEX\\_T](#) \*[waiting\\_lock](#); [COND\\_T](#) \*[waiting\\_cond](#);) **STRUCT**([listen\\_thread\\_arg](#)
- [acc\\_p](#) [acc\\_Initialize](#) (struct [mc\\_platform\\_s](#) \*[mc\\_platform](#))
- [int](#) [acc\\_Destroy](#) ([acc\\_p](#) [acc](#))
- [void](#) [acc\\_Start](#) (struct [mc\\_platform\\_s](#) \*[mc\\_platform](#))
- [void](#) \* [acc\\_MessageHandlerThread](#) ([void](#) \*[arg](#))
- [void](#) \* [acc\\_Thread](#) ([void](#) \*[arg](#))
- [void](#) \* [listen\\_Thread](#) ([void](#) \*[arg](#))

### Variables

- [u\\_long](#) [port](#)
- struct [sockaddr\\_in](#) \* [addr](#)
- unsigned long [int](#) [client\\_fd](#)
- unsigned long [int](#) [server\\_fd](#)

### 8.16.1 Function Documentation

#### 8.16.1.1 [int](#) [acc\\_Destroy](#) ([acc\\_p](#) [acc](#))

Definition at line 82 of file [acc.c](#).

References [MC\\_SUCCESS](#).

Referenced by [mc\\_platform\\_Destroy\(\)](#).

#### 8.16.1.2 [acc\\_p](#) [acc\\_Initialize](#) (struct [mc\\_platform\\_s](#) \* [mc\\_platform](#))

Definition at line 66 of file [acc.c](#).

References [COND\\_INIT](#), [COND\\_T](#), [MUTEX\\_INIT](#), and [MUTEX\\_T](#).

Referenced by [mc\\_platform\\_Initialize\(\)](#).

#### 8.16.1.3 [void](#)\* [acc\\_MessageHandlerThread](#) ([void](#) \* [arg](#))

Definition at line 94 of file [acc.c](#).

References `agent_Initialize()`, `mc_platform_s::agent_queue`, `agent_s::agent_status`, `AGENT_UPDATE`, `mc_platform_s::ams`, `mc_platform_s::asm_message_queue`, `CANCEL`, `COND_BROADCAST`, `COND_SIGNAL`, `COND_WAIT`, `agent_s::datastate`, `ENCRYPTED_DATA`, `ENCRYPTION_INITIALIZE`, `FIPA_ACL`, `message_s::from_address`, `mc_platform_s::giant`, `mc_platform_s::giant_cond`, `mc_platform_s::giant_lock`, `agent_s::lock`, `MC_AGENT_NEUTRAL`, `mc_platform`, `MC_RECV_AGENT`, `MC_RECV_MESSAGE`, `MC_RECV_RETURN`, `mc_platform_s::MC_signal`, `mc_platform_s::MC_signal_cond`, `mc_platform_s::MC_signal_lock`, `message_Destroy()`, `mc_platform_s::message_queue`, `message_Send()`, `message_s::message_type`, `MOBILE_AGENT`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `N_UNDRSTD`, `agent_s::name`, `agent_datastate_s::persistent`, `QUER_IF`, `QUER_REF`, `mc_platform_s::quit`, `mc_platform_s::quit_lock`, `RELAY`, `REQUEST`, `REQUEST_ENCRYPTION_INITIALIZE`, `RETURN_MSG`, `SUBSCRIBE`, `message_s::to_address`, and `WARN`.

Referenced by `acc_Start()`.

#### 8.16.1.4 `void acc_Start (struct mc_platform_s * mc_platform)`

#### 8.16.1.5 `void* acc_Thread (void * arg)`

Definition at line 328 of file `acc.c`.

References `agent_mailbox_Post()`, `mc_platform_s::agent_queue`, `AGENT_UPDATE`, `mc_platform_s::asm_message_queue`, `CANCEL`, `COND_BROADCAST`, `COND_WAIT`, `connection_Destroy()`, `mc_platform_s::connection_queue`, `mtp_http_s::content`, `mtp_http_content_s::data`, `ENCRYPTED_DATA`, `ENCRYPTION_INITIALIZE`, `FIPA_ACL`, `fipa_acl_envelope_Destroy()`, `fipa_acl_envelope_New()`, `fipa_acl_message_Destroy()`, `fipa_acl_message_New()`, `fipa_acl_Parse()`, `fipa_agent_identifier_set_s::fipa_agent_identifiers`, `fipa_envelope_Parse()`, `fipa_message_string_Destroy()`, `fipa_message_string_New()`, `mc_platform_s::giant`, `mc_platform_s::giant_cond`, `mc_platform_s::giant_lock`, `mtp_http_s::http_performative`, `HTTP_POST`, `HTTP_PUT`, `agent_s::mailbox`, `mc_platform`, `MC_RECV_CONNECTION`, `mc_platform_s::MC_signal`, `mc_platform_s::MC_signal_cond`, `mc_platform_s::MC_signal_lock`, `fipa_message_string_s::message`, `message_s::message_body`, `message_Destroy()`, `message_New()`, `mtp_http_s::message_parts`, `mc_platform_s::message_queue`, `message_s::message_type`, `message_xml_parse()`, `MOBILE_AGENT`, `mtp_http_Destroy()`, `mtp_http_InitializeFromConnection()`, `mtp_http_New()`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `MXML_NO_CALLBACK`, `mxmlLoadString()`, `N_UNDRSTD`, `fipa_agent_identifier_s::name`, `fipa_agent_identifier_set_s::num`, `fipa_acl_envelope_s::num_params`, `fipa_acl_envelope_s::params`, `fipa_message_string_s::parse`, `QUER_IF`, `QUER_REF`, `mc_platform_s::quit`, `mc_platform_s::quit_lock`, `RELAY`, `REQUEST`, `REQUEST_ENCRYPTION_INITIALIZE`, `RETURN_MSG`, `SUBSCRIBE`, `mtp_http_s::target`, `fipa_acl_Param_s::to`, `WARN`, and `message_s::xml_root`.

Referenced by `acc_Start()`.

#### 8.16.1.6 `void* listen_Thread (void * arg)`

Definition at line 583 of file `acc.c`.

References `mc_platform_s::acc`, `BACKLOG`, `CHECK_NULL`, `COND_BROADCAST`, `mc_platform_s::connection_queue`, `mc_platform`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, and `mc_platform_s::port`.

Referenced by `acc_Start()`.

**8.16.1.7** **STRUCT** (acc, struct mc\_platform\_s \*mc\_platform;THREAD\_T thread;THREAD\_T message\_handler\_thread;THREAD\_T listen\_thread;int waiting;MUTEX\_T \*waiting\_lock;COND\_T \*waiting\_cond;)

## **8.16.2 Variable Documentation**

### **8.16.2.1 struct sockaddr\_in\* addr**

Definition at line 64 of file acc.h.

### **8.16.2.2 unsigned long int client\_fd**

Definition at line 65 of file acc.h.

Referenced by main().

### **8.16.2.3 u\_long port**

Definition at line 63 of file acc.h.

Referenced by MC\_AclSend(), and message\_Send().

### **8.16.2.4 unsigned long int server\_fd**

Definition at line 66 of file acc.h.

Referenced by main().

## 8.17 /home/dko/projects/mobilec/trunk/src/include/agent.h File Reference

```
#include "agent_datastate.h"
#include "agent_task.h"
#include "libmc.h"
#include "message.h"
#include "macros.h"
#include "data_structures.h"
#include "agent_mailbox.h"
```

### Data Structures

- struct [agent\\_s](#)

### Functions

- [int agent\\_AddPersistentVariable](#) ([agent\\_p](#) agent, [int](#) task\_num, const char \*var\_name)
- [agent\\_p agent\\_New](#) (void)
- [agent\\_p agent\\_Copy](#) (const [agent\\_p](#) agent)
- [int agent\\_Destroy](#) ([agent\\_p](#) agent)
- [agent\\_p agent\\_Initialize](#) (struct [mc\\_platform\\_s](#) \*mc\_platform, [message\\_p](#) message, [int](#) id)
- void [agent\\_RunChScript](#) ([MCAgent\\_t](#) agent, struct [mc\\_platform\\_s](#) \*global)
- void \* [agent\\_RunChScriptThread](#) (void \*agent)

### 8.17.1 Function Documentation

#### 8.17.1.1 [int agent\\_AddPersistentVariable](#) ([agent\\_p](#) agent, [int](#) task\_num, const char \* var\_name)

Definition at line 49 of file agent.c.

References [agent\\_s::agent\\_interp](#), [agent\\_task\\_s::agent\\_variable\\_list](#), [interpreter\\_variable\\_data\\_s::array\\_dim](#), [interpreter\\_variable\\_data\\_s::array\\_extent](#), [CH\\_DATATYPE\\_SIZE](#), [CHECK\\_NULL](#), [interpreter\\_variable\\_data\\_s::data](#), [interpreter\\_variable\\_data\\_s::data\\_type](#), [agent\\_s::datastate](#), [MC\\_ERR](#), [agent\\_s::name](#), [interpreter\\_variable\\_data\\_s::name](#), [interpreter\\_variable\\_data\\_s::size](#), [size](#), [agent\\_datastate\\_s::task\\_progress](#), and [agent\\_datastate\\_s::tasks](#).

#### 8.17.1.2 [agent\\_p agent\\_Copy](#) (const [agent\\_p](#) agent)

Definition at line 131 of file agent.c.

References [agent\\_datastate\\_Copy\(\)](#), [agent\\_s::agent\\_interp](#), [agent\\_s::agent\\_persistent](#), [agent\\_s::agent\\_status](#), [agent\\_s::agent\\_thread](#), [agent\\_s::agent\\_type](#), [agent\\_s::arrival\\_time](#), [agent\\_s::datastate](#), [agent\\_s::home](#), [agent\\_s::home\\_port](#), [agent\\_s::id](#), [agent\\_s::lock](#), [MUTEX\\_INIT](#), [MUTEX\\_LOCK](#), [MUTEX\\_T](#), [agent\\_s::name](#), [agent\\_s::orphan](#), [agent\\_s::owner](#), [agent\\_s::return\\_data](#), and [agent\\_s::run\\_lock](#).

Referenced by [MC\\_CopyAgent\(\)](#).

**8.17.1.3 int agent\_Destroy (agent\_p agent)**

Definition at line 322 of file agent.c.

References agent\_datastate\_Destroy(), agent\_s::agent\_interp, agent\_mailbox\_Destroy(), agent\_s::agent\_status, agent\_s::agent\_thread, agent\_s::datastate, agent\_s::home, agent\_s::lock, agent\_s::mailbox, MC\_AGENT\_NEUTRAL, MC\_SUCCESS, MUTEX\_DESTROY, MUTEX\_LOCK, agent\_s::name, agent\_s::owner, and agent\_s::run\_lock.

Referenced by agent\_Initialize().

**8.17.1.4 agent\_p agent\_Initialize (struct mc\_platform\_s \* mc\_platform, message\_p message, int id)**

Definition at line 216 of file agent.c.

References agent\_datastate\_New(), agent\_Destroy(), agent\_mailbox\_New(), agent\_s::agent\_pipe\_active, agent\_s::agent\_pipe\_ready\_to\_read, agent\_s::agent\_ready\_to\_send, agent\_s::agent\_script\_ready, agent\_s::agent\_status, agent\_s::agent\_thread, agent\_s::agent\_thread\_id, agent\_s::agent\_type, message\_s::agent\_xml\_flag, agent\_xml\_parse(), agent\_s::arrival\_time, agent\_s::datastate, mc\_platform\_s::default\_agentstatus, mc\_platform\_s::err, agent\_s::id, agent\_s::lock, agent\_s::mailbox, MC\_ERR\_PARSE, agent\_s::mc\_platform, MC\_REMOTE\_AGENT, MC\_RETURN\_AGENT, MC\_WAIT\_CH, message\_s::message\_type, MOBILE\_AGENT, MUTEX\_DESTROY, MUTEX\_INIT, MUTEX\_T, agent\_s::orphan, RETURN\_MSG, agent\_s::run\_lock, agent\_datastate\_s::xml\_agent\_root, message\_s::xml\_payload, message\_s::xml\_root, and agent\_datastate\_s::xml\_root.

Referenced by acc\_MessageHandlerThread().

**8.17.1.5 agent\_p agent\_New (void)**

Definition at line 193 of file agent.c.

References agent\_s::lock, MUTEX\_INIT, MUTEX\_NEW, and agent\_s::run\_lock.

Referenced by MC\_ComposeAgent().

**8.17.1.6 void agent\_RunChScript (MCagent\_t agent, struct mc\_platform\_s \* global)****8.17.1.7 void\* agent\_RunChScriptThread (void \* agent)**

Definition at line 397 of file agent.c.

References agent\_s::agent\_interp, mc\_platform\_s::ams, COND\_SIGNAL, agent\_s::datastate, agent\_s::id, mc\_platform\_s::interp\_options, interpreter\_variable\_data\_Destroy(), interpreter\_variable\_data\_Initialize(), interpreter\_variable\_data\_InitializeFromAgent(), MC\_AclAddReceiver\_chdl(), MC\_AclAddReplyTo\_chdl(), MC\_AclDestroy\_chdl(), MC\_AclNew\_chdl(), MC\_AclPost\_chdl(), MC\_AclReply\_chdl(), MC\_AclRetrieve\_chdl(), MC\_AclSend\_chdl(), MC\_AclSetContent\_chdl(), MC\_AclSetPerformative\_chdl(), MC\_AclSetSender\_chdl(), MC\_AclWaitRetrieve\_chdl(), MC\_AddAgent\_chdl(), MC\_AGENT\_NEUTRAL, MC\_AgentVariableRetrieve\_chdl(), MC\_AgentVariableSave\_chdl(), MC\_Barrier\_chdl(), MC\_BarrierDelete\_chdl(), MC\_BarrierInit\_chdl(), MC\_CallAgentFunc\_chdl(), MC\_ComposeAgent\_chdl(), MC\_CondBroadcast\_chdl(), MC\_CondReset\_chdl(), MC\_CondSignal\_chdl(), MC\_CondWait\_chdl(), MC\_DeleteAgent\_chdl(), MC\_DeregisterService\_chdl(), MC\_DestroyServiceSearchResult\_chdl(), MC\_End\_chdl(), MC\_EXEC\_AGENT, MC\_FindAgentByID\_chdl(), MC\_FindAgentByName\_chdl(), MC\_GetAgentID\_chdl(), MC\_GetAgentName\_chdl(), MC\_GetAgentStatus\_chdl(), MC\_GetAgentXMLString\_chdl(), MC\_GetTimeOfDay\_chdl(), MC\_HaltAgency\_chdl(), MC\_MutexLock\_chdl(), MC\_MutexUnlock\_chdl(),

agent\_s::mc\_platform, mc\_platform, MC\_PrintAgentCode\_chdl(), MC\_RegisterService\_chdl(), MC\_ResumeAgency\_chdl(), MC\_RetrieveAgent\_chdl(), MC\_RetrieveAgentCode\_chdl(), MC\_RETURN\_AGENT, MC\_SearchForService\_chdl(), MC\_SemaphorePost\_chdl(), MC\_SemaphoreWait\_chdl(), MC\_SendAgentMigrationMessage\_chdl(), MC\_SendAgentMigrationMessageFile\_chdl(), MC\_SendSteerCommand\_chdl(), MC\_SetAgentStatus\_chdl(), MC\_SetDefaultAgentStatus\_chdl(), mc\_platform\_s::MC\_signal, mc\_platform\_s::MC\_signal\_cond, mc\_platform\_s::MC\_signal\_lock, MC\_SyncDelete\_chdl(), MC\_SyncInit\_chdl(), MC\_TerminateAgent\_chdl(), MC\_WAIT\_FINISHED, MC\_WAIT\_MESSGSEND, MUTEX\_LOCK, MUTEX\_UNLOCK, agent\_s::name, SIGNAL, and agent\_datastate\_s::task\_progress.

Referenced by agent\_RunChScript().

## 8.18 /home/dko/projects/mobilec/trunk/src/include/agent\_datastate.h File Reference

```
#include <mxml.h>
#include "agent_task.h"
```

### Data Structures

- struct [agent\\_datastate\\_s](#)

### Typedefs

- typedef struct [agent\\_datastate\\_s](#) [agent\\_datastate\\_t](#)
- typedef [agent\\_datastate\\_t](#) \* [agent\\_datastate\\_p](#)

### Functions

- [agent\\_datastate\\_p](#) [agent\\_datastate\\_Copy](#) (const [agent\\_datastate\\_p](#) *datastate*)
- [agent\\_datastate\\_p](#) [agent\\_datastate\\_New](#) (void)
- int [agent\\_datastate\\_Destroy](#) ([agent\\_datastate\\_p](#) *agent\_datastate*)

#### 8.18.1 Typedef Documentation

##### 8.18.1.1 typedef [agent\\_datastate\\_t](#)\* [agent\\_datastate\\_p](#)

Definition at line 66 of file [agent\\_datastate.h](#).

##### 8.18.1.2 typedef struct [agent\\_datastate\\_s](#) [agent\\_datastate\\_t](#)

#### 8.18.2 Function Documentation

##### 8.18.2.1 [agent\\_datastate\\_p](#) [agent\\_datastate\\_Copy](#) (const [agent\\_datastate\\_p](#) *datastate*)

Definition at line 41 of file [agent\\_datastate.c](#).

References [agent\\_datastate\\_s::agent\\_code](#), [agent\\_datastate\\_s::agent\\_code\\_ids](#), [agent\\_datastate\\_s::agent\\_codes](#), [agent\\_datastate\\_New\(\)](#), [agent\\_task\\_Copy\(\)](#), [agent\\_datastate\\_s::init\\_agent\\_status](#), [agent\\_datastate\\_s::number\\_of\\_tasks](#), [agent\\_datastate\\_s::persistent](#), [agent\\_datastate\\_s::return\\_data](#), [agent\\_datastate\\_s::task\\_progress](#), and [agent\\_datastate\\_s::tasks](#).

Referenced by [agent\\_Copy\(\)](#).

##### 8.18.2.2 int [agent\\_datastate\\_Destroy](#) ([agent\\_datastate\\_p](#) *agent\_datastate*)

Definition at line 129 of file [agent\\_datastate.c](#).

References [agent\\_datastate\\_s::agent\\_code\\_ids](#), [agent\\_datastate\\_s::agent\\_codes](#), [agent\\_task\\_Destroy\(\)](#), [MC\\_SUCCESS](#), [mxmlDelete\(\)](#), [agent\\_datastate\\_s::number\\_of\\_tasks](#), [agent\\_datastate\\_s::tasks](#), and [agent\\_datastate\\_s::xml\\_root](#).



Referenced by agent\_Destroy().

### 8.18.2.3 agent\_datastate\_p agent\_datastate\_New (void)

Definition at line 109 of file agent\_datastate.c.

References agent\_datastate\_s::agent\_code, CHECK\_NULL, agent\_datastate\_s::init\_agent\_status, agent\_datastate\_s::number\_of\_tasks, agent\_datastate\_s::persistent, agent\_datastate\_s::return\_data, agent\_datastate\_s::task\_progress, agent\_datastate\_s::tasks, agent\_datastate\_s::xml\_agent\_root, and agent\_datastate\_s::xml\_root.

Referenced by agent\_datastate\_Copy(), agent\_Initialize(), and MC\_ComposeAgent().

## 8.19 /home/dko/projects/mobilec/trunk/src/include/agent\_lib.h File Reference

### Functions

- EXPORTCH [int MC\\_AclDestroy\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclNew\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclPost\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclReply\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclRetrieve\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AclSend\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclWaitRetrieve\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclSetPerformative\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclSetSender\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclAddReceiver\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclAddReplyTo\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclSetContent\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AddAgent\\_chdl](#) (void \*varg)
- EXPORTCH const void \* [MC\\_AgentVariableRetrieve\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AgentVariableSave\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_Barrier\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_BarrierDelete\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_BarrierInit\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_CallAgentFunc\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_ComposeAgent\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_CondBroadcast\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_CondWait\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_CondReset\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_CondSignal\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_DeleteAgent\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_DeregisterService\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_DestroyServiceSearchResult\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_End\\_chdl](#) (void \*varg)
- EXPORTCH [MCAgent\\_t MC\\_FindAgentByID\\_chdl](#) (void \*varg)
- EXPORTCH [MCAgent\\_t MC\\_FindAgentByName\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_GetAgentID\\_chdl](#) (void \*varg)
- EXPORTCH char \* [MC\\_GetAgentName\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_GetAgentNumTasks\\_chdl](#) (void \*varg)
- EXPORTCH char \* [MC\\_GetAgentXMLString\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_GetTimeOfDay\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_HaltAgency\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_PrintAgentCode\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_MutexLock\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_MutexUnlock\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_RegisterService\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_ResumeAgency\\_chdl](#) (void \*varg)
- EXPORTCH [MCAgent\\_t MC\\_RetrieveAgent\\_chdl](#) (void \*varg)
- EXPORTCH char \* [MC\\_RetrieveAgentCode\\_chdl](#) (void \*varg)
- EXPORTCH char \* [MC\\_SearchForService\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SemaphorePost\\_chdl](#) (void \*varg)

- EXPORTCH [int MC\\_SemaphoreWait\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SendSteerCommand\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_TerminateAgent\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_GetAgentStatus\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SendAgentMigrationMessage\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SendAgentMigrationMessageFile\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SetAgentStatus\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SetDefaultAgentStatus\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SyncDelete\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SyncInit\\_chdl](#) (void \*varg)

## 8.19.1 Function Documentation

### 8.19.1.1 EXPORTCH void\* MC\_AclAddReceiver\_chdl (void \* *varg*)

Definition at line 1851 of file libmc.c.

References [MC\\_AclAddReceiver\(\)](#).

Referenced by [agent\\_RunChScriptThread\(\)](#).

### 8.19.1.2 EXPORTCH void\* MC\_AclAddReplyTo\_chdl (void \* *varg*)

Definition at line 1871 of file libmc.c.

References [MC\\_AclAddReplyTo\(\)](#).

Referenced by [agent\\_RunChScriptThread\(\)](#).

### 8.19.1.3 EXPORTCH int MC\_AclDestroy\_chdl (void \* *varg*)

Definition at line 1698 of file libmc.c.

References [MC\\_AclDestroy\(\)](#).

Referenced by [agent\\_RunChScriptThread\(\)](#).

### 8.19.1.4 EXPORTCH void\* MC\_AclNew\_chdl (void \* *varg*)

Definition at line 1713 of file libmc.c.

References [MC\\_AclNew\(\)](#).

Referenced by [agent\\_RunChScriptThread\(\)](#).

### 8.19.1.5 EXPORTCH void\* MC\_AclPost\_chdl (void \* *varg*)

Definition at line 1721 of file libmc.c.

References [MC\\_AclPost\(\)](#).

Referenced by [agent\\_RunChScriptThread\(\)](#).

**8.19.1.6 EXPORTCH void\* MC\_AclReply\_chdl (void \* *varg*)**

Definition at line 1739 of file libmc.c.

References MC\_AclReply().

Referenced by agent\_RunChScriptThread().

**8.19.1.7 EXPORTCH void\* MC\_AclRetrieve\_chdl (void \* *varg*)**

Definition at line 1755 of file libmc.c.

References MC\_AclRetrieve().

Referenced by agent\_RunChScriptThread().

**8.19.1.8 EXPORTCH int MC\_AclSend\_chdl (void \* *varg*)**

Definition at line 1771 of file libmc.c.

References CHECK\_NULL, MC\_AclSend(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

**8.19.1.9 EXPORTCH void\* MC\_AclSetContent\_chdl (void \* *varg*)**

Definition at line 1891 of file libmc.c.

References fipa\_acl\_message\_s::content, and MC\_AclSetContent().

Referenced by agent\_RunChScriptThread().

**8.19.1.10 EXPORTCH void\* MC\_AclSetPerformative\_chdl (void \* *varg*)**

Definition at line 1813 of file libmc.c.

References MC\_AclSetPerformative(), and fipa\_acl\_message\_s::performative.

Referenced by agent\_RunChScriptThread().

**8.19.1.11 EXPORTCH void\* MC\_AclSetSender\_chdl (void \* *varg*)**

Definition at line 1831 of file libmc.c.

References MC\_AclSetSender().

Referenced by agent\_RunChScriptThread().

**8.19.1.12 EXPORTCH void\* MC\_AclWaitRetrieve\_chdl (void \* *varg*)**

Definition at line 1795 of file libmc.c.

References MC\_AclWaitRetrieve().

Referenced by agent\_RunChScriptThread().

**8.19.1.13 EXPORTCH int MC\_AddAgent\_chdl (void \* *varg*)**

Definition at line 1911 of file libmc.c.

References CHECK\_NULL, MC\_AddAgent(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

**8.19.1.14 EXPORTCH const void\* MC\_AgentVariableRetrieve\_chdl (void \* *varg*)**

Definition at line 1935 of file libmc.c.

References MC\_AgentVariableRetrieve().

Referenced by agent\_RunChScriptThread().

**8.19.1.15 EXPORTCH int MC\_AgentVariableSave\_chdl (void \* *varg*)**

Definition at line 1959 of file libmc.c.

References MC\_AgentVariableSave().

Referenced by agent\_RunChScriptThread().

**8.19.1.16 EXPORTCH int MC\_Barrier\_chdl (void \* *varg*)**

Definition at line 2011 of file libmc.c.

References CHECK\_NULL, MC\_Barrier(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

**8.19.1.17 EXPORTCH int MC\_BarrierDelete\_chdl (void \* *varg*)**

Definition at line 2034 of file libmc.c.

References CHECK\_NULL, MC\_BarrierDelete(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

**8.19.1.18 EXPORTCH int MC\_BarrierInit\_chdl (void \* *varg*)**

Definition at line 2057 of file libmc.c.

References CHECK\_NULL, MC\_BarrierInit(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

**8.19.1.19 EXPORTCH int MC\_CallAgentFunc\_chdl (void \* *varg*)**

Definition at line 1981 of file libmc.c.

References MC\_CallAgentFunc().

Referenced by agent\_RunChScriptThread().

**8.19.1.20 EXPORTCH int MC\_ComposeAgent\_chdl (void \* *varg*)**

Definition at line 2105 of file libmc.c.

References MC\_ComposeAgent().

Referenced by agent\_RunChScriptThread().

**8.19.1.21 EXPORTCH int MC\_CondBroadcast\_chdl (void \* *varg*)**

Definition at line 2082 of file libmc.c.

References CHECK\_NULL, MC\_CondBroadcast(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

**8.19.1.22 EXPORTCH int MC\_CondReset\_chdl (void \* *varg*)**

Definition at line 2168 of file libmc.c.

References CHECK\_NULL, MC\_CondReset(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

**8.19.1.23 EXPORTCH int MC\_CondSignal\_chdl (void \* *varg*)**

Definition at line 2145 of file libmc.c.

References CHECK\_NULL, MC\_CondSignal(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

**8.19.1.24 EXPORTCH int MC\_CondWait\_chdl (void \* *varg*)**

Definition at line 2191 of file libmc.c.

References CHECK\_NULL, MC\_CondWait(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

**8.19.1.25 EXPORTCH int MC\_DeleteAgent\_chdl (void \* *varg*)**

Definition at line 2213 of file libmc.c.

References MC\_DeleteAgent().

Referenced by agent\_RunChScriptThread().

**8.19.1.26 EXPORTCH int MC\_DeregisterService\_chdl (void \* *varg*)**

Definition at line 2255 of file libmc.c.

References CHECK\_NULL, MC\_DeregisterService(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

**8.19.1.27 EXPORTCH int MC\_DestroyServiceSearchResult\_chdl (void \* *varg*)**

Definition at line 2228 of file libmc.c.

References MC\_DestroyServiceSearchResult().

Referenced by agent\_RunChScriptThread().

**8.19.1.28 EXPORTCH int MC\_End\_chdl (void \* *varg*)**

Definition at line 2282 of file libmc.c.

References CHECK\_NULL, MC\_End(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

**8.19.1.29 EXPORTCH MCAgent\_t MC\_FindAgentByID\_chdl (void \* *varg*)**

Definition at line 2299 of file libmc.c.

References CHECK\_NULL, MC\_FindAgentByID(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

**8.19.1.30 EXPORTCH MCAgent\_t MC\_FindAgentByName\_chdl (void \* *varg*)**

Definition at line 2322 of file libmc.c.

References CHECK\_NULL, MC\_FindAgentByName(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

**8.19.1.31 EXPORTCH int MC\_GetAgentID\_chdl (void \* *varg*)**

Definition at line 2367 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.19.1.32 EXPORTCH char\* MC\_GetAgentName\_chdl (void \* *varg*)**

Definition at line 2383 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.19.1.33 EXPORTCH int MC\_GetAgentNumTasks\_chdl (void \* *varg*)**

Definition at line 2399 of file libmc.c.

**8.19.1.34 EXPORTCH int MC\_GetAgentStatus\_chdl (void \* *varg*)**

Definition at line 2415 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.19.1.35 EXPORTCH char\* MC\_GetAgentXMLString\_chdl (void \* *varg*)**

Definition at line 2431 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.19.1.36 EXPORTCH int MC\_GetTimeOfDay\_chdl (void \* *varg*)**

Definition at line 2447 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.19.1.37 EXPORTCH int MC\_HaltAgency\_chdl (void \* *varg*)**

Definition at line 2462 of file libmc.c.

References CHECK\_NULL, MC\_HaltAgency(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

**8.19.1.38 EXPORTCH int MC\_MutexLock\_chdl (void \* *varg*)**

Definition at line 2480 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.19.1.39 EXPORTCH int MC\_MutexUnlock\_chdl (void \* *varg*)**

Definition at line 2504 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.19.1.40 EXPORTCH int MC\_PrintAgentCode\_chdl (void \* *varg*)**

Definition at line 2528 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.19.1.41 EXPORTCH int MC\_RegisterService\_chdl (void \* *varg*)**

Definition at line 2544 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.19.1.42 EXPORTCH int MC\_ResumeAgency\_chdl (void \* *varg*)**

Definition at line 2581 of file libmc.c.

Referenced by agent\_RunChScriptThread().



**8.19.1.43 EXPORTCH MC\_Agent\_t MC\_RetrieveAgent\_chdl (void \* *varg*)**

Definition at line 2599 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.19.1.44 EXPORTCH char\* MC\_RetrieveAgentCode\_chdl (void \* *varg*)**

Definition at line 2616 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.19.1.45 EXPORTCH char\* MC\_SearchForService\_chdl (void \* *varg*)**

Definition at line 2632 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.19.1.46 EXPORTCH int MC\_SemaphorePost\_chdl (void \* *varg*)**

Definition at line 2673 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.19.1.47 EXPORTCH int MC\_SemaphoreWait\_chdl (void \* *varg*)**

Definition at line 2697 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.19.1.48 EXPORTCH int MC\_SendAgentMigrationMessage\_chdl (void \* *varg*)**

Definition at line 2721 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.19.1.49 EXPORTCH int MC\_SendAgentMigrationMessageFile\_chdl (void \* *varg*)**

Definition at line 2746 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.19.1.50 EXPORTCH int MC\_SendSteerCommand\_chdl (void \* *varg*)**

Definition at line 2766 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.19.1.51 EXPORTCH int MC\_SetAgentStatus\_chdl (void \* *varg*)**

Definition at line 2789 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.19.1.52 EXPORTCH int MC\_SetDefaultAgentStatus\_chdl (void \* *varg*)**

Definition at line 2807 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.19.1.53 EXPORTCH int MC\_SyncDelete\_chdl (void \* *varg*)**

Definition at line 2830 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.19.1.54 EXPORTCH int MC\_SyncInit\_chdl (void \* *varg*)**

Definition at line 2853 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.19.1.55 EXPORTCH int MC\_TerminateAgent\_chdl (void \* *varg*)**

Definition at line 2877 of file libmc.c.

Referenced by agent\_RunChScriptThread().

## 8.20 /home/dko/projects/mobilec/trunk/src/include/agent\_mailbox.h File Reference

### Data Structures

- struct [agent\\_mailbox\\_s](#)

### Typedefs

- typedef struct [agent\\_mailbox\\_s](#) [agent\\_mailbox\\_t](#)
- typedef [agent\\_mailbox\\_t](#) \* [agent\\_mailbox\\_p](#)

### Functions

- [agent\\_mailbox\\_p](#) [agent\\_mailbox\\_New](#) (void)
- [agent\\_mailbox\\_p](#) [agent\\_mailbox\\_Copy](#) ([agent\\_mailbox\\_p](#) src)
- int [agent\\_mailbox\\_Destroy](#) ([agent\\_mailbox\\_t](#) \*mailbox)
- int [agent\\_mailbox\\_Post](#) ([agent\\_mailbox\\_p](#) mailbox, [fipa\\_acl\\_message\\_t](#) \*message)
- [fipa\\_acl\\_message\\_t](#) \* [agent\\_mailbox\\_Retrieve](#) ([agent\\_mailbox\\_p](#) mailbox)
- [fipa\\_acl\\_message\\_t](#) \* [agent\\_mailbox\\_WaitRetrieve](#) ([agent\\_mailbox\\_p](#) mailbox)

### 8.20.1 Typedef Documentation

#### 8.20.1.1 typedef [agent\\_mailbox\\_t](#)\* [agent\\_mailbox\\_p](#)

Definition at line 12 of file [agent\\_mailbox.h](#).

#### 8.20.1.2 typedef struct [agent\\_mailbox\\_s](#) [agent\\_mailbox\\_t](#)

### 8.20.2 Function Documentation

#### 8.20.2.1 [agent\\_mailbox\\_p](#) [agent\\_mailbox\\_Copy](#) ([agent\\_mailbox\\_p](#) src)

Definition at line 15 of file [agent\\_mailbox.c](#).

References [agent\\_mailbox\\_New\(\)](#), and [agent\\_mailbox\\_s::mail\\_queue](#).

#### 8.20.2.2 int [agent\\_mailbox\\_Destroy](#) ([agent\\_mailbox\\_t](#) \* mailbox)

Definition at line 23 of file [agent\\_mailbox.c](#).

References [agent\\_mailbox\\_s::mail\\_queue](#).

Referenced by [agent\\_Destroy\(\)](#).

#### 8.20.2.3 [agent\\_mailbox\\_p](#) [agent\\_mailbox\\_New](#) (void)

Definition at line 6 of file [agent\\_mailbox.c](#).

References [agent\\_mailbox\\_s::mail\\_queue](#).

Referenced by `agent_Initialize()`, and `agent_mailbox_Copy()`.

#### **8.20.2.4** `int agent_mailbox_Post (agent_mailbox_p mailbox, fipa_acl_message_t * message)`

Definition at line 32 of file `agent_mailbox.c`.

References `agent_mailbox_s::mail_queue`.

Referenced by `acc_Thread()`, and `MC_AclPost()`.

#### **8.20.2.5** `fipa_acl_message_t* agent_mailbox_Retrieve (agent_mailbox_p mailbox)`

Definition at line 38 of file `agent_mailbox.c`.

References `agent_mailbox_s::mail_queue`.

Referenced by `agent_mailbox_WaitRetrieve()`, and `MC_AclRetrieve()`.

#### **8.20.2.6** `fipa_acl_message_t* agent_mailbox_WaitRetrieve (agent_mailbox_p mailbox)`

Definition at line 43 of file `agent_mailbox.c`.

References `agent_mailbox_Retrieve()`, `COND_WAIT`, `agent_mailbox_s::mail_queue`, `MUTEX_LOCK`, and `MUTEX_UNLOCK`.

Referenced by `MC_AclWaitRetrieve()`.

## 8.21 /home/dko/projects/mobilec/trunk/src/include/agent\_task.h File Reference

```
#include "interpreter_variable_data.h"
#include "data_structures.h"
```

### Data Structures

- struct [agent\\_task\\_s](#)

### Typedefs

- typedef struct [agent\\_task\\_s](#) [agent\\_task\\_t](#)
- typedef [agent\\_task\\_t](#) \* [agent\\_task\\_p](#)

### Functions

- [agent\\_task\\_p](#) [agent\\_task\\_New](#) (void)
- [agent\\_task\\_p](#) [agent\\_task\\_Copy](#) ([agent\\_task\\_p](#) task)
- int [agent\\_task\\_Destroy](#) ([agent\\_task\\_p](#) agent\_task)

#### 8.21.1 Typedef Documentation

##### 8.21.1.1 typedef [agent\\_task\\_t](#)\* [agent\\_task\\_p](#)

Definition at line 61 of file [agent\\_task.h](#).

##### 8.21.1.2 typedef struct [agent\\_task\\_s](#) [agent\\_task\\_t](#)

#### 8.21.2 Function Documentation

##### 8.21.2.1 [agent\\_task\\_p](#) [agent\\_task\\_Copy](#) ([agent\\_task\\_p](#) task)

Definition at line 60 of file [agent\\_task.c](#).

References [agent\\_task\\_s::agent\\_return\\_data](#), [agent\\_task\\_s::code\\_id](#), [agent\\_task\\_s::init\\_agent\\_status](#), [agent\\_task\\_s::number\\_of\\_elements](#), [agent\\_task\\_s::persistent](#), [agent\\_task\\_s::server\\_name](#), [agent\\_task\\_s::size\\_of\\_element\\_array](#), [agent\\_task\\_s::task\\_completed](#), and [agent\\_task\\_s::var\\_name](#).

Referenced by [agent\\_datastate\\_Copy](#)().

##### 8.21.2.2 int [agent\\_task\\_Destroy](#) ([agent\\_task\\_p](#) agent\_task)

Definition at line 102 of file [agent\\_task.c](#).

References [agent\\_task\\_s::agent\\_return\\_data](#), [agent\\_task\\_s::agent\\_variable\\_list](#), [agent\\_task\\_s::code\\_id](#), [interpreter\\_variable\\_data\\_Destroy](#)(), [MC\\_SUCCESS](#), [agent\\_task\\_s::saved\\_variables](#), [agent\\_task\\_s::server\\_name](#), and [agent\\_task\\_s::var\\_name](#).

Referenced by [agent\\_datastate\\_Destroy](#)().

### 8.21.2.3 `agent_task_p agent_task_New (void)`

Definition at line 42 of file `agent_task.c`.

References `agent_task_s::agent_variable_list`, `agent_task_s::num_saved_variables`, and `agent_task_s::saved_variables`.

Referenced by `agent_xml_parse__tasks()`, and `MC_ComposeAgent()`.

## 8.22 /home/dko/projects/mobilec/trunk/src/include/ams.h File Reference

```
#include "macros.h"
```

### Functions

- **STRUCT** (ams, struct [mc\\_platform\\_s](#) \*[mc\\_platform](#);MUTEX\_T \*[runflag\\_lock](#);COND\_T \*[runflag\\_cond](#);int [run](#);int [waiting](#);MUTEX\_T \*[waiting\\_lock](#);COND\_T \*[waiting\\_cond](#);THREAD\_T [thread](#);) ams\_p ams\_Initialize(struct [mc\\_platform\\_s](#) \*\_mc\_platform)
- int [ams\\_Destroy](#) (ams\_p ams)
- int [ams\\_RemoveFinishedAgents](#) (ams\_p ams)
- int [ams\\_ManageAgentList](#) (ams\_p ams)
- void [ams\\_Print](#) (ams\_p ams)
- void [ams\\_Start](#) (struct [mc\\_platform\\_s](#) \*[mc\\_platform](#))
- void \* [ams\\_Thread](#) (void \*arg)

### 8.22.1 Function Documentation

#### 8.22.1.1 int [ams\\_Destroy](#) (ams\_p *ams*)

Definition at line 41 of file [ams.c](#).

References [COND\\_DESTROY](#), [MC\\_SUCCESS](#), and [MUTEX\\_DESTROY](#).

Referenced by [mc\\_platform\\_Destroy\(\)](#).

#### 8.22.1.2 int [ams\\_ManageAgentList](#) (ams\_p *ams*)

Definition at line 111 of file [ams.c](#).

References [agent\\_RunChScript\(\)](#), [agent\\_s::agent\\_status](#), [ListSearch\(\)](#), [agent\\_s::lock](#), [MC\\_AGENT\\_ACTIVE](#), [MC\\_AGENT\\_NEUTRAL](#), [MC\\_WAIT\\_CH](#), [MC\\_WAIT\\_FINISHED](#), [MC\\_WAIT\\_MESSGSEND](#), [message\\_Destroy\(\)](#), [message\\_InitializeFromAgent\(\)](#), [message\\_New\(\)](#), [MUTEX\\_LOCK](#), [MUTEX\\_UNLOCK](#), and [agent\\_s::orphan](#).

Referenced by [ams\\_Thread\(\)](#).

#### 8.22.1.3 void [ams\\_Print](#) (ams\_p *ams*)

Definition at line 79 of file [ams.c](#).

References [agent\\_s::agent\\_status](#), [agent\\_s::connect\\_id](#), [agent\\_s::id](#), [ListSearch\(\)](#), [MUTEX\\_LOCK](#), and [MUTEX\\_UNLOCK](#).

#### 8.22.1.4 int [ams\\_RemoveFinishedAgents](#) (ams\_p *ams*)

#### 8.22.1.5 void [ams\\_Start](#) (struct [mc\\_platform\\_s](#) \* *mc\_platform*)

#### 8.22.1.6 void\* [ams\\_Thread](#) (void \* *arg*)

Definition at line 225 of file [ams.c](#).

References `mc_platform_s::ams`, `ams_ManageAgentList()`, `COND_BROADCAST`, `COND_WAIT`, `mc_platform`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `mc_platform_s::quit`, and `mc_platform_s::quit_lock`.

Referenced by `ams_Start()`.

**8.22.1.7** **STRUCT** (`ams`, `struct mc_platform_s *mc_platform`; `MUTEX_T *runflag_lock`; `COND_T *runflag_cond`; `int run`; `int waiting`; `MUTEX_T *waiting_lock`; `COND_T *waiting_cond`; `THREAD_T thread`;)



## 8.23 /home/dko/projects/mobilec/trunk/src/include/ap\_queue\_template.h File Reference

```
#include "macros.h"
#include "mc_error.h"
```

### Defines

- #define [AP\\_QUEUE\\_DECL\\_TEMPLATE](#)(name, node\_type)
- #define [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#)(name, func\_name, return\_type, search\_type) return\_type name##\_##\_func\_name(name##\_p name, const search\_type key);
- #define [AP\\_QUEUE\\_STD\\_DEFN\\_TEMPLATE](#)(name, node\_type)
- #define [AP\\_QUEUE\\_SEARCH\\_TEMPLATE](#)(name, func\_name, node\_type, search\_type, search\_expression)
- #define [AP\\_QUEUE\\_REMOVE\\_TEMPLATE](#)(name, func\_name, node\_type, search\_type, search\_expression)

### 8.23.1 Define Documentation

#### 8.23.1.1 #define AP\_QUEUE\_DECL\_TEMPLATE(name, node\_type)

##### Value:

```
typedef struct name##_s \
{ \
    int size; \
    list_p list; \
    MUTEX_T* lock; \
    COND_T* cond; \
} name##_t; \
\
typedef name##_t* name##_p; \
\
name##_p name##_New( void ); \
int name##_Destroy( name##_p name ); \
int name##_Add( name##_p name, struct node_type##_s* node ); \
name##_p name##_Copy( name##_p name ); \
struct node_type##_s* name##_Pop( name##_p name ); \
struct node_type##_s* name##_SearchIndex( name##_p name, int index ); \
int name##_RemoveIndex( name##_p name, int index );
```

Definition at line 40 of file ap\_queue\_template.h.

#### 8.23.1.2 #define AP\_QUEUE\_GENERIC\_DECL\_TEMPLATE(name, func\_name, return\_type, search\_type) return\_type name##\_##\_func\_name(name##\_p name, const search\_type key);

Definition at line 59 of file ap\_queue\_template.h.

#### 8.23.1.3 #define AP\_QUEUE\_REMOVE\_TEMPLATE(name, func\_name, node\_type, search\_type, search\_expression)

##### Value:

```

int name##_func_name( name##_p name, const search_type key ) \
{ \
    int err_code = MC_ERR_NOT_FOUND; \
    struct listNode_s* parsenode; \
    struct node_type##_s* node; \
    node = NULL; \
    \
    MUTEX_LOCK(name->lock); \
    if (name->list->listhead == NULL) { \
        MUTEX_UNLOCK(name->lock); \
        return MC_ERR_NOT_FOUND; \
    } \
    for( \
        parsenode = (listNode_t*)name->list->listhead; \
        parsenode->next != NULL; \
        parsenode = (listNode_t*)parsenode->next \
    ) \
    { \
        node = (node_type##_t*)parsenode->node_data; \
        if (search_expression) { \
            break; \
            err_code = MC_SUCCESS; \
        } \
    } \
    \
    MUTEX_UNLOCK(name->lock); \
    return err_code; \
}

```

Definition at line 203 of file ap\_queue\_template.h.

#### 8.23.1.4 #define AP\_QUEUE\_SEARCH\_TEMPLATE(name, func\_name, node\_type, search\_type, search\_expression)

##### Value:

```

struct node_type##_s* name##_func_name( name##_p name, const search_type key ) \
{ \
    listNode_t* parsenode; \
    struct node_type##_s* node; \
    struct node_type##_s* ret = NULL; \
    node = NULL; \
    \
    MUTEX_LOCK(name->lock); \
    if (name->list->listhead == NULL) { \
        MUTEX_UNLOCK(name->lock); \
        return NULL; \
    } \
    for( \
        parsenode = (listNode_t*)name->list->listhead; \
        parsenode != NULL; \
        parsenode = (listNode_t*)parsenode->next \
    ) \
    { \
        node = (node_type##_t*)parsenode->node_data; \
        if (search_expression){ \
            ret = node; \
            break; \
        } \
    } \
    \
    MUTEX_UNLOCK(name->lock); \
    return ret; \
}

```

Definition at line 173 of file ap\_queue\_template.h.

**8.23.1.5 #define AP\_QUEUE\_STD\_DEFN\_TEMPLATE(name, node\_type)**

Definition at line 62 of file ap\_queue\_template.h.

## 8.24 /home/dko/projects/mobilec/trunk/src/include/barrier.h File Reference

```
#include "macros.h"
#include "../mc_list/list.h"
#include "mc_rwlock.h"
```

### Data Structures

- struct [barrier\\_node\\_s](#)
- struct [barrier\\_queue\\_s](#)

### Typedefs

- typedef struct [barrier\\_node\\_s](#) [barrier\\_node\\_t](#)
- typedef [barrier\\_node\\_t](#) \* [barrier\\_node\\_p](#)
- typedef struct [barrier\\_queue\\_s](#) [barrier\\_queue\\_t](#)
- typedef [barrier\\_queue\\_t](#) \* [barrier\\_queue\\_p](#)

### Functions

- [barrier\\_node\\_p](#) [barrier\\_node\\_Initialize](#) (int id, int num\_registered)
- int [barrier\\_node\\_Destroy](#) ([barrier\\_node\\_p](#) node)
- int [barrier\\_queue\\_Add](#) ([barrier\\_queue\\_p](#) list, [barrier\\_node\\_p](#) node)
- int [barrier\\_queue\\_Delete](#) (int id, [barrier\\_queue\\_p](#) list)
- int [barrier\\_queue\\_Destroy](#) ([barrier\\_queue\\_p](#) queue)
- [barrier\\_node\\_p](#) [barrier\\_queue\\_Get](#) ([barrier\\_queue\\_p](#) list, int id)
- [barrier\\_queue\\_p](#) [barrier\\_queue\\_New](#) (void)
- [barrier\\_node\\_p](#) [barrier\\_queue\\_Pop](#) ([barrier\\_queue\\_p](#) queue)

#### 8.24.1 Typedef Documentation

##### 8.24.1.1 typedef [barrier\\_node\\_t](#)\* [barrier\\_node\\_p](#)

Definition at line 49 of file [barrier.h](#).

##### 8.24.1.2 typedef struct [barrier\\_node\\_s](#) [barrier\\_node\\_t](#)

##### 8.24.1.3 typedef [barrier\\_queue\\_t](#)\* [barrier\\_queue\\_p](#)

Definition at line 57 of file [barrier.h](#).

#### 8.24.1.4 typedef struct barrier\_queue\_s barrier\_queue\_t

### 8.24.2 Function Documentation

#### 8.24.2.1 int barrier\_node\_Destroy (barrier\_node\_p node)

Definition at line 65 of file barrier.c.

References barrier\_node\_s::cond, COND\_DESTROY, barrier\_node\_s::lock, MC\_SUCCESS, and MUTEX\_DESTROY.

Referenced by barrier\_queue\_Delete(), and barrier\_queue\_Destroy().

#### 8.24.2.2 barrier\_node\_p barrier\_node\_Initialize (int id, int num\_registered)

Definition at line 40 of file barrier.c.

References CHECK\_NULL, barrier\_node\_s::cond, COND\_INIT, COND\_T, barrier\_node\_s::id, barrier\_node\_s::lock, MUTEX\_INIT, MUTEX\_T, node, barrier\_node\_s::num\_registered, and barrier\_node\_s::num\_waiting.

Referenced by MC\_BarrierInit().

#### 8.24.2.3 int barrier\_queue\_Add (barrier\_queue\_p list, barrier\_node\_p node)

Definition at line 82 of file barrier.c.

References DATA, barrier\_node\_s::id, barrier\_queue\_s::list, ListAdd(), list\_s::listhead, barrier\_queue\_s::lock, MC\_SUCCESS, MC\_WARN\_DUPLICATE, listNode\_s::next, listNode\_s::node\_data, RWLOCK\_WRLOCK, RWLOCK\_WRUNLOCK, and barrier\_queue\_s::size.

Referenced by MC\_BarrierInit().

#### 8.24.2.4 int barrier\_queue\_Delete (int id, barrier\_queue\_p list)

Definition at line 106 of file barrier.c.

References barrier\_node\_Destroy(), barrier\_node\_s::id, barrier\_queue\_s::list, ListDelete(), ListSearch(), barrier\_queue\_s::lock, MC\_ERR\_NOT\_FOUND, MC\_SUCCESS, RWLOCK\_WRLOCK, RWLOCK\_WRUNLOCK, barrier\_queue\_s::size, and list\_s::size.

Referenced by MC\_BarrierDelete().

#### 8.24.2.5 int barrier\_queue\_Destroy (barrier\_queue\_p queue)

Definition at line 126 of file barrier.c.

References barrier\_node\_Destroy(), barrier\_queue\_Pop(), barrier\_queue\_s::list, ListTerminate(), barrier\_queue\_s::lock, MC\_SUCCESS, node, and RWLOCK\_DESTROY.

Referenced by mc\_platform\_Destroy().

#### 8.24.2.6 barrier\_node\_p barrier\_queue\_Get (barrier\_queue\_p list, int id)

Definition at line 140 of file barrier.c.

References `barrier_queue_s::list`, `list_s::listhead`, `barrier_queue_s::lock`, `listNode_s::next`, `listNode_s::node_data`, `RWLOCK_RDLOCK`, and `RWLOCK_RDUNLOCK`.

Referenced by `MC_Barrier()`, and `MC_BarrierInit()`.

#### **8.24.2.7 `barrier_queue_p barrier_queue_New (void)`**

Definition at line 157 of file `barrier.c`.

References `CHECK_NULL`, `barrier_queue_s::list`, `ListInitialize()`, `barrier_queue_s::lock`, `RWLOCK_INIT`, and `RWLOCK_T`.

Referenced by `mc_platform_Initialize()`.

#### **8.24.2.8 `barrier_node_p barrier_queue_Pop (barrier_queue_p queue)`**

Definition at line 171 of file `barrier.c`.

References `barrier_queue_s::list`, `ListPop()`, and `node`.

Referenced by `barrier_queue_Destroy()`.

## 8.25 /home/dko/projects/mobilec/trunk/src/include/cmd\_prompt.h File Reference

```
#include "macros.h"
```

### Data Structures

- struct [cmd\\_prompt\\_s](#)
- struct [command\\_s](#)

### Defines

- #define [CMDLINE\\_SIZE](#) 80

### Typedefs

- typedef struct [cmd\\_prompt\\_s](#) [cmd\\_prompt\\_t](#)
- typedef [cmd\\_prompt\\_t](#) \* [cmd\\_prompt\\_p](#)
- typedef struct [command\\_s](#) [command\\_t](#)
- typedef [command\\_t](#) \* [command\\_p](#)

### Functions

- [int cmd\\_prompt\\_Destroy](#) ([cmd\\_prompt\\_p](#) cmd\_prompt)
- [cmd\\_prompt\\_p cmd\\_prompt\\_Initialize](#) (struct [mc\\_platform\\_s](#) \*[mc\\_platform](#))
- [void cmd\\_prompt\\_Start](#) (struct [mc\\_platform\\_s](#) \*[mc\\_platform](#))
- [int split\\_string](#) (char \*\*\*args, const char \*buf)
- [int process\\_command](#) ([command\\_t](#) \*cmd)
- [void \\* cmd\\_prompt\\_Thread](#) (void \*arg)
- [int exec\\_command](#) ([command\\_t](#) cmd, struct [mc\\_platform\\_s](#) \*global)
- [int dealloc\\_command](#) ([command\\_t](#) \*cmd)

#### 8.25.1 Define Documentation

##### 8.25.1.1 #define CMDLINE\_SIZE 80

Definition at line 37 of file cmd\_prompt.h.

#### 8.25.2 Typedef Documentation

##### 8.25.2.1 typedef cmd\_prompt\_t\* cmd\_prompt\_p

Definition at line 44 of file cmd\_prompt.h.

**8.25.2.2** `typedef struct cmd_prompt_s cmd_prompt_t`

**8.25.2.3** `typedef command_t* command_p`

Definition at line 51 of file `cmd_prompt.h`.

**8.25.2.4** `typedef struct command_s command_t`

### 8.25.3 Function Documentation

**8.25.3.1** `int cmd_prompt_Destroy (cmd_prompt_p cmd_prompt)`

Definition at line 63 of file `cmd_prompt.c`.

References `MC_SUCCESS`.

Referenced by `mc_platform_Destroy()`.

**8.25.3.2** `cmd_prompt_p cmd_prompt_Initialize (struct mc_platform_s * mc_platform)`

**8.25.3.3** `void cmd_prompt_Start (struct mc_platform_s * mc_platform)`

**8.25.3.4** `void* cmd_prompt_Thread (void * arg)`

Definition at line 101 of file `cmd_prompt.c`.

References `command_s::args`, `dealloc_command()`, `exec_command()`, `command_s::index`, `mc_platform`, `command_s::num_args`, `process_command()`, and `split_string()`.

Referenced by `cmd_prompt_Start()`.

**8.25.3.5** `int dealloc_command (command_t * cmd)`

Definition at line 228 of file `cmd_prompt.c`.

Referenced by `cmd_prompt_Thread()`.

**8.25.3.6** `int exec_command (command_t cmd, struct mc_platform_s * global)`

**8.25.3.7** `int process_command (command_t * cmd)`

Definition at line 194 of file `cmd_prompt.c`.

References `command_s::args`, `command_cmds`, `command_s::index`, and `command_s::num_args`.

Referenced by `cmd_prompt_Thread()`.

**8.25.3.8** `int split_string (char *** args, const char * buf)`

Definition at line 146 of file `cmd_prompt.c`.

References `int`.

Referenced by `cmd_prompt_Thread()`.



## 8.26 /home/dko/projects/mobilec/trunk/src/include/commands.h File Reference

```
#include "macros.h"
#include "mc_platform.h"
#include "commands.x.h"
```

### Defines

- #define [COMMAND](#)(name, cmd, desc) [int](#) handler\_##name (void \*arg, [mc\\_platform\\_p](#) global);
- #define [COMMAND](#)(name, cmd, description) [COMMAND\\_##name](#) ,
- #define [COMMAND](#)(name, cmd, description) &handler\_##name ,
- #define [COMMAND](#)(name, cmd, description) description ,
- #define [COMMAND](#)(name, cmd, description) cmd ,

### Typedefs

- typedef [int](#)(\* [cmd\\_handler\\_t](#) )(void \*, [mc\\_platform\\_p](#) global)

### Enumerations

- enum [command\\_indices\\_e](#) { [COMMAND](#) }

### Variables

- [cmd\\_handler\\_t](#) [cmd\\_handlers](#) []
- char \* [command\\_descriptions](#) []
- char \* [command\\_cmds](#) []

#### 8.26.1 Define Documentation

##### 8.26.1.1 #define [COMMAND](#)(name, cmd, description) cmd ,

Definition at line 41 of file [commands.h](#).

##### 8.26.1.2 #define [COMMAND](#)(name, cmd, description) description ,

Definition at line 41 of file [commands.h](#).

##### 8.26.1.3 #define [COMMAND](#)(name, cmd, description) &handler\_##name ,

Definition at line 41 of file [commands.h](#).

##### 8.26.1.4 #define [COMMAND](#)(name, cmd, description) [COMMAND\\_##name](#) ,

Definition at line 41 of file [commands.h](#).

**8.26.1.5** `#define COMMAND(name, cmd, desc) int handler_##name (void *arg, mc_platform_p global);`

Definition at line 41 of file commands.h.

## 8.26.2 Typedef Documentation

**8.26.2.1** `typedef int(* cmd_handler_t)(void *, mc_platform_p global)`

Definition at line 45 of file commands.h.

## 8.26.3 Enumeration Type Documentation

**8.26.3.1** `enum command_indices_e`

Enumerator:

*COMMAND*

Definition at line 47 of file commands.h.

## 8.26.4 Variable Documentation

**8.26.4.1** `cmd_handler_t cmd_handlers[ ]`

Definition at line 54 of file commands.h.

**8.26.4.2** `char* command_cmds[ ]`

Definition at line 186 of file commands.h.

Referenced by `process_command()`.

**8.26.4.3** `char* command_descriptions[ ]`

Definition at line 120 of file commands.h.

## 8.27 /home/dko/projects/mobilec/trunk/src/include/commands.x.h File Reference

### Functions

- **COMMAND** (QUIT,"quit","This command ends the MobileC application.") **COMMAND**(HELP
- This command displays [help](#) for any command Usage This command sends a client to a host Usage This command prints the list entries in the connectList **COMMAND** (PRINTLIST\_MESSAGE,"printlist\_message","This command prints all the list entries on the linked list:\n \ message\_queue") **COMMAND**(PRINTLIST\_AGENTS

### Variables

- [help](#)
- This command displays [help](#) for any command Usage [is](#)
- This command displays [help](#) for any command Usage [send](#)
- This command displays [help](#) for any command Usage This command sends a client to a host Usage [print\\_connectlist](#)
- This command displays [help](#) for any command Usage This command sends a client to a host Usage This command prints the list entries in the connectList [printlist\\_agents](#)

### 8.27.1 Function Documentation

**8.27.1.1 This command displays help for any command Usage This command sends a client to a host Usage This command prints the list entries in the connectList COMMAND (PRINTLIST\_MESSAGE, "printlist\_message", "This command prints all the list entries on the linked list:\n \ message\_queue")**

Type constraints

**8.27.1.2 COMMAND (QUIT, "quit", "This command ends the MobileC application.")**

### 8.27.2 Variable Documentation

#### 8.27.2.1 help

Definition at line 40 of file commands.x.h.

**8.27.2.2 This command displays help for any command Usage This command sends a client to a host Usage is**

Definition at line 40 of file commands.x.h.

**8.27.2.3 This command displays help for any command Usage This command sends a client to a host Usage print\_connectlist**

Definition at line 40 of file commands.x.h.

**8.27.2.4** This command displays help for any command Usage This command sends a client to a host Usage This command prints the list entries in the connectList printlist\_agents

Definition at line 55 of file commands.x.h.

**8.27.2.5** This command displays help for any command Usage send

Definition at line 40 of file commands.x.h.

Referenced by message\_Send(), and mtp\_http\_InitializeFromConnection().

## 8.28 /home/dko/projects/mobilec/trunk/src/include/connection.h File Reference

```
#include <sys/types.h>
#include <netinet/in.h>
```

### Data Structures

- struct [connection\\_s](#)

### Typedefs

- typedef struct [connection\\_s](#) [connection\\_t](#)
- typedef [connection\\_t](#) \* [connection\\_p](#)

### Functions

- [connection\\_p](#) [connection\\_New](#) (void)
- int [connection\\_Destroy](#) ([connection\\_p](#) connection)
- [connection\\_p](#) [connection\\_Copy](#) ([connection\\_p](#) connection)
- void [connection\\_Close](#) ([connection\\_p](#) connection)
- void [connection\\_SetHosts](#) ([connection\\_p](#) connection, char \*remote\_host, char \*local\_host)
- int [connection\\_CreateMessage](#) ([connection\\_p](#) connection)
- int [connection\\_SendtoConnection](#) ([connection\\_p](#) connection, char \*send\_buffer)
- int [connection\\_RecvFromConnection](#) ([connection\\_p](#) connection, struct [mc\\_platform\\_s](#) \*global)

### 8.28.1 Typedef Documentation

#### 8.28.1.1 typedef [connection\\_t](#)\* [connection\\_p](#)

Definition at line 59 of file [connection.h](#).

#### 8.28.1.2 typedef struct [connection\\_s](#) [connection\\_t](#)

### 8.28.2 Function Documentation

#### 8.28.2.1 void [connection\\_Close](#) ([connection\\_p](#) *connection*)

#### 8.28.2.2 [connection\\_p](#) [connection\\_Copy](#) ([connection\\_p](#) *connection*)

Definition at line 72 of file [connection.c](#).

References [connection\\_s::addr](#), [connection\\_s::clientfd](#), [connection\\_s::connect\\_id](#), [connection\\_New\(\)](#), [connection\\_s::remote\\_hostname](#), and [connection\\_s::serverfd](#).

**8.28.2.3 int connection\_CreateMessage (connection\_p *connection*)****8.28.2.4 int connection\_Destroy (connection\_p *connection*)**

Definition at line 42 of file connection.c.

References connection\_s::clientfd, MC\_SUCCESS, and connection\_s::remote\_hostname.

Referenced by acc\_Thread().

**8.28.2.5 connection\_p connection\_New (void)**

Definition at line 61 of file connection.c.

Referenced by connection\_Copy().

**8.28.2.6 int connection\_RecvFromConnection (connection\_p *connection*, struct mc\_platform\_s \* *global*)****8.28.2.7 int connection\_SendtoConnection (connection\_p *connection*, char \* *send\_buffer*)****8.28.2.8 void connection\_SetHosts (connection\_p *connection*, char \* *remote\_host*, char \* *local\_host*)**

## 8.29 /home/dko/projects/mobilec/trunk/src/include/data\_structures.h File Reference

```
#include "ap_queue_template.h"
#include "../mc_list/list.h"
#include "connection.h"
#include "fipa_acl.h"
#include "agent_mailbox.h"
#include "message.h"
#include "interpreter_variable_data.h"
```

### Functions

- [AP\\_QUEUE\\_DECL\\_TEMPLATE](#) (connection\_queue, connection) [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#)(connection\_queue
- [connection\\_t](#) [int](#) [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#) (connection\_queue, [Remove](#), [int](#), [int](#)) [int](#) [connection\\_queue\\_Print](#)(connection\_queue\_p clist)
- [AP\\_QUEUE\\_DECL\\_TEMPLATE](#) (message\_queue, message) [int](#) [message\\_queue\\_Print](#)(message\_queue\_p queue)
- [AP\\_QUEUE\\_DECL\\_TEMPLATE](#) (agent\_queue, agent) [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#)(agent\_queue
- [struct agent\\_s](#) [int](#) [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#) (agent\_queue, [SearchName](#), [struct agent\\_s](#) \*, [char](#) \*) [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#)(agent\_queue
- [struct agent\\_s](#) [int](#) [int](#) [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#) (agent\_queue, [RemoveName](#), [int](#), [char](#) \*) [int](#) [agent\\_queue\\_Print](#)(agent\_queue\_p queue)
- [AP\\_QUEUE\\_DECL\\_TEMPLATE](#) (agent\_variable\_list, interpreter\_variable\_data) [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#)(agent\_variable\_list
- [interpreter\\_variable\\_data\\_t](#) [char](#) \* [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#) (agent\_variable\_list, [Remove](#), [int](#), [char](#) \*) [AP\\_QUEUE\\_DECL\\_TEMPLATE](#)(mail\_queue
- [interpreter\\_variable\\_data\\_t](#) [char](#) \*fipa\_acl\_message [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#) (mail\_queue, [SearchReceivers](#), [fipa\\_acl\\_message\\_p](#), [char](#) \*) [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#)(mail\_queue
- [interpreter\\_variable\\_data\\_t](#) [char](#) \*fipa\_acl\_message [char](#) \* [AP\\_QUEUE\\_DECL\\_TEMPLATE](#) (mailbox\_queue, agent\_mailbox) [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#)(mailbox\_queue
- [interpreter\\_variable\\_data\\_t](#) [char](#) \*fipa\_acl\_message [char](#) [char](#) \* [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#) (mailbox\_queue, [SearchID](#), [agent\\_mailbox\\_p](#), [int](#)) [AP\\_QUEUE\\_GENERIC\\_DECL\\_TEMPLATE](#)(mailbox\_queue

### Variables

- [Search](#)
- [struct agent\\_s](#) [int](#) [Remove](#)
- [struct agent\\_s](#) [int](#) [int](#)
- [interpreter\\_variable\\_data\\_t](#) [char](#) \*fipa\_acl\_message [RemoveName](#)
- [interpreter\\_variable\\_data\\_t](#) [char](#) \*fipa\_acl\_message [char](#) \* [SearchName](#)
- [interpreter\\_variable\\_data\\_t](#) [char](#) \*fipa\_acl\_message [char](#) [agent\\_mailbox\\_p](#)
- [interpreter\\_variable\\_data\\_t](#) [char](#) \*fipa\_acl\_message [char](#) [char](#) \* [RemoveID](#)

## 8.29.1 Function Documentation

**8.29.1.1** `interpreter_variable_data_t char* fipa_acl_message char* AP_QUEUE_DECL_TEMPLATE (mailbox_queue, agent_mailbox)`

Type constraints

**8.29.1.2** `AP_QUEUE_DECL_TEMPLATE (agent_variable_list, interpreter_variable_data)`

**8.29.1.3** `AP_QUEUE_DECL_TEMPLATE (agent_queue, agent)`

**8.29.1.4** `AP_QUEUE_DECL_TEMPLATE (message_queue, message)`

**8.29.1.5** `AP_QUEUE_DECL_TEMPLATE (connection_queue, connection)`

**8.29.1.6** `interpreter_variable_data_t char* fipa_acl_message char char* AP_QUEUE_GENERIC_DECL_TEMPLATE (mailbox_queue, SearchID, agent_mailbox_p, int)`

Type constraints

**8.29.1.7** `interpreter_variable_data_t char* fipa_acl_message AP_QUEUE_GENERIC_DECL_TEMPLATE (mail_queue, SearchReceivers, fipa_acl_message_p, char *)`

**8.29.1.8** `interpreter_variable_data_t char* AP_QUEUE_GENERIC_DECL_TEMPLATE (agent_variable_list, Remove, int, char *)`

Type constraints

**8.29.1.9** `struct agent_s int int AP_QUEUE_GENERIC_DECL_TEMPLATE (agent_queue, RemoveName, int, char *)`

Type constraints

**8.29.1.10** `struct agent_s int AP_QUEUE_GENERIC_DECL_TEMPLATE (agent_queue, SearchName, struct agent_s *, char *) [read]`

Type constraints



### 8.29.1.11 `connection_t` `int` `AP_QUEUE_GENERIC_DECL_TEMPLATE` (`connection_queue`, `Remove`, `int`, `int`)

Type constraints

## 8.29.2 Variable Documentation

### 8.29.2.1 `interpreter_variable_data_t` `char*` `fipa_acl_message` `char` `agent_mailbox_p`

Definition at line 162 of file `data_structures.h`.

### 8.29.2.2 `interpreter_variable_data_t` `char` `*fipa_acl_message` `char` `char` `int`

Definition at line 101 of file `data_structures.h`.

Referenced by `LibMC::MCAgency::ChInitializeOptions()`, `default_callback()`, `main()`, `mpi_lsb()`, `mpi_read_string()`, `net_accept()`, `rsa_pkcs1_decrypt()`, `rsa_pkcs1_verify()`, `sha4_finish()`, `sha4_update()`, `split_string()`, `ssl_decrypt_buf()`, `ssl_get_session()`, `ssl_parse_certificate_verify()`, `ssl_parse_client_hello()`, `ssl_parse_server_hello()`, and `ssl_set_session()`.

### 8.29.2.3 `struct agent_s` `int` `Remove`

Definition at line 101 of file `data_structures.h`.

### 8.29.2.4 `interpreter_variable_data_t` `char*` `fipa_acl_message` `char` `char*` `RemoveID`

Definition at line 176 of file `data_structures.h`.

### 8.29.2.5 `interpreter_variable_data_t` `char*` `fipa_acl_message` `RemoveName`

Definition at line 149 of file `data_structures.h`.

### 8.29.2.6 `Search`

Definition at line 53 of file `data_structures.h`.

### 8.29.2.7 `interpreter_variable_data_t` `char*` `fipa_acl_message` `char*` `SearchName`

Definition at line 162 of file `data_structures.h`.

## 8.30 /home/dko/projects/mobilec/trunk/src/include/df.h File Reference

```
#include "../mc_list/list.h"
#include "../include/macros.h"
#include "df_request.x.h"
```

### Defines

- #define [REQUEST](#)(name, string, description) [REQUEST\\_##name](#),
- #define [REQUEST](#)(name, string, description) [int](#) request\_handler\_##name (struct [mc\\_platform\\_s](#) \*global, void\* data);

### Enumerations

- enum [df\\_request\\_list\\_index\\_e](#) { [REQUEST](#) }
- enum [service\\_types\\_e](#) {  
[ZERO](#), [MISC](#), [INSERT](#), [SOME](#),  
[TYPES](#), [HERE](#) }

### Functions

- [STRUCT](#) (df\_request\_list\_node, [MUTEX\\_T](#) \*[lock](#); [COND\\_T](#) \*[cond](#); const char \*command; void \*data; [int](#) data\_size;) [STRUCT](#)(df\_request\_list
- [STRUCT](#) (df\_search\_results, char \*\*agent\_names; char \*\*service\_names; [int](#) \*agent\_ids; [int](#) num\_results;) [STRUCT](#)(df\_request\_search
- [STRUCT](#) (df\_node, [MUTEX\\_T](#) \*[lock](#); [int](#) agent\_id; char \*agent\_name; [int](#) num\_services; char \*\*service\_names; enum [service\\_types\\_e](#) service\_types;) [STRUCT](#)(df
- [STRUCT](#) (df\_deregister, [int](#) agent\_id; char \*service\_name;) [int](#) df\_Add(struct df\_s \*df
- [int](#) df\_AddRequest (struct df\_s \*df, struct df\_request\_list\_node\_s \*[node](#))
- [int](#) df\_Destroy (df\_p df)
- df\_p df\_Initialize (struct [mc\\_platform\\_s](#) \*[mc\\_platform](#))
- [int](#) df\_ProcessRequest (struct [mc\\_platform\\_s](#) \*[global](#))
- [int](#) df\_SearchForService (df\_p df, const char \*searchstring, char \*\*\*agent\_names, char \*\*\*service\_names, [int](#) \*\*agent\_ids, [int](#) \*num\_entries)
- void df\_Start (struct [mc\\_platform\\_s](#) \*[mc\\_platform](#))
- [int](#) df\_node\_Destroy (df\_node\_p df\_node)
- [int](#) df\_request\_list\_node\_Destroy (df\_request\_list\_node\_p [node](#))
- df\_request\_list\_node\_p df\_request\_list\_node\_New (void)
- [int](#) df\_request\_list\_Destroy (df\_request\_list\_p df\_request\_list)
- df\_request\_list\_p df\_request\_list\_New (void)
- df\_request\_list\_node\_p df\_request\_list\_Pop (df\_request\_list\_p requests)
- df\_request\_search\_p df\_request\_search\_New (void)
- [int](#) df\_request\_search\_Destroy (df\_request\_search\_p [node](#))
- void \* df\_Thread (void \*arg)

## Variables

- MUTEX\_T \* [lock](#)
- COND\_T \* [cond](#)
- [int](#) [size](#)
- [list\\_p](#) [request\\_list](#)
- [char](#) \* [search\\_string](#)
- [df\\_search\\_results\\_p](#) [search\\_results](#)
- [struct mc\\_platform\\_s](#) \* [mc\\_platform](#)
- [list\\_p](#) [service\\_list](#)
- [int](#) [num\\_entries](#)
- [int](#) [waiting](#)
- MUTEX\_T \* [waiting\\_lock](#)
- COND\_T \* [waiting\\_cond](#)
- [THREAD\\_T](#) [thread](#)
- [struct df\\_node\\_s](#) \* [node](#)

### 8.30.1 Define Documentation

**8.30.1.1** `#define REQUEST(name, string, description) int request_handler_##name (struct mc_platform_s *global, void* data);`

Definition at line 192 of file df.h.

**8.30.1.2** `#define REQUEST(name, string, description) REQUEST_##name,`

Definition at line 192 of file df.h.

### 8.30.2 Enumeration Type Documentation

**8.30.2.1** `enum df_request_list_index_e`

Enumerator:

*REQUEST*

Definition at line 46 of file df.h.

**8.30.2.2** `enum service_types_e`

Enumerator:

*ZERO*

*MISC*

*INSERT*

*SOME*

*TYPES*

*HERE*

Definition at line 53 of file df.h.

### 8.30.3 Function Documentation

#### 8.30.3.1 `int df_AddRequest (struct df_s * df, struct df_request_list_node_s * node)`

Definition at line 63 of file df.c.

References ListAdd(), and SIGNAL.

Referenced by MC\_DeregisterService(), MC\_RegisterService(), and MC\_SearchForService().

#### 8.30.3.2 `int df_Destroy (df_p df)`

Definition at line 80 of file df.c.

References COND\_DESTROY, df\_node\_Destroy(), df\_request\_list\_Destroy(), ListPop(), ListTerminate(), MC\_SUCCESS, MUTEX\_DESTROY, and MUTEX\_LOCK.

Referenced by mc\_platform\_Destroy().

#### 8.30.3.3 `df_p df_Initialize (struct mc_platform_s * mc_platform)`

#### 8.30.3.4 `int df_node_Destroy (df_node_p df_node)`

Definition at line 408 of file df.c.

References MC\_SUCCESS, and MUTEX\_LOCK.

Referenced by df\_Destroy().

#### 8.30.3.5 `int df_ProcessRequest (struct mc_platform_s * global)`

Definition at line 129 of file df.c.

References mc\_platform\_s::df, df\_request\_list\_Pop(), MC\_ERR\_EMPTY, and MC\_ERR\_INVALID.

Referenced by df\_Thread().

#### 8.30.3.6 `int df_request_list_Destroy (df_request_list_p df_request_list)`

Definition at line 317 of file df.c.

References df\_request\_list\_node\_Destroy(), ListPop(), ListTerminate(), MC\_SUCCESS, and node.

Referenced by df\_Destroy().

#### 8.30.3.7 `df_request_list_p df_request_list_New (void)`

Definition at line 339 of file df.c.

References CHECK\_NULL, COND\_INIT, COND\_T, ListInitialize(), MUTEX\_INIT, and MUTEX\_T.

Referenced by df\_Initialize().

#### 8.30.3.8 `int df_request_list_node_Destroy (df_request_list_node_p node)`

Definition at line 288 of file df.c.

References COND\_DESTROY, MC\_SUCCESS, and MUTEX\_DESTROY.

Referenced by df\_request\_list\_Destroy(), and MC\_SearchForService().

#### 8.30.3.9 df\_request\_list\_node\_p df\_request\_list\_node\_New (void)

Definition at line 299 of file df.c.

References CHECK\_NULL, COND\_INIT, COND\_T, MUTEX\_INIT, MUTEX\_T, and node.

Referenced by MC\_DeregisterService(), MC\_RegisterService(), and MC\_SearchForService().

#### 8.30.3.10 df\_request\_list\_node\_p df\_request\_list\_Pop (df\_request\_list\_p requests)

Definition at line 364 of file df.c.

References ListPop(), MUTEX\_LOCK, MUTEX\_UNLOCK, and node.

Referenced by df\_ProcessRequest().

#### 8.30.3.11 int df\_request\_search\_Destroy (df\_request\_search\_p node)

Definition at line 395 of file df.c.

References COND\_DESTROY, MC\_SUCCESS, and MUTEX\_DESTROY.

Referenced by MC\_SearchForService().

#### 8.30.3.12 df\_request\_search\_p df\_request\_search\_New (void)

Definition at line 380 of file df.c.

References CHECK\_NULL, COND\_INIT, COND\_T, MUTEX\_INIT, MUTEX\_T, and search.

Referenced by MC\_SearchForService().

#### 8.30.3.13 int df\_SearchForService (df\_p df, const char \* searchstring, char \*\*\* agent\_names, char \*\*\* service\_names, int \*\* agent\_ids, int \* num\_entries)

Definition at line 173 of file df.c.

References MC\_ERR\_NOT\_FOUND, MC\_SUCCESS, MUTEX\_LOCK, MUTEX\_UNLOCK, listNode\_s::next, and listNode\_s::node\_data.

#### 8.30.3.14 void df\_Start (struct mc\_platform\_s \* mc\_platform)

#### 8.30.3.15 void\* df\_Thread (void \* arg)

Definition at line 422 of file df.c.

References COND\_BROADCAST, COND\_WAIT, mc\_platform\_s::df, df\_ProcessRequest(), MC\_SUCCESS, MUTEX\_LOCK, MUTEX\_UNLOCK, mc\_platform\_s::quit, and mc\_platform\_s::quit\_lock.

Referenced by df\_Start().

**8.30.3.16** **STRUCT** (df\_deregister, int agent\_id;char \*service\_name;)

**8.30.3.17** **STRUCT** (df\_node, MUTEX\_T \*lock;int agent\_id;char \*agent\_name;int num\_services;char \*\*service\_names;enum service\_types\_e service\_types;)

**8.30.3.18** **STRUCT** (df\_search\_results, char \*\*agent\_names;char \*\*service\_names;int \*agent\_ids;int num\_results;)

**8.30.3.19** **STRUCT** (df\_request\_list\_node, MUTEX\_T \*lock;COND\_T \*cond;const char \*command;void \*data;int data\_size;)

## 8.30.4 Variable Documentation

**8.30.4.1** **COND\_T \* cond**

Definition at line 74 of file df.h.

**8.30.4.2** **MUTEX\_T \* lock**

Definition at line 73 of file df.h.

**8.30.4.3** **struct mc\_platform\_s\* mc\_platform**

Definition at line 107 of file df.h.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), agent\_RunChScriptThread(), ams\_Thread(), cmd\_prompt\_Thread(), listen\_Thread(), mc\_platform\_Initialize(), and MC\_WaitAgent().

**8.30.4.4** **struct df\_node\_s\* node**

Definition at line 129 of file df.h.

Referenced by add\_variable(), agent\_xml\_compose(), agent\_xml\_compose\_\_agent\_code(), agent\_xml\_compose\_\_agent\_data(), agent\_xml\_compose\_\_create\_row\_nodes(), agent\_xml\_compose\_\_data(), agent\_xml\_compose\_\_gaf\_message(), agent\_xml\_compose\_\_home(), agent\_xml\_compose\_\_message(), agent\_xml\_compose\_\_mobile\_agent(), agent\_xml\_compose\_\_name(), agent\_xml\_compose\_\_owner(), agent\_xml\_compose\_\_row(), agent\_xml\_compose\_\_task(), agent\_xml\_compose\_\_tasks(), AP\_QUEUE\_SEARCH\_TEMPLATE(), AP\_QUEUE\_STD\_DEFN\_TEMPLATE(), barrier\_node\_Initialize(), barrier\_queue\_Destroy(), barrier\_queue\_Pop(), df\_request\_list\_Destroy(), df\_request\_list\_node\_New(), df\_request\_list\_Pop(), fipa\_envelope\_Compose(), fipa\_envelope\_Compose\_\_acl\_representation(), fipa\_envelope\_Compose\_\_date(), fipa\_envelope\_Compose\_\_envelope(), fipa\_envelope\_Compose\_\_intended\_receiver(), fipa\_envelope\_Compose\_\_params(), fipa\_envelope\_Compose\_\_payload\_encoding(), fipa\_envelope\_Compose\_\_to(), fipa\_envelope\_HandleReceived(), main(), MC\_Barrier(), MC\_BarrierInit(), MC\_SyncInit(), mxml\_new(), mxmlNewCustom(), mxmlNewElement(), mxmlNewInteger(), mxmlNewOpaque(), mxmlNewReal(), mxmlNewText(), mxmlNewTextf(), scan\_file(), syncListNew(), write\_element(), xml\_get\_deep\_child(), and xml\_new\_cdata().

**8.30.4.5** **int num\_entries**

Definition at line 113 of file df.h.

**8.30.4.6 df\_request\_list\_p request\_list**

Definition at line 76 of file df.h.

**8.30.4.7 df\_search\_results\_p search\_results**

Definition at line 91 of file df.h.

**8.30.4.8 char\* search\_string**

Definition at line 90 of file df.h.

**8.30.4.9 list\_p service\_list**

Definition at line 110 of file df.h.

**8.30.4.10 int size**

Definition at line 75 of file df.h.

Referenced by agent\_AddPersistentVariable(), agent\_xml\_compose\_\_create\_row\_nodes(), interpreter\_variable\_data\_Initialize(), interpreter\_variable\_data\_InitializeFromAgent(), MC\_GetAgentReturnData(), MC\_WaitAgent(), and mxml\_vsnprintf().

**8.30.4.11 THREAD\_T thread**

Definition at line 117 of file df.h.

**8.30.4.12 int waiting**

Definition at line 114 of file df.h.

**8.30.4.13 COND\_T\* waiting\_cond**

Definition at line 116 of file df.h.

**8.30.4.14 MUTEX\_T\* waiting\_lock**

Definition at line 115 of file df.h.

## 8.31 /home/dko/projects/mobilec/trunk/src/include/df\_request.x.h File Reference

### Functions

- [REQUEST](#) (SUBSCRIBE, "subscribe", "Subscribe to a directory facilitator") [REQUEST](#) (REGISTER
- Register services with the directory facilitator [REQUEST](#) (DEREGISTER, "deregister", "Deregisters mobile agent services from a directory facilitator.") [REQUEST](#) (SEARCH

### Variables

- [register](#)
- Register services with the directory facilitator [search](#)

### 8.31.1 Function Documentation

- 8.31.1.1 Register services with the directory facilitator [REQUEST](#) (DEREGISTER, "deregister", "Deregisters mobile agent services from a directory facilitator.")**

Type constraints

- 8.31.1.2 [REQUEST](#) (SUBSCRIBE, "subscribe", "Subscribe to a directory facilitator")**

### 8.31.2 Variable Documentation

#### 8.31.2.1 [register](#)

Definition at line 43 of file df\_request.x.h.

#### 8.31.2.2 Register services with the directory facilitator [search](#)

Definition at line 54 of file df\_request.x.h.

Referenced by [df\\_request\\_search\\_New\(\)](#), and [MC\\_SearchForService\(\)](#).



## 8.32 /home/dko/projects/mobilec/trunk/src/include/dynstring.h File Reference

### Data Structures

- struct [dynstring\\_s](#)

### Defines

- #define [COMPOSE\\_BLOCKSIZE](#) 1024

### Typedefs

- typedef struct [dynstring\\_s](#) [dynstring\\_t](#)
- typedef [dynstring\\_t](#) \* [dynstring\\_p](#)

### Functions

- [dynstring\\_t](#) \* [dynstring\\_New](#) (void)
- [int](#) [dynstring\\_Append](#) ([dynstring\\_t](#) \*msg, char \*str)
- [int](#) [dynstring\\_Destroy](#) ([dynstring\\_t](#) \*dynstring)

#### 8.32.1 Define Documentation

##### 8.32.1.1 #define COMPOSE\_BLOCKSIZE 1024

Definition at line 8 of file [dynstring.h](#).

Referenced by [dynstring\\_Append\(\)](#), and [dynstring\\_New\(\)](#).

#### 8.32.2 Typedef Documentation

##### 8.32.2.1 typedef [dynstring\\_t](#)\* [dynstring\\_p](#)

Definition at line 14 of file [dynstring.h](#).

##### 8.32.2.2 typedef struct [dynstring\\_s](#) [dynstring\\_t](#)

#### 8.32.3 Function Documentation

##### 8.32.3.1 [int](#) [dynstring\\_Append](#) ([dynstring\\_t](#) \* *msg*, char \* *str*)

Definition at line 24 of file [dynstring.c](#).

References [COMPOSE\\_BLOCKSIZE](#), [dynstring\\_s::len](#), [dynstring\\_s::message](#), and [dynstring\\_s::size](#).

Referenced by [fipa\\_acl\\_Compose\(\)](#), [fipa\\_agent\\_identifier\\_Compose\(\)](#), [fipa\\_agent\\_identifier\\_set\\_-Compose\(\)](#), [fipa\\_DateTime\\_Compose\(\)](#), [fipa\\_envelope\\_Compose\\_\\_from\(\)](#), [fipa\\_number\\_Compose\(\)](#), [fipa\\_performative\\_Compose\(\)](#), [fipa\\_string\\_Compose\(\)](#), [fipa\\_url\\_Compose\(\)](#), [fipa\\_url\\_sequence\\_-Compose\(\)](#), [fipa\\_word\\_Compose\(\)](#), and [mtp\\_http\\_CreateMessage\(\)](#).

**8.32.3.2 int dynstring\_Destroy (dynstring\_t \* *dynstring*)**

Definition at line 49 of file dynstring.c.

References dynstring\_s::message.

Referenced by fipa\_envelope\_Compose\_\_from(), MC\_AclSend(), and mtp\_http\_CreateMessage().

**8.32.3.3 dynstring\_t\* dynstring\_New (void)**

Definition at line 8 of file dynstring.c.

References COMPOSE\_BLOCKSIZE, dynstring\_s::len, dynstring\_s::message, and dynstring\_s::size.

Referenced by fipa\_acl\_Compose(), fipa\_envelope\_Compose\_\_from(), and mtp\_http\_CreateMessage().

## 8.33 /home/dko/projects/mobilec/trunk/src/include/fipa\_acl.h File Reference

```
#include "dynstring.h"
```

### Data Structures

- struct [fipa\\_acl\\_message\\_s](#)
- struct [fipa\\_message\\_string\\_s](#)
- struct [fipa\\_url\\_sequence\\_s](#)
- struct [fipa\\_agent\\_identifier\\_set\\_s](#)
- struct [fipa\\_agent\\_identifier\\_s](#)
- struct [fipa\\_expression\\_s](#)
- union [fipa\\_expression\\_s::content\\_u](#)
- struct [fipa\\_word\\_s](#)
- struct [fipa\\_string\\_s](#)
- struct [fipa\\_DateTime\\_s](#)
- struct [fipa\\_url\\_s](#)
- struct [fipa\\_number\\_s](#)

### Typedefs

- typedef enum [fipa\\_expression\\_type\\_e](#) [fipa\\_expression\\_type\\_t](#)
- typedef struct [fipa\\_acl\\_message\\_s](#) [fipa\\_acl\\_message\\_t](#)
- typedef [fipa\\_acl\\_message\\_t](#) \* [fipa\\_acl\\_message\\_p](#)
- typedef struct [fipa\\_message\\_string\\_s](#) [fipa\\_message\\_string\\_t](#)
- typedef [fipa\\_message\\_string\\_t](#) \* [fipa\\_message\\_string\\_p](#)
- typedef struct [fipa\\_url\\_sequence\\_s](#) [fipa\\_url\\_sequence\\_t](#)
- typedef [fipa\\_url\\_sequence\\_t](#) \* [fipa\\_url\\_sequence\\_p](#)
- typedef struct [fipa\\_agent\\_identifier\\_set\\_s](#) [fipa\\_agent\\_identifier\\_set\\_t](#)
- typedef [fipa\\_agent\\_identifier\\_set\\_t](#) \* [fipa\\_agent\\_identifier\\_set\\_p](#)
- typedef struct [fipa\\_agent\\_identifier\\_s](#) [fipa\\_agent\\_identifier\\_t](#)
- typedef [fipa\\_agent\\_identifier\\_t](#) \* [fipa\\_agent\\_identifier\\_p](#)
- typedef struct [fipa\\_expression\\_s](#) [fipa\\_expression\\_t](#)
- typedef [fipa\\_expression\\_t](#) \* [fipa\\_expression\\_p](#)
- typedef struct [fipa\\_word\\_s](#) [fipa\\_word\\_t](#)
- typedef [fipa\\_word\\_t](#) \* [fipa\\_word\\_p](#)
- typedef struct [fipa\\_string\\_s](#) [fipa\\_string\\_t](#)
- typedef [fipa\\_string\\_t](#) \* [fipa\\_string\\_p](#)
- typedef struct [fipa\\_DateTime\\_s](#) [fipa\\_DateTime\\_t](#)
- typedef [fipa\\_DateTime\\_t](#) \* [fipa\\_DateTime\\_p](#)
- typedef struct [fipa\\_url\\_s](#) [fipa\\_url\\_t](#)
- typedef [fipa\\_url\\_t](#) \* [fipa\\_url\\_p](#)
- typedef struct [fipa\\_number\\_s](#) [fipa\\_number\\_t](#)
- typedef [fipa\\_number\\_t](#) \* [fipa\\_number\\_p](#)

## Enumerations

- enum `fipa_performative_e` {  
`FIPA_ERROR` = -1, `FIPA_ZERO`, `FIPA_ACCEPT_PROPOSAL`, `FIPA_AGREE`,  
`FIPA_CANCEL`, `FIPA_CALL_FOR_PROPOSAL`, `FIPA_CONFIRM`, `FIPA_DISCONFIRM`,  
`FIPA_FAILURE`, `FIPA_INFORM`, `FIPA_INFORM_IF`, `FIPA_INFORM_REF`,  
`FIPA_NOT_UNDERSTOOD`, `FIPA_PROPOGATE`, `FIPA_PROPOSE`, `FIPA_PROXY`,  
`FIPA_QUERY_IF`, `FIPA_QUERY_REF`, `FIPA_REFUSE`, `FIPA_REJECT_PROPOSAL`,  
`FIPA_REQUEST`, `FIPA_REQUEST_WHEN`, `FIPA_REQUEST_WHENEVER`, `FIPA_-`  
`SUBSCRIBE` }
- enum `fipa_expression_type_e` {  
`FIPA_EXPR_ZERO`, `FIPA_EXPR_WORD`, `FIPA_EXPR_STRING`, `FIPA_EXPR_NUMBER`,  
`FIPA_EXPR_DATETIME`, `FIPA_EXPR_EXPRESSION` }

## Functions

- `int fipa_performative_Compose` (`dynstring_t *msg`, enum `fipa_performative_e` `performative`)
- `fipa_acl_message_t * fipa_acl_message_New` (`void`)
- `int fipa_acl_message_Destroy` (`fipa_acl_message_t *message`)
- `fipa_acl_message_t * fipa_acl_message_Copy` (`fipa_acl_message_t *src`)
- `int fipa_acl_Compose` (`dynstring_t **msg`, `fipa_acl_message_t *acl`)
- `fipa_message_string_t * fipa_message_string_New` (`void`)
- `int fipa_message_string_Destroy` (`fipa_message_string_t *message`)
- `fipa_message_string_t * fipa_message_string_Copy` (`fipa_message_string_t *src`)
- `fipa_url_sequence_t * fipa_url_sequence_New` (`void`)
- `int fipa_url_sequence_Destroy` (`fipa_url_sequence_t *sequence`)
- `fipa_url_sequence_t * fipa_url_sequence_Copy` (`fipa_url_sequence_t *src`)
- `int fipa_url_sequence_Compose` (`dynstring_t *msg`, `fipa_url_sequence_t *urls`)
- `fipa_agent_identifier_set_t * fipa_agent_identifier_set_New` (`void`)
- `int fipa_agent_identifier_set_Destroy` (`fipa_agent_identifier_set_t *idset`)
- `fipa_agent_identifier_set_t * fipa_agent_identifier_set_Copy` (`fipa_agent_identifier_set_t *src`)
- `int fipa_agent_identifier_set_Compose` (`dynstring_t *msg`, `fipa_agent_identifier_set_t *ids`)
- `fipa_agent_identifier_t * fipa_agent_identifier_New` (`void`)
- `int fipa_agent_identifier_Destroy` (`fipa_agent_identifier_t *id`)
- `fipa_agent_identifier_t * fipa_agent_identifier_Copy` (`fipa_agent_identifier_t *src`)
- `int fipa_agent_identifier_Compose` (`dynstring_t *msg`, `fipa_agent_identifier_t *id`)
- `fipa_expression_t * fipa_expression_New` (`void`)
- `int fipa_expression_Destroy` (`fipa_expression_t *expr`)
- `fipa_expression_t * fipa_expression_Copy` (`fipa_expression_t *src`)
- `int fipa_expression_Compose` (`dynstring_t *msg`, `fipa_expression_t *expr`)
- `fipa_word_t * fipa_word_New` (`void`)
- `int fipa_word_Destroy` (`fipa_word_t *word`)
- `fipa_word_t * fipa_word_Copy` (`fipa_word_t *src`)
- `int fipa_word_Compose` (`dynstring_t *msg`, `fipa_word_t *word`)
- `fipa_string_t * fipa_string_New` (`void`)
- `int fipa_string_Destroy` (`fipa_string_t *str`)
- `fipa_string_t * fipa_string_Copy` (`fipa_string_t *src`)
- `int fipa_string_Compose` (`dynstring_t *msg`, `fipa_string_t *string`)

- [fipa\\_DateTime\\_t \\* fipa\\_DateTime\\_New](#) (void)
- [int fipa\\_DateTime\\_Destroy](#) (fipa\_DateTime\_t \*dt)
- [fipa\\_DateTime\\_t \\* fipa\\_DateTime\\_Copy](#) (fipa\_DateTime\_t \*src)
- [int fipa\\_DateTime\\_Compose](#) (dynstring\_t \*msg, fipa\_DateTime\_t \*date)
- [fipa\\_url\\_t \\* fipa\\_url\\_New](#) (void)
- [int fipa\\_url\\_Destroy](#) (fipa\_url\_t \*url)
- [fipa\\_url\\_t \\* fipa\\_url\\_Copy](#) (fipa\_url\_t \*src)
- [int fipa\\_url\\_Compose](#) (dynstring\_t \*msg, fipa\_url\_t \*url)
- [fipa\\_number\\_t \\* fipa\\_number\\_New](#) (void)
- [int fipa\\_number\\_Destroy](#) (fipa\_number\_t \*number)
- [fipa\\_number\\_t \\* fipa\\_number\\_Copy](#) (fipa\_number\_t \*src)
- [int fipa\\_number\\_Compose](#) (dynstring\_t \*msg, fipa\_number\_t \*number)
- [int fipa\\_acl\\_Parse](#) (struct fipa\_acl\_message\_s \*acl, fipa\_message\_string\_p message)
- [int fipa\\_message\\_parameter\\_Parse](#) (struct fipa\_acl\_message\_s \*acl, fipa\_message\_string\_p message)
- [int fipa\\_message\\_type\\_Parse](#) (enum fipa\_performative\_e \*performative, fipa\_message\_string\_p message)
- [int fipa\\_GetAtom](#) (fipa\_message\_string\_p message, char expected\_atom)
- [int fipa\\_word\\_Parse](#) (fipa\_word\_t \*\*word, fipa\_message\_string\_p message)
- [int fipa\\_CheckNextToken](#) (const fipa\_message\_string\_p message, const char \*token)
- [int fipa\\_expression\\_Parse](#) (fipa\_expression\_t \*\*expression, fipa\_message\_string\_p message)
- [int fipa\\_GetNextWord](#) (char \*\*word, const fipa\_message\_string\_p message)
- [int fipa\\_GetWholeToken](#) (char \*\*word, const fipa\_message\_string\_p message)
- [int fipa\\_datetime\\_Parse](#) (fipa\_DateTime\_p \*datetime, fipa\_message\_string\_p message)
- [int fipa\\_string\\_Parse](#) (fipa\_string\_p \*fipa\_string, fipa\_message\_string\_p message)
- [int fipa\\_agent\\_identifier\\_Parse](#) (fipa\_agent\_identifier\_p \*aid, fipa\_message\_string\_p message)
- [int fipa\\_agent\\_identifier\\_set\\_Parse](#) (fipa\_agent\_identifier\_set\_p \*agent\_ids, fipa\_message\_string\_p message)
- [int fipa\\_url\\_sequence\\_Parse](#) (fipa\_url\_sequence\_p \*urls, fipa\_message\_string\_p message)
- [int fipa\\_url\\_Parse](#) (fipa\_url\_p \*url, fipa\_message\_string\_p message)
- [struct fipa\\_acl\\_message\\_s \\* fipa\\_Reply](#) (struct fipa\_acl\_message\_s \*acl)

### 8.33.1 Typedef Documentation

#### 8.33.1.1 typedef fipa\_acl\_message\_t\* fipa\_acl\_message\_p

Definition at line 111 of file fipa\_acl.h.

#### 8.33.1.2 typedef struct fipa\_acl\_message\_s fipa\_acl\_message\_t

#### 8.33.1.3 typedef fipa\_agent\_identifier\_t\* fipa\_agent\_identifier\_p

Definition at line 161 of file fipa\_acl.h.

#### 8.33.1.4 typedef fipa\_agent\_identifier\_set\_t\* fipa\_agent\_identifier\_set\_p

Definition at line 147 of file fipa\_acl.h.

**8.33.1.5** `typedef struct fipa_agent_identifier_set_s fipa_agent_identifier_set_t`

**8.33.1.6** `typedef struct fipa_agent_identifier_s fipa_agent_identifier_t`

**8.33.1.7** `typedef fipa_DateTime_t* fipa_DateTime_p`

Definition at line 221 of file fipa\_acl.h.

**8.33.1.8** `typedef struct fipa_DateTime_s fipa_DateTime_t`

**8.33.1.9** `typedef fipa_expression_t* fipa_expression_p`

Definition at line 180 of file fipa\_acl.h.

**8.33.1.10** `typedef struct fipa_expression_s fipa_expression_t`

**8.33.1.11** `typedef enum fipa_expression_type_e fipa_expression_type_t`

**8.33.1.12** `typedef fipa_message_string_t* fipa_message_string_p`

Definition at line 123 of file fipa\_acl.h.

**8.33.1.13** `typedef struct fipa_message_string_s fipa_message_string_t`

**8.33.1.14** `typedef fipa_number_t* fipa_number_p`

Definition at line 243 of file fipa\_acl.h.

**8.33.1.15** `typedef struct fipa_number_s fipa_number_t`

**8.33.1.16** `typedef fipa_string_t* fipa_string_p`

Definition at line 202 of file fipa\_acl.h.

**8.33.1.17** `typedef struct fipa_string_s fipa_string_t`

**8.33.1.18** `typedef fipa_url_t* fipa_url_p`

Definition at line 232 of file fipa\_acl.h.

**8.33.1.19** `typedef fipa_url_sequence_t* fipa_url_sequence_p`

Definition at line 134 of file fipa\_acl.h.

**8.33.1.20** typedef struct fipa\_url\_sequence\_s fipa\_url\_sequence\_t

**8.33.1.21** typedef struct fipa\_url\_s fipa\_url\_t

**8.33.1.22** typedef fipa\_word\_t\* fipa\_word\_p

Definition at line 191 of file fipa\_acl.h.

**8.33.1.23** typedef struct fipa\_word\_s fipa\_word\_t

## 8.33.2 Enumeration Type Documentation

**8.33.2.1** enum fipa\_expression\_type\_e

Enumerator:

*FIPA\_EXPR\_ZERO*  
*FIPA\_EXPR\_WORD*  
*FIPA\_EXPR\_STRING*  
*FIPA\_EXPR\_NUMBER*  
*FIPA\_EXPR\_DATETIME*  
*FIPA\_EXPR\_EXPRESSION*

Definition at line 70 of file fipa\_acl.h.

**8.33.2.2** enum fipa\_performative\_e

Enumerator:

*FIPA\_ERROR*  
*FIPA\_ZERO*  
*FIPA\_ACCEPT\_PROPOSAL*  
*FIPA\_AGREE*  
*FIPA\_CANCEL*  
*FIPA\_CALL\_FOR\_PROPOSAL*  
*FIPA\_CONFIRM*  
*FIPA\_DISCONFIRM*  
*FIPA\_FAILURE*  
*FIPA\_INFORM*  
*FIPA\_INFORM\_IF*  
*FIPA\_INFORM\_REF*  
*FIPA\_NOT\_UNDERSTOOD*  
*FIPA\_PROPOGATE*  
*FIPA\_PROPOSE*  
*FIPA\_PROXY*  
*FIPA\_QUERY\_IF*

*FIPA\_QUERY\_REF**FIPA\_REFUSE**FIPA\_REJECT\_PROPOSAL**FIPA\_REQUEST**FIPA\_REQUEST\_WHEN**FIPA\_REQUEST\_WHENEVER**FIPA\_SUBSCRIBE*

Definition at line 40 of file `fipa_acl.h`.

### 8.33.3 Function Documentation

#### 8.33.3.1 `int fipa_acl_Compose (dynstring_t ** msg, fipa_acl_message_t * acl)`

Definition at line 1185 of file `fipa_acl.c`.

References `fipa_acl_message_s::content`, `fipa_acl_message_s::conversation_id`, `dynstring_Append()`, `dynstring_New()`, `fipa_acl_message_s::encoding`, `fipa_agent_identifier_Compose()`, `fipa_agent_identifier_set_Compose()`, `fipa_DateTime_Compose()`, `fipa_expression_Compose()`, `fipa_performative_Compose()`, `fipa_string_Compose()`, `fipa_word_Compose()`, `fipa_acl_message_s::in_reply_to`, `fipa_acl_message_s::language`, `fipa_acl_message_s::ontology`, `fipa_acl_message_s::performative`, `fipa_acl_message_s::protocol`, `fipa_acl_message_s::receiver`, `fipa_acl_message_s::reply_by`, `fipa_acl_message_s::reply_to`, `fipa_acl_message_s::reply_with`, and `fipa_acl_message_s::sender`.

Referenced by `MC_AclSend()`.

#### 8.33.3.2 `fipa_acl_message_t* fipa_acl_message_Copy (fipa_acl_message_t * src)`

Definition at line 81 of file `fipa_acl.c`.

References `fipa_acl_message_s::content`, `fipa_acl_message_s::conversation_id`, `fipa_acl_message_s::encoding`, `fipa_acl_message_New()`, `fipa_agent_identifier_Copy()`, `fipa_agent_identifier_set_Copy()`, `fipa_DateTime_Copy()`, `fipa_expression_Copy()`, `fipa_string_Copy()`, `fipa_word_Copy()`, `fipa_acl_message_s::in_reply_to`, `fipa_acl_message_s::language`, `fipa_acl_message_s::ontology`, `fipa_acl_message_s::performative`, `fipa_acl_message_s::protocol`, `fipa_acl_message_s::receiver`, `fipa_acl_message_s::reply_by`, `fipa_acl_message_s::reply_to`, `fipa_acl_message_s::reply_with`, and `fipa_acl_message_s::sender`.

#### 8.33.3.3 `int fipa_acl_message_Destroy (fipa_acl_message_t * message)`

Definition at line 60 of file `fipa_acl.c`.

References `fipa_acl_message_s::content`, `fipa_acl_message_s::conversation_id`, `fipa_acl_message_s::encoding`, `fipa_agent_identifier_Destroy()`, `fipa_agent_identifier_set_Destroy()`, `fipa_DateTime_Destroy()`, `fipa_expression_Destroy()`, `fipa_string_Destroy()`, `fipa_word_Destroy()`, `fipa_acl_message_s::in_reply_to`, `fipa_acl_message_s::language`, `fipa_acl_message_s::ontology`, `fipa_acl_message_s::protocol`, `fipa_acl_message_s::receiver`, `fipa_acl_message_s::reply_by`, `fipa_acl_message_s::reply_to`, `fipa_acl_message_s::reply_with`, and `fipa_acl_message_s::sender`.

Referenced by `acc_Thread()`, and `MC_AclDestroy()`.



**8.33.3.4 fipa\_acl\_message\_t\* fipa\_acl\_message\_New (void)**

Definition at line 52 of file fipa\_acl.c.

Referenced by acc\_Thread(), fipa\_acl\_message\_Copy(), fipa\_Reply(), and MC\_AclNew().

**8.33.3.5 int fipa\_acl\_Parse (struct fipa\_acl\_message\_s \* *acl*, fipa\_message\_string\_p *message*)****8.33.3.6 int fipa\_agent\_identifier\_Compose (dynstring\_t \* *msg*, fipa\_agent\_identifier\_t \* *id*)**

Definition at line 1355 of file fipa\_acl.c.

References fipa\_agent\_identifier\_s::addresses, dynstring\_Append(), fipa\_agent\_identifier\_set\_Compose(), fipa\_url\_sequence\_Compose(), fipa\_agent\_identifier\_s::name, fipa\_agent\_identifier\_set\_s::num, fipa\_url\_sequence\_s::num, and fipa\_agent\_identifier\_s::resolvers.

Referenced by fipa\_acl\_Compose(), and fipa\_agent\_identifier\_set\_Compose().

**8.33.3.7 fipa\_agent\_identifier\_t\* fipa\_agent\_identifier\_Copy (fipa\_agent\_identifier\_t \* *src*)**

Definition at line 229 of file fipa\_acl.c.

References fipa\_agent\_identifier\_s::addresses, fipa\_agent\_identifier\_New(), fipa\_agent\_identifier\_set\_Copy(), fipa\_url\_sequence\_Copy(), fipa\_agent\_identifier\_s::name, and fipa\_agent\_identifier\_s::resolvers.

Referenced by fipa\_acl\_message\_Copy(), fipa\_acl\_Param\_Copy(), fipa\_agent\_identifier\_set\_Copy(), and fipa\_Reply().

**8.33.3.8 int fipa\_agent\_identifier\_Destroy (fipa\_agent\_identifier\_t \* *id*)**

Definition at line 217 of file fipa\_acl.c.

References fipa\_agent\_identifier\_s::addresses, fipa\_agent\_identifier\_set\_Destroy(), fipa\_url\_sequence\_Destroy(), fipa\_agent\_identifier\_s::name, and fipa\_agent\_identifier\_s::resolvers.

Referenced by fipa\_acl\_message\_Destroy(), fipa\_acl\_Param\_Destroy(), fipa\_agent\_identifier\_set\_Destroy(), and MC\_AclSetSender().

**8.33.3.9 fipa\_agent\_identifier\_t\* fipa\_agent\_identifier\_New (void)**

Definition at line 209 of file fipa\_acl.c.

Referenced by fipa\_agent\_identifier\_Copy(), fipa\_envelope\_ParseAgentIdentifier(), MC\_AclAddReceiver(), MC\_AclAddReplyTo(), and MC\_AclSetSender().

**8.33.3.10 int fipa\_agent\_identifier\_Parse (fipa\_agent\_identifier\_p \* *aid*, fipa\_message\_string\_p *message*)**

Definition at line 985 of file fipa\_acl.c.

References CHECK\_NULL, fipa\_word\_s::content, fipa\_agent\_identifier\_set\_Parse(), fipa\_GetAtom(), fipa\_url\_sequence\_Parse(), fipa\_word\_Destroy(), fipa\_word\_Parse(), MC\_ERR\_PARSE, MC\_SUCCESS, and fipa\_message\_string\_s::parse.

Referenced by fipa\_agent\_identifier\_set\_Parse(), and fipa\_message\_parameter\_Parse().

### 8.33.3.11 `int fipa_agent_identifier_set_Compose (dynstring_t * msg, fipa_agent_identifier_set_t * ids)`

Definition at line 1342 of file fipa\_acl.c.

References `dynstring_Append()`, `fipa_agent_identifier_Compose()`, `fipa_agent_identifier_set_s::fipa_agent_identifiers`, and `fipa_agent_identifier_set_s::num`.

Referenced by `fipa_acl_Compose()`, and `fipa_agent_identifier_Compose()`.

### 8.33.3.12 `fipa_agent_identifier_set_t* fipa_agent_identifier_set_Copy (fipa_agent_identifier_set_t * src)`

Definition at line 188 of file fipa\_acl.c.

References `fipa_agent_identifier_Copy()`, `fipa_agent_identifier_set_New()`, `fipa_agent_identifier_set_s::fipa_agent_identifiers`, `fipa_agent_identifier_set_s::num`, and `fipa_agent_identifier_set_s::retain_order`.

Referenced by `fipa_acl_message_Copy()`, `fipa_acl_Param_Copy()`, `fipa_agent_identifier_Copy()`, and `fipa_Reply()`.

### 8.33.3.13 `int fipa_agent_identifier_set_Destroy (fipa_agent_identifier_set_t * idset)`

Definition at line 176 of file fipa\_acl.c.

References `fipa_agent_identifier_Destroy()`, `fipa_agent_identifier_set_s::fipa_agent_identifiers`, and `fipa_agent_identifier_set_s::num`.

Referenced by `fipa_acl_message_Destroy()`, `fipa_acl_Param_Destroy()`, and `fipa_agent_identifier_Destroy()`.

### 8.33.3.14 `fipa_agent_identifier_set_t* fipa_agent_identifier_set_New (void)`

Definition at line 168 of file fipa\_acl.c.

Referenced by `fipa_agent_identifier_set_Copy()`, `fipa_envelope_HandleIntendedReceiver()`, `fipa_envelope_HandleTo()`, `fipa_Reply()`, `MC_AclAddReceiver()`, and `MC_AclAddReplyTo()`.

### 8.33.3.15 `int fipa_agent_identifier_set_Parse (fipa_agent_identifier_set_p * agent_ids, fipa_message_string_p message)`

Definition at line 1144 of file fipa\_acl.c.

References `fipa_word_s::content`, `fipa_agent_identifier_Parse()`, `fipa_GetAtom()`, `fipa_word_Parse()`, `MC_ERR_PARSE`, `MC_SUCCESS`, and `fipa_agent_identifier_set_s::retain_order`.

Referenced by `fipa_agent_identifier_Parse()`, and `fipa_message_parameter_Parse()`.

### 8.33.3.16 `int fipa_CheckNextToken (const fipa_message_string_p message, const char * token)`

Definition at line 647 of file fipa\_acl.c.

References `fipa_message_string_s::parse`.

Referenced by `fipa_expression_Parse()`.

**8.33.3.17 int fipa\_DateTime\_Compose (dynstring\_t \* msg, fipa\_DateTime\_t \* date)**

Definition at line 1428 of file fipa\_acl.c.

References fipa\_DateTime\_s::day, dynstring\_Append(), fipa\_DateTime\_s::hour, fipa\_DateTime\_s::millisecond, fipa\_DateTime\_s::minute, fipa\_DateTime\_s::month, fipa\_DateTime\_s::second, fipa\_DateTime\_s::sign, and fipa\_DateTime\_s::year.

Referenced by fipa\_acl\_Compose(), and fipa\_expression\_Compose().

**8.33.3.18 fipa\_DateTime\_t\* fipa\_DateTime\_Copy (fipa\_DateTime\_t \* src)**

Definition at line 392 of file fipa\_acl.c.

References fipa\_DateTime\_New().

Referenced by fipa\_acl\_envelope\_Received\_Copy(), fipa\_acl\_message\_Copy(), fipa\_acl\_Param\_Copy(), and fipa\_expression\_Copy().

**8.33.3.19 int fipa\_DateTime\_Destroy (fipa\_DateTime\_t \* dt)**

Definition at line 385 of file fipa\_acl.c.

Referenced by fipa\_acl\_envelope\_Received\_Destroy(), fipa\_acl\_message\_Destroy(), fipa\_acl\_Param\_Destroy(), and fipa\_expression\_Destroy().

**8.33.3.20 fipa\_DateTime\_t\* fipa\_DateTime\_New (void)**

Definition at line 377 of file fipa\_acl.c.

Referenced by fipa\_DateTime\_Copy().

**8.33.3.21 int fipa\_datetime\_Parse (fipa\_DateTime\_p \* datetime, fipa\_message\_string\_p message)**

Definition at line 802 of file fipa\_acl.c.

References fipa\_GetWholeToken(), MC\_ERR\_PARSE, MC\_SUCCESS, fipa\_message\_string\_s::parse, and fipa\_DateTime\_s::sign.

Referenced by fipa\_envelope\_HandleDate(), fipa\_envelope\_HandleReceived(), fipa\_expression\_Parse(), and fipa\_message\_parameter\_Parse().

**8.33.3.22 int fipa\_expression\_Compose (dynstring\_t \* msg, fipa\_expression\_t \* expr)**

Definition at line 1381 of file fipa\_acl.c.

References fipa\_expression\_s::content, fipa\_expression\_s::content\_u::datetime, fipa\_expression\_s::content\_u::expression, fipa\_DateTime\_Compose(), FIPA\_EXPR\_DATETIME, FIPA\_EXPR\_EXPRESSION, FIPA\_EXPR\_NUMBER, FIPA\_EXPR\_STRING, FIPA\_EXPR\_WORD, fipa\_expression\_Compose(), fipa\_number\_Compose(), fipa\_string\_Compose(), fipa\_word\_Compose(), MC\_ERR\_PARSE, fipa\_expression\_s::content\_u::number, fipa\_expression\_s::content\_u::string, fipa\_expression\_s::type, and fipa\_expression\_s::content\_u::word.

Referenced by fipa\_acl\_Compose(), and fipa\_expression\_Compose().

**8.33.3.23 fipa\_expression\_t\* fipa\_expression\_Copy (fipa\_expression\_t \* src)**

Definition at line 280 of file fipa\_acl.c.

References fipa\_expression\_s::content, fipa\_expression\_s::content\_u::datetime, fipa\_expression\_s::content\_u::expression, fipa\_DateTime\_Copy(), FIPA\_EXPR\_DATETIME, FIPA\_EXPR\_EXPRESSION, FIPA\_EXPR\_NUMBER, FIPA\_EXPR\_STRING, FIPA\_EXPR\_WORD, fipa\_expression\_Copy(), fipa\_expression\_Destroy(), fipa\_expression\_New(), fipa\_number\_Copy(), fipa\_string\_Copy(), fipa\_word\_Copy(), fipa\_expression\_s::content\_u::number, fipa\_expression\_s::content\_u::string, fipa\_expression\_s::type, and fipa\_expression\_s::content\_u::word.

Referenced by fipa\_acl\_message\_Copy(), and fipa\_expression\_Copy().

**8.33.3.24 int fipa\_expression\_Destroy (fipa\_expression\_t \* expr)**

Definition at line 249 of file fipa\_acl.c.

References fipa\_expression\_s::content, fipa\_expression\_s::content\_u::datetime, fipa\_expression\_s::content\_u::expression, fipa\_DateTime\_Destroy(), FIPA\_EXPR\_DATETIME, FIPA\_EXPR\_EXPRESSION, FIPA\_EXPR\_NUMBER, FIPA\_EXPR\_STRING, FIPA\_EXPR\_WORD, fipa\_expression\_Destroy(), fipa\_number\_Destroy(), fipa\_string\_Destroy(), fipa\_word\_Destroy(), FREEMEM, fipa\_expression\_s::content\_u::number, fipa\_expression\_s::content\_u::string, fipa\_expression\_s::type, and fipa\_expression\_s::content\_u::word.

Referenced by fipa\_acl\_message\_Destroy(), fipa\_expression\_Copy(), and fipa\_expression\_Destroy().

**8.33.3.25 fipa\_expression\_t\* fipa\_expression\_New (void)**

Definition at line 241 of file fipa\_acl.c.

Referenced by fipa\_expression\_Copy().

**8.33.3.26 int fipa\_expression\_Parse (fipa\_expression\_t \*\* expression, fipa\_message\_string\_p message)**

Definition at line 666 of file fipa\_acl.c.

References fipa\_CheckNextToken(), fipa\_datetime\_Parse(), FIPA\_EXPR\_DATETIME, FIPA\_EXPR\_EXPRESSION, FIPA\_EXPR\_STRING, FIPA\_EXPR\_WORD, fipa\_expression\_Parse(), fipa\_GetAtom(), fipa\_string\_Parse(), fipa\_word\_Parse(), MC\_ERR\_PARSE, MC\_SUCCESS, and fipa\_expression\_s::type.

Referenced by fipa\_expression\_Parse(), and fipa\_message\_parameter\_Parse().

**8.33.3.27 int fipa\_GetAtom (fipa\_message\_string\_p message, char expected\_atom)**

Definition at line 582 of file fipa\_acl.c.

References MC\_ERR\_PARSE, MC\_SUCCESS, and fipa\_message\_string\_s::parse.

Referenced by fipa\_acl\_Parse(), fipa\_agent\_identifier\_Parse(), fipa\_agent\_identifier\_set\_Parse(), fipa\_expression\_Parse(), fipa\_message\_parameter\_Parse(), fipa\_string\_Parse(), and fipa\_url\_sequence\_Parse().

**8.33.3.28 int fipa\_GetNextWord (char \*\* *word*, const fipa\_message\_string\_p *message*)**

Definition at line 717 of file fipa\_acl.c.

References ERR, MC\_SUCCESS, and fipa\_message\_string\_s::parse.

**8.33.3.29 int fipa\_GetWholeToken (char \*\* *word*, const fipa\_message\_string\_p *message*)**

Definition at line 765 of file fipa\_acl.c.

References MC\_SUCCESS, and fipa\_message\_string\_s::parse.

Referenced by fipa\_datetime\_Parse().

**8.33.3.30 int fipa\_message\_parameter\_Parse (struct fipa\_acl\_message\_s \* *acl*, fipa\_message\_string\_p *message*)****8.33.3.31 fipa\_message\_string\_t\* fipa\_message\_string\_Copy (fipa\_message\_string\_t \* *src*)**

Definition at line 122 of file fipa\_acl.c.

References fipa\_message\_string\_s::message, and fipa\_message\_string\_s::parse.

**8.33.3.32 int fipa\_message\_string\_Destroy (fipa\_message\_string\_t \* *message*)**

Definition at line 112 of file fipa\_acl.c.

References fipa\_message\_string\_s::message.

Referenced by acc\_Thread().

**8.33.3.33 fipa\_message\_string\_t\* fipa\_message\_string\_New (void)**

Definition at line 104 of file fipa\_acl.c.

Referenced by acc\_Thread().

**8.33.3.34 int fipa\_message\_type\_Parse (enum fipa\_performative\_e \* *performative*, fipa\_message\_string\_p *message*)**

Definition at line 520 of file fipa\_acl.c.

References fipa\_word\_s::content, FIPA\_ACCEPT\_PROPOSAL, FIPA\_AGREE, FIPA\_CALL\_FOR\_PROPOSAL, FIPA\_CANCEL, FIPA\_CONFIRM, FIPA\_DISCONFIRM, FIPA\_FAILURE, FIPA\_INFORM, FIPA\_INFORM\_IF, FIPA\_INFORM\_REF, FIPA\_NOT\_UNDERSTOOD, FIPA\_PROPOGATE, FIPA\_PROPOSE, FIPA\_PROXY, FIPA\_QUERY\_IF, FIPA\_QUERY\_REF, FIPA\_REFUSE, FIPA\_REJECT\_PROPOSAL, FIPA\_REQUEST, FIPA\_REQUEST\_WHEN, FIPA\_REQUEST\_WHENEVER, FIPA\_SUBSCRIBE, fipa\_word\_Destroy(), fipa\_word\_Parse(), and MC\_ERR\_PARSE.

Referenced by fipa\_acl\_Parse().

**8.33.3.35 int fipa\_number\_Compose (dynstring\_t \* msg, fipa\_number\_t \* number)**

Definition at line 1456 of file fipa\_acl.c.

References dynstring\_Append(), and fipa\_number\_s::str.

Referenced by fipa\_expression\_Compose().

**8.33.3.36 fipa\_number\_t\* fipa\_number\_Copy (fipa\_number\_t \* src)**

Definition at line 448 of file fipa\_acl.c.

References fipa\_number\_New(), and fipa\_number\_s::str.

Referenced by fipa\_expression\_Copy().

**8.33.3.37 int fipa\_number\_Destroy (fipa\_number\_t \* number)**

Definition at line 438 of file fipa\_acl.c.

References fipa\_number\_s::str.

Referenced by fipa\_expression\_Destroy().

**8.33.3.38 fipa\_number\_t\* fipa\_number\_New (void)**

Definition at line 430 of file fipa\_acl.c.

Referenced by fipa\_number\_Copy().

**8.33.3.39 int fipa\_performative\_Compose (dynstring\_t \* msg, enum fipa\_performative\_e performative)**

Definition at line 1254 of file fipa\_acl.c.

References dynstring\_Append(), FIPA\_ACCEPT\_PROPOSAL, FIPA\_AGREE, FIPA\_CALL\_FOR\_PROPOSAL, FIPA\_CANCEL, FIPA\_CONFIRM, FIPA\_DISCONFIRM, FIPA\_FAILURE, FIPA\_INFORM, FIPA\_INFORM\_IF, FIPA\_INFORM\_REF, FIPA\_NOT\_UNDERSTOOD, FIPA\_PROPOGATE, FIPA\_PROPOSE, FIPA\_PROXY, FIPA\_QUERY\_IF, FIPA\_QUERY\_REF, FIPA\_REFUSE, FIPA\_REJECT\_PROPOSAL, FIPA\_REQUEST, FIPA\_REQUEST\_WHEN, FIPA\_REQUEST\_WHENEVER, FIPA\_SUBSCRIBE, and MC\_ERR\_PARSE.

Referenced by fipa\_acl\_Compose().

**8.33.3.40 struct fipa\_acl\_message\_s\* fipa\_Reply (struct fipa\_acl\_message\_s \* acl) [read]**

Definition at line 1464 of file fipa\_acl.c.

References fipa\_acl\_message\_New(), fipa\_agent\_identifier\_Copy(), fipa\_agent\_identifier\_set\_Copy(), fipa\_agent\_identifier\_set\_New(), fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_agent\_identifier\_set\_s::num, fipa\_acl\_message\_s::receiver, fipa\_acl\_message\_s::reply\_to, fipa\_agent\_identifier\_set\_s::retain\_order, and fipa\_acl\_message\_s::sender.

Referenced by MC\_AclReply().

**8.33.3.41 int fipa\_string\_Compose (dynstring\_t \* msg, fipa\_string\_t \* string)**

Definition at line 1419 of file fipa\_acl.c.

References fipa\_string\_s::content, and dynstring\_Append().

Referenced by fipa\_acl\_Compose(), and fipa\_expression\_Compose().

**8.33.3.42 fipa\_string\_t\* fipa\_string\_Copy (fipa\_string\_t \* src)**

Definition at line 367 of file fipa\_acl.c.

References fipa\_string\_s::content, and fipa\_string\_New().

Referenced by fipa\_acl\_message\_Copy(), and fipa\_expression\_Copy().

**8.33.3.43 int fipa\_string\_Destroy (fipa\_string\_t \* str)**

Definition at line 357 of file fipa\_acl.c.

References fipa\_string\_s::content.

Referenced by fipa\_acl\_message\_Destroy(), fipa\_expression\_Destroy(), and MC\_AclSetContent().

**8.33.3.44 fipa\_string\_t\* fipa\_string\_New (void)**

Definition at line 349 of file fipa\_acl.c.

Referenced by fipa\_string\_Copy(), and MC\_AclSetContent().

**8.33.3.45 int fipa\_string\_Parse (fipa\_string\_p \* fipa\_string, fipa\_message\_string\_p message)**

Definition at line 940 of file fipa\_acl.c.

References fipa\_string\_s::content, fipa\_GetAtom(), MC\_ERR\_PARSE, MC\_SUCCESS, and fipa\_message\_string\_s::parse.

Referenced by fipa\_expression\_Parse(), and fipa\_message\_parameter\_Parse().

**8.33.3.46 int fipa\_url\_Compose (dynstring\_t \* msg, fipa\_url\_t \* url)**

Definition at line 1448 of file fipa\_acl.c.

References dynstring\_Append(), and fipa\_url\_s::str.

Referenced by fipa\_url\_sequence\_Compose().

**8.33.3.47 fipa\_url\_t\* fipa\_url\_Copy (fipa\_url\_t \* src)**

Definition at line 420 of file fipa\_acl.c.

References fipa\_url\_New(), and fipa\_url\_s::str.

Referenced by fipa\_acl\_envelope\_Received\_Copy(), and fipa\_url\_sequence\_Copy().

**8.33.3.48 int fipa\_url\_Destroy (fipa\_url\_t \* url)**

Definition at line 410 of file fipa\_acl.c.

References fipa\_url\_s::str.

Referenced by fipa\_acl\_envelope\_Received\_Destroy(), and fipa\_url\_sequence\_Destroy().

**8.33.3.49 fipa\_url\_t\* fipa\_url\_New (void)**

Definition at line 402 of file fipa\_acl.c.

Referenced by fipa\_envelope\_HandleReceived(), fipa\_envelope\_ParseAddresses(), fipa\_url\_Copy(), MC\_AclAddReceiver(), MC\_AclAddReplyTo(), and MC\_AclSetSender().

**8.33.3.50 int fipa\_url\_Parse (fipa\_url\_p \* url, fipa\_message\_string\_p message)**

Definition at line 1124 of file fipa\_acl.c.

References fipa\_word\_s::content, fipa\_word\_Destroy(), and fipa\_word\_Parse().

Referenced by fipa\_url\_sequence\_Parse().

**8.33.3.51 int fipa\_url\_sequence\_Compose (dynstring\_t \* msg, fipa\_url\_sequence\_t \* urls)**

Definition at line 1329 of file fipa\_acl.c.

References dynstring\_Append(), fipa\_url\_Compose(), fipa\_url\_sequence\_s::num, and fipa\_url\_sequence\_s::urls.

Referenced by fipa\_agent\_identifier\_Compose().

**8.33.3.52 fipa\_url\_sequence\_t\* fipa\_url\_sequence\_Copy (fipa\_url\_sequence\_t \* src)**

Definition at line 152 of file fipa\_acl.c.

References fipa\_url\_Copy(), fipa\_url\_sequence\_New(), fipa\_url\_sequence\_s::num, and fipa\_url\_sequence\_s::urls.

Referenced by fipa\_agent\_identifier\_Copy().

**8.33.3.53 int fipa\_url\_sequence\_Destroy (fipa\_url\_sequence\_t \* sequence)**

Definition at line 140 of file fipa\_acl.c.

References fipa\_url\_Destroy(), fipa\_url\_sequence\_s::num, and fipa\_url\_sequence\_s::urls.

Referenced by fipa\_agent\_identifier\_Destroy().

**8.33.3.54 fipa\_url\_sequence\_t\* fipa\_url\_sequence\_New (void)**

Definition at line 132 of file fipa\_acl.c.

Referenced by fipa\_url\_sequence\_Copy(), fipa\_url\_sequence\_Parse(), MC\_AclAddReceiver(), MC\_AclAddReplyTo(), and MC\_AclSetSender().



**8.33.3.55 int fipa\_url\_sequence\_Parse (fipa\_url\_sequence\_p \* *urls*, fipa\_message\_string\_p *message*)**

Definition at line 1092 of file fipa\_acl.c.

References fipa\_word\_s::content, fipa\_GetAtom(), fipa\_url\_Parse(), fipa\_url\_sequence\_New(), fipa\_word\_Destroy(), fipa\_word\_Parse(), and MC\_ERR\_PARSE.

Referenced by fipa\_agent\_identifier\_Parse().

**8.33.3.56 int fipa\_word\_Compose (dynstring\_t \* *msg*, fipa\_word\_t \* *word*)**

Definition at line 1411 of file fipa\_acl.c.

References fipa\_word\_s::content, and dynstring\_Append().

Referenced by fipa\_acl\_Compose(), and fipa\_expression\_Compose().

**8.33.3.57 fipa\_word\_t\* fipa\_word\_Copy (fipa\_word\_t \* *src*)**

Definition at line 339 of file fipa\_acl.c.

References fipa\_word\_s::content, and fipa\_word\_New().

Referenced by fipa\_acl\_message\_Copy(), and fipa\_expression\_Copy().

**8.33.3.58 int fipa\_word\_Destroy (fipa\_word\_t \* *word*)**

Definition at line 329 of file fipa\_acl.c.

References fipa\_word\_s::content.

Referenced by fipa\_acl\_message\_Destroy(), fipa\_agent\_identifier\_Parse(), fipa\_expression\_Destroy(), fipa\_message\_parameter\_Parse(), fipa\_message\_type\_Parse(), fipa\_url\_Parse(), and fipa\_url\_sequence\_Parse().

**8.33.3.59 fipa\_word\_t\* fipa\_word\_New (void)**

Definition at line 321 of file fipa\_acl.c.

Referenced by fipa\_word\_Copy().

**8.33.3.60 int fipa\_word\_Parse (fipa\_word\_t \*\* *word*, fipa\_message\_string\_p *message*)**

Definition at line 605 of file fipa\_acl.c.

References CHECK\_NULL, MC\_ERR\_PARSE, MC\_SUCCESS, and fipa\_message\_string\_s::parse.

Referenced by fipa\_agent\_identifier\_Parse(), fipa\_agent\_identifier\_set\_Parse(), fipa\_expression\_Parse(), fipa\_message\_parameter\_Parse(), fipa\_message\_type\_Parse(), fipa\_url\_Parse(), and fipa\_url\_sequence\_Parse().

## 8.34 /home/dko/projects/mobilec/trunk/src/include/fipa\_acl\_envelope.h File Reference

```
#include "fipa_acl.h"
```

### Data Structures

- struct [fipa\\_acl\\_envelope\\_Received\\_s](#)
- struct [fipa\\_acl\\_Param\\_s](#)
- struct [fipa\\_acl\\_envelope\\_s](#)

### Typedefs

- typedef struct [fipa\\_acl\\_envelope\\_Received\\_s](#) [fipa\\_acl\\_envelope\\_Received\\_t](#)
- typedef struct [fipa\\_acl\\_Param\\_s](#) [fipa\\_acl\\_Param\\_t](#)
- typedef struct [fipa\\_acl\\_envelope\\_s](#) [fipa\\_acl\\_envelope\\_t](#)
- typedef [fipa\\_acl\\_envelope\\_t](#) \* [fipa\\_acl\\_envelope\\_p](#)

### Functions

- [fipa\\_acl\\_envelope\\_Received\\_t](#) \* [fipa\\_acl\\_envelope\\_Received\\_New](#) (void)
- int [fipa\\_acl\\_envelope\\_Received\\_Destroy](#) ([fipa\\_acl\\_envelope\\_Received\\_t](#) \*received)
- [fipa\\_acl\\_envelope\\_Received\\_t](#) \* [fipa\\_acl\\_envelope\\_Received\\_Copy](#) ([fipa\\_acl\\_envelope\\_Received\\_t](#) \*received)
- [fipa\\_acl\\_Param\\_t](#) \* [fipa\\_acl\\_Param\\_New](#) (void)
- int [fipa\\_acl\\_Param\\_Destroy](#) ([fipa\\_acl\\_Param\\_t](#) \*param)
- [fipa\\_acl\\_Param\\_t](#) \* [fipa\\_acl\\_Param\\_Copy](#) ([fipa\\_acl\\_Param\\_t](#) \*param)
- [fipa\\_acl\\_envelope\\_t](#) \* [fipa\\_acl\\_envelope\\_New](#) (void)
- int [fipa\\_acl\\_envelope\\_Destroy](#) ([fipa\\_acl\\_envelope\\_t](#) \*envelope)
- [fipa\\_acl\\_envelope\\_t](#) \* [fipa\\_acl\\_envelope\\_Copy](#) ([fipa\\_acl\\_envelope\\_t](#) \*envelope)
- int [fipa\\_envelope\\_Parse](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, const char \*message)
- int [fipa\\_envelope\\_HandleEnvelope](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, [mxml\\_node\\_t](#) \*node)
- int [fipa\\_envelope\\_HandleParams](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, [mxml\\_node\\_t](#) \*node)
- int [fipa\\_envelope\\_HandleTo](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, [mxml\\_node\\_t](#) \*param\_node, int cur\_param)
- int [fipa\\_envelope\\_HandleFrom](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, [mxml\\_node\\_t](#) \*param\_node, int cur\_param)
- int [fipa\\_envelope\\_HandleComments](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, [mxml\\_node\\_t](#) \*param\_node, int cur\_param)
- int [fipa\\_envelope\\_HandleAclRepresentation](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, [mxml\\_node\\_t](#) \*param\_node, int cur\_param)
- int [fipa\\_envelope\\_HandlePayloadLength](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, [mxml\\_node\\_t](#) \*param\_node, int cur\_param)
- int [fipa\\_envelope\\_HandlePayloadEncoding](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, [mxml\\_node\\_t](#) \*param\_node, int cur\_param)
- int [fipa\\_envelope\\_HandleDate](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, [mxml\\_node\\_t](#) \*param\_node, int cur\_param)
- int [fipa\\_envelope\\_HandleIntendedReceiver](#) (struct [fipa\\_acl\\_envelope\\_s](#) \*envelope, [mxml\\_node\\_t](#) \*param\_node, int cur\_param)

- `int fipa_envelope_HandleReceived` (struct `fipa_acl_envelope_s` \*envelope, `mxml_node_t` \*param\_node, `int` cur\_param)
- `int fipa_envelope_ParseAgentIdentifier` (struct `fipa_agent_identifier_s` \*\*aid, `mxml_node_t` \*agent\_identifier\_node)
- `int fipa_envelope_ParseAddresses` (struct `fipa_agent_identifier_s` \*aid, `mxml_node_t` \*addresses\_node)
- `int fipa_envelope_ParseResolvers` (struct `fipa_agent_identifier_s` \*aid, `mxml_node_t` \*resolvers\_node)
- `char * fipa_envelope_Compose` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__envelope` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__params` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__to` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__from` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__acl_representation` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__payload_encoding` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__date` (`fipa_acl_message_t` \*fipa\_acl)
- `mxml_node_t * fipa_envelope_Compose__intended_receiver` (`fipa_acl_message_t` \*fipa\_acl)

### 8.34.1 Typedef Documentation

#### 8.34.1.1 `typedef fipa_acl_envelope_t* fipa_acl_envelope_p`

Definition at line 45 of file `fipa_acl_envelope.h`.

#### 8.34.1.2 `typedef struct fipa_acl_envelope_Received_s fipa_acl_envelope_Received_t`

#### 8.34.1.3 `typedef struct fipa_acl_envelope_s fipa_acl_envelope_t`

#### 8.34.1.4 `typedef struct fipa_acl_Param_s fipa_acl_Param_t`

### 8.34.2 Function Documentation

#### 8.34.2.1 `fipa_acl_envelope_t* fipa_acl_envelope_Copy` (`fipa_acl_envelope_t` \* envelope)

Definition at line 145 of file `fipa_envelope.c`.

References `fipa_acl_envelope_New()`, `fipa_acl_Param_Copy()`, `fipa_acl_envelope_s::num_params`, and `fipa_acl_envelope_s::params`.

#### 8.34.2.2 `int fipa_acl_envelope_Destroy` (`fipa_acl_envelope_t` \* envelope)

Definition at line 132 of file `fipa_envelope.c`.

References `fipa_acl_Param_Destroy()`, `fipa_acl_envelope_s::num_params`, and `fipa_acl_envelope_s::params`.

Referenced by `acc_Thread()`.

**8.34.2.3 fipa\_acl\_envelope\_t\* fipa\_acl\_envelope\_New (void)**

Definition at line 124 of file fipa\_envelope.c.

Referenced by acc\_Thread(), and fipa\_acl\_envelope\_Copy().

**8.34.2.4 fipa\_acl\_envelope\_Received\_t\* fipa\_acl\_envelope\_Received\_Copy (fipa\_acl\_envelope\_Received\_t \* *received*)**

Definition at line 64 of file fipa\_envelope.c.

References fipa\_acl\_envelope\_Received\_New(), fipa\_DateTime\_Copy(), fipa\_url\_Copy(), fipa\_acl\_envelope\_Received\_s::received\_by, fipa\_acl\_envelope\_Received\_s::received\_date, fipa\_acl\_envelope\_Received\_s::received\_from, fipa\_acl\_envelope\_Received\_s::received\_id, and fipa\_acl\_envelope\_Received\_s::received\_via.

Referenced by fipa\_acl\_Param\_Copy().

**8.34.2.5 int fipa\_acl\_envelope\_Received\_Destroy (fipa\_acl\_envelope\_Received\_t \* *received*)**

Definition at line 52 of file fipa\_envelope.c.

References fipa\_DateTime\_Destroy(), fipa\_url\_Destroy(), fipa\_acl\_envelope\_Received\_s::received\_by, fipa\_acl\_envelope\_Received\_s::received\_date, fipa\_acl\_envelope\_Received\_s::received\_from, fipa\_acl\_envelope\_Received\_s::received\_id, and fipa\_acl\_envelope\_Received\_s::received\_via.

Referenced by fipa\_acl\_Param\_Destroy().

**8.34.2.6 fipa\_acl\_envelope\_Received\_t\* fipa\_acl\_envelope\_Received\_New (void)**

Definition at line 43 of file fipa\_envelope.c.

Referenced by fipa\_acl\_envelope\_Received\_Copy(), and fipa\_envelope\_HandleReceived().

**8.34.2.7 fipa\_acl\_Param\_t\* fipa\_acl\_Param\_Copy (fipa\_acl\_Param\_t \* *param*)**

Definition at line 105 of file fipa\_envelope.c.

References fipa\_acl\_Param\_s::acl\_representation, fipa\_acl\_Param\_s::comments, fipa\_acl\_Param\_s::date, fipa\_acl\_envelope\_Received\_Copy(), fipa\_acl\_Param\_New(), fipa\_agent\_identifier\_Copy(), fipa\_agent\_identifier\_set\_Copy(), fipa\_DateTime\_Copy(), fipa\_acl\_Param\_s::from, fipa\_acl\_Param\_s::intended\_receiver, fipa\_acl\_Param\_s::payload\_encoding, fipa\_acl\_Param\_s::payload\_length, fipa\_acl\_Param\_s::received, and fipa\_acl\_Param\_s::to.

Referenced by fipa\_acl\_envelope\_Copy().

**8.34.2.8 int fipa\_acl\_Param\_Destroy (fipa\_acl\_Param\_t \* *param*)**

Definition at line 88 of file fipa\_envelope.c.

References fipa\_acl\_Param\_s::acl\_representation, fipa\_acl\_Param\_s::comments, fipa\_acl\_Param\_s::date, fipa\_acl\_envelope\_Received\_Destroy(), fipa\_agent\_identifier\_Destroy(), fipa\_agent\_identifier\_set\_Destroy(), fipa\_DateTime\_Destroy(), fipa\_acl\_Param\_s::from, fipa\_acl\_Param\_s::intended\_receiver, fipa\_acl\_Param\_s::payload\_encoding, fipa\_acl\_Param\_s::payload\_length, fipa\_acl\_Param\_s::received, and fipa\_acl\_Param\_s::to.

Referenced by fipa\_acl\_envelope\_Destroy().

#### 8.34.2.9 fipa\_acl\_Param\_t\* fipa\_acl\_Param\_New (void)

Definition at line 80 of file fipa\_envelope.c.

Referenced by fipa\_acl\_Param\_Copy(), and fipa\_envelope\_HandleTo().

#### 8.34.2.10 char\* fipa\_envelope\_Compose (fipa\_acl\_message\_t \* *fipa\_acl*)

Definition at line 864 of file fipa\_envelope.c.

References fipa\_envelope\_Compose\_\_envelope(), MXML\_ADD\_AFTER, MXML\_ADD\_TO\_PARENT, MXML\_NO\_CALLBACK, mxmlAdd(), mxmlDelete(), mxmlLoadString(), mxmlSaveAllocString(), and node.

Referenced by MC\_AclSend().

#### 8.34.2.11 mxml\_node\_t\* fipa\_envelope\_Compose\_\_acl\_representation (fipa\_acl\_message\_t \* *fipa\_acl*)

Definition at line 1081 of file fipa\_envelope.c.

References mxmlNewElement(), mxmlNewText(), and node.

Referenced by fipa\_envelope\_Compose\_\_params().

#### 8.34.2.12 mxml\_node\_t\* fipa\_envelope\_Compose\_\_date (fipa\_acl\_message\_t \* *fipa\_acl*)

Definition at line 1109 of file fipa\_envelope.c.

References mxmlNewElement(), mxmlNewText(), and node.

Referenced by fipa\_envelope\_Compose\_\_params().

#### 8.34.2.13 mxml\_node\_t\* fipa\_envelope\_Compose\_\_envelope (fipa\_acl\_message\_t \* *fipa\_acl*)

Definition at line 889 of file fipa\_envelope.c.

References fipa\_envelope\_Compose\_\_params(), MXML\_ADD\_AFTER, MXML\_ADD\_TO\_PARENT, mxmlAdd(), mxmlNewElement(), and node.

Referenced by fipa\_envelope\_Compose().

#### 8.34.2.14 mxml\_node\_t\* fipa\_envelope\_Compose\_\_from (fipa\_acl\_message\_t \* *fipa\_acl*)

Definition at line 1016 of file fipa\_envelope.c.

References fipa\_agent\_identifier\_s::addresses, dynstring\_Append(), dynstring\_Destroy(), dynstring\_New(), g\_mc\_platform, mc\_platform\_s::hostname, dynstring\_s::message, mxmlNewElement(), mxmlNewText(), fipa\_agent\_identifier\_s::name, fipa\_url\_sequence\_s::num, mc\_platform\_s::port, and fipa\_acl\_message\_s::sender.

Referenced by fipa\_envelope\_Compose\_\_params().

#### 8.34.2.15 **mxml\_node\_t\* fipa\_envelope\_Compose\_\_intended\_receiver (fipa\_acl\_message\_t \* *fipa\_acl*)**

Definition at line 1138 of file fipa\_envelope.c.

References fipa\_agent\_identifier\_s::addresses, fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, mxml-NewElement(), mxmlNewText(), fipa\_agent\_identifier\_s::name, node, fipa\_url\_sequence\_s::num, fipa\_acl\_message\_s::receiver, fipa\_url\_s::str, and fipa\_url\_sequence\_s::urls.

Referenced by fipa\_envelope\_Compose\_\_params().

#### 8.34.2.16 **mxml\_node\_t\* fipa\_envelope\_Compose\_\_params (fipa\_acl\_message\_t \* *fipa\_acl*)**

Definition at line 909 of file fipa\_envelope.c.

References fipa\_envelope\_Compose\_\_acl\_representation(), fipa\_envelope\_Compose\_\_date(), fipa\_envelope\_Compose\_\_from(), fipa\_envelope\_Compose\_\_intended\_receiver(), fipa\_envelope\_Compose\_\_payload\_encoding(), fipa\_envelope\_Compose\_\_to(), MXML\_ADD\_AFTER, MXML\_ADD\_TO\_PARENT, mxmlAdd(), mxmlElementSetAttr(), mxmlNewElement(), and node.

Referenced by fipa\_envelope\_Compose\_\_envelope().

#### 8.34.2.17 **mxml\_node\_t\* fipa\_envelope\_Compose\_\_payload\_encoding (fipa\_acl\_message\_t \* *fipa\_acl*)**

Definition at line 1095 of file fipa\_envelope.c.

References mxmlNewElement(), mxmlNewText(), and node.

Referenced by fipa\_envelope\_Compose\_\_params().

#### 8.34.2.18 **mxml\_node\_t\* fipa\_envelope\_Compose\_\_to (fipa\_acl\_message\_t \* *fipa\_acl*)**

Definition at line 970 of file fipa\_envelope.c.

References fipa\_agent\_identifier\_s::addresses, fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, mxml-NewElement(), mxmlNewText(), fipa\_agent\_identifier\_s::name, node, fipa\_url\_sequence\_s::num, fipa\_agent\_identifier\_set\_s::num, fipa\_acl\_message\_s::receiver, fipa\_url\_s::str, and fipa\_url\_sequence\_s::urls.

Referenced by fipa\_envelope\_Compose\_\_params().

#### 8.34.2.19 **int fipa\_envelope\_HandleAclRepresentation (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *param\_node*, int *cur\_param*)**

Definition at line 376 of file fipa\_envelope.c.

References fipa\_acl\_Param\_s::acl\_representation, mxml\_node\_s::child, MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_TEXT, mxmlFindElement(), fipa\_acl\_envelope\_s::params, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, and mxml\_node\_s::value.

Referenced by fipa\_envelope\_HandleParams().

**8.34.2.20** `int fipa_envelope_HandleComments (struct fipa_acl_envelope_s * envelope,  
mxml_node_t * param_node, int cur_param)`

Definition at line 352 of file fipa\_envelope.c.

References mxml\_node\_s::child, fipa\_acl\_Param\_s::comments, MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_TEXT, mxmlFindElement(), fipa\_acl\_envelope\_s::params, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, and mxml\_node\_s::value.

Referenced by fipa\_envelope\_HandleParams().

**8.34.2.21** `int fipa_envelope_HandleDate (struct fipa_acl_envelope_s * envelope, mxml_node_t *  
param_node, int cur_param)`

Definition at line 447 of file fipa\_envelope.c.

References mxml\_node\_s::child, fipa\_acl\_Param\_s::date, fipa\_datetime\_Parse(), MC\_ERR\_PARSE, fipa\_message\_string\_s::message, MXML\_DESCEND\_FIRST, MXML\_TEXT, mxmlFindElement(), fipa\_acl\_envelope\_s::params, fipa\_message\_string\_s::parse, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, and mxml\_node\_s::value.

Referenced by fipa\_envelope\_HandleParams().

**8.34.2.22** `int fipa_envelope_HandleEnvelope (struct fipa_acl_envelope_s * envelope, mxml_node_t  
* node)`

Definition at line 173 of file fipa\_envelope.c.

References fipa\_envelope\_HandleParams(), MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, and mxmlFindElement().

Referenced by fipa\_envelope\_Parse().

**8.34.2.23** `int fipa_envelope_HandleFrom (struct fipa_acl_envelope_s * envelope, mxml_node_t *  
param_node, int cur_param)`

Definition at line 323 of file fipa\_envelope.c.

References fipa\_envelope\_ParseAgentIdentifier(), fipa\_acl\_Param\_s::from, MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, mxmlFindElement(), and fipa\_acl\_envelope\_s::params.

Referenced by fipa\_envelope\_HandleParams().

**8.34.2.24** `int fipa_envelope_HandleIntendedReceiver (struct fipa_acl_envelope_s * envelope,  
mxml_node_t * param_node, int cur_param)`

Definition at line 477 of file fipa\_envelope.c.

References fipa\_agent\_identifier\_set\_New(), fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_envelope\_ParseAgentIdentifier(), fipa\_acl\_Param\_s::intended\_receiver, MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_NO\_DESCEND, mxmlFindElement(), fipa\_agent\_identifier\_set\_s::num, fipa\_acl\_envelope\_s::params, and fipa\_agent\_identifier\_set\_s::retain\_order.

Referenced by fipa\_envelope\_HandleParams().

### 8.34.2.25 **int fipa\_envelope\_HandleParams** (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *node*)

Definition at line 193 of file fipa\_envelope.c.

References fipa\_envelope\_HandleAclRepresentation(), fipa\_envelope\_HandleComments(), fipa\_envelope\_HandleDate(), fipa\_envelope\_HandleFrom(), fipa\_envelope\_HandleIntendedReceiver(), fipa\_envelope\_HandlePayloadEncoding(), fipa\_envelope\_HandlePayloadLength(), fipa\_envelope\_HandleReceived(), fipa\_envelope\_HandleTo(), MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, mxmlFindElement(), fipa\_acl\_envelope\_s::num\_params, and fipa\_acl\_envelope\_s::params.

Referenced by fipa\_envelope\_HandleEnvelope().

### 8.34.2.26 **int fipa\_envelope\_HandlePayloadEncoding** (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *param\_node*, int *cur\_param*)

Definition at line 423 of file fipa\_envelope.c.

References mxml\_node\_s::child, MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_TEXT, mxmlFindElement(), fipa\_acl\_envelope\_s::params, fipa\_acl\_Param\_s::payload\_encoding, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, and mxml\_node\_s::value.

Referenced by fipa\_envelope\_HandleParams().

### 8.34.2.27 **int fipa\_envelope\_HandlePayloadLength** (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *param\_node*, int *cur\_param*)

Definition at line 400 of file fipa\_envelope.c.

References mxml\_node\_s::child, MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_TEXT, mxmlFindElement(), fipa\_acl\_envelope\_s::params, fipa\_acl\_Param\_s::payload\_length, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, and mxml\_node\_s::value.

Referenced by fipa\_envelope\_HandleParams().

### 8.34.2.28 **int fipa\_envelope\_HandleReceived** (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *param\_node*, int *cur\_param*)

Definition at line 555 of file fipa\_envelope.c.

References fipa\_acl\_envelope\_Received\_New(), fipa\_datetime\_Parse(), fipa\_url\_New(), MC\_ERR\_PARSE, fipa\_message\_string\_s::message, MXML\_DESCEND\_FIRST, mxmlElementGetAttr(), mxmlFindElement(), node, fipa\_acl\_envelope\_s::params, fipa\_message\_string\_s::parse, fipa\_acl\_Param\_s::received, fipa\_acl\_envelope\_Received\_s::received\_by, fipa\_acl\_envelope\_Received\_s::received\_date, fipa\_acl\_envelope\_Received\_s::received\_from, fipa\_acl\_envelope\_Received\_s::received\_id, fipa\_acl\_envelope\_Received\_s::received\_via, and fipa\_url\_s::str.

Referenced by fipa\_envelope\_HandleParams().

### 8.34.2.29 **int fipa\_envelope\_HandleTo** (struct fipa\_acl\_envelope\_s \* *envelope*, mxml\_node\_t \* *param\_node*, int *cur\_param*)

Definition at line 245 of file fipa\_envelope.c.

References fipa\_acl\_Param\_New(), fipa\_agent\_identifier\_set\_New(), fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_envelope\_ParseAgentIdentifier(), MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST,



MXML\_NO\_DESCEND, mxmlFindElement(), fipa\_agent\_identifier\_set\_s::num, fipa\_acl\_envelope\_s::params, fipa\_agent\_identifier\_set\_s::retain\_order, and fipa\_acl\_Param\_s::to.

Referenced by fipa\_envelope\_HandleParams().

#### 8.34.2.30 int fipa\_envelope\_Parse (struct fipa\_acl\_envelope\_s \* *envelope*, const char \* *message*)

Definition at line 160 of file fipa\_envelope.c.

References fipa\_envelope\_HandleEnvelope(), MXML\_NO\_CALLBACK, mxmlDelete(), and mxmlLoadString().

Referenced by acc\_Thread().

#### 8.34.2.31 int fipa\_envelope\_ParseAddresses (struct fipa\_agent\_identifier\_s \* *aid*, mxml\_node\_t \* *addresses\_node*)

Definition at line 732 of file fipa\_envelope.c.

References fipa\_agent\_identifier\_s::addresses, mxml\_node\_s::child, fipa\_url\_New(), MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_NO\_DESCEND, MXML\_TEXT, mxmlFindElement(), fipa\_url\_sequence\_s::num, fipa\_url\_s::str, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, fipa\_url\_sequence\_s::urls, and mxml\_node\_s::value.

Referenced by fipa\_envelope\_ParseAgentIdentifier().

#### 8.34.2.32 int fipa\_envelope\_ParseAgentIdentifier (struct fipa\_agent\_identifier\_s \*\* *aid*, mxml\_node\_t \* *agent\_identifier\_node*)

Definition at line 682 of file fipa\_envelope.c.

References mxml\_node\_s::child, fipa\_agent\_identifier\_New(), fipa\_envelope\_ParseAddresses(), fipa\_envelope\_ParseResolvers(), MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_TEXT, mxmlFindElement(), mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, and mxml\_node\_s::value.

Referenced by fipa\_envelope\_HandleFrom(), fipa\_envelope\_HandleIntendedReceiver(), fipa\_envelope\_HandleTo(), and fipa\_envelope\_ParseResolvers().

#### 8.34.2.33 int fipa\_envelope\_ParseResolvers (struct fipa\_agent\_identifier\_s \* *aid*, mxml\_node\_t \* *resolvers\_node*)

Definition at line 799 of file fipa\_envelope.c.

References fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_envelope\_ParseAgentIdentifier(), MC\_ERR\_PARSE, MXML\_DESCEND\_FIRST, MXML\_NO\_DESCEND, mxmlFindElement(), fipa\_agent\_identifier\_set\_s::num, fipa\_agent\_identifier\_s::resolvers, and fipa\_agent\_identifier\_set\_s::retain\_order.

Referenced by fipa\_envelope\_ParseAgentIdentifier().

## 8.35 /home/dko/projects/mobilec/trunk/src/include/host\_id.h File Reference

### Data Structures

- struct [host\\_id\\_s](#)

### Typedefs

- typedef struct [host\\_id\\_s](#) [host\\_id\\_t](#)
- typedef [host\\_id\\_t](#) \* [host\\_id\\_p](#)

### Functions

- [host\\_id\\_p](#) [host\\_id\\_Initialize](#) (const char \*hostname, [int](#) port)
- void [host\\_id\\_Destroy](#) ([host\\_id\\_p](#) host\_id)
- [int](#) [host\\_id\\_Compare](#) ([host\\_id\\_p](#) host1, [host\\_id\\_p](#) host2)

### 8.35.1 Typedef Documentation

#### 8.35.1.1 typedef [host\\_id\\_t](#)\* [host\\_id\\_p](#)

Definition at line 43 of file [host\\_id.h](#).

#### 8.35.1.2 typedef struct [host\\_id\\_s](#) [host\\_id\\_t](#)

### 8.35.2 Function Documentation

#### 8.35.2.1 [int](#) [host\\_id\\_Compare](#) ([host\\_id\\_p](#) *host1*, [host\\_id\\_p](#) *host2*)

#### 8.35.2.2 void [host\\_id\\_Destroy](#) ([host\\_id\\_p](#) *host\_id*)

#### 8.35.2.3 [host\\_id\\_p](#) [host\\_id\\_Initialize](#) (const char \* *hostname*, [int](#) *port*)

## 8.36 /home/dko/projects/mobilec/trunk/src/include/interpreter\_variable\_data.h File Reference

```
#include <ch.h>
```

### Data Structures

- struct [interpreter\\_variable\\_data\\_s](#)

### Typedefs

- typedef struct [interpreter\\_variable\\_data\\_s](#) [interpreter\\_variable\\_data\\_t](#)
- typedef [interpreter\\_variable\\_data\\_t](#) \* [interpreter\\_variable\\_data\\_p](#)

### Functions

- [interpreter\\_variable\\_data\\_p](#) [interpreter\\_variable\\_data\\_New](#) (void)
- [interpreter\\_variable\\_data\\_p](#) [interpreter\\_variable\\_data\\_Copy](#) ([interpreter\\_variable\\_data\\_p](#) src)
- [interpreter\\_variable\\_data\\_p](#) [interpreter\\_variable\\_data\\_InitializeFromAgent](#) (struct [agent\\_s](#) \*agent)
- [interpreter\\_variable\\_data\\_p](#) [interpreter\\_variable\\_data\\_Initialize](#) (struct [agent\\_s](#) \*agent, const char \*varname)
- int [interpreter\\_variable\\_data\\_Destroy](#) ([interpreter\\_variable\\_data\\_p](#) agent\_return\_data)

### 8.36.1 Typedef Documentation

#### 8.36.1.1 typedef [interpreter\\_variable\\_data\\_t](#)\* [interpreter\\_variable\\_data\\_p](#)

Definition at line 51 of file [interpreter\\_variable\\_data.h](#).

#### 8.36.1.2 typedef struct [interpreter\\_variable\\_data\\_s](#) [interpreter\\_variable\\_data\\_t](#)

Definition at line 50 of file [interpreter\\_variable\\_data.h](#).

### 8.36.2 Function Documentation

#### 8.36.2.1 [interpreter\\_variable\\_data\\_p](#) [interpreter\\_variable\\_data\\_Copy](#) ([interpreter\\_variable\\_data\\_p](#) src)

Definition at line 229 of file [agent\\_return\\_data.c](#).

References [interpreter\\_variable\\_data\\_s::array\\_dim](#), [interpreter\\_variable\\_data\\_s::array\\_extent](#), [interpreter\\_variable\\_data\\_s::data](#), [interpreter\\_variable\\_data\\_s::data\\_type](#), [interpreter\\_variable\\_data\\_New\(\)](#), [interpreter\\_variable\\_data\\_s::name](#), and [interpreter\\_variable\\_data\\_s::size](#).

**8.36.2.2 int interpreter\_variable\_data\_Destroy (interpreter\_variable\_data\_p agent\_return\_data)**

Definition at line 210 of file agent\_return\_data.c.

References interpreter\_variable\_data\_s::array\_extent, interpreter\_variable\_data\_s::data, MC\_SUCCESS, and interpreter\_variable\_data\_s::name.

Referenced by agent\_RunChScriptThread(), and agent\_task\_Destroy().

**8.36.2.3 interpreter\_variable\_data\_p interpreter\_variable\_data\_Initialize (struct agent\_s \* agent, const char \* varname)****8.36.2.4 interpreter\_variable\_data\_p interpreter\_variable\_data\_InitializeFromAgent (struct agent\_s \* agent)****8.36.2.5 interpreter\_variable\_data\_p interpreter\_variable\_data\_New (void)**

Definition at line 40 of file agent\_return\_data.c.

References interpreter\_variable\_data\_s::array\_dim, interpreter\_variable\_data\_s::array\_extent, CHECK\_NULL, interpreter\_variable\_data\_s::data, interpreter\_variable\_data\_s::data\_type, interpreter\_variable\_data\_s::name, and interpreter\_variable\_data\_s::size.

Referenced by agent\_xml\_parse\_\_data(), and interpreter\_variable\_data\_Copy().

## 8.37 /home/dko/projects/mobilec/trunk/src/include/libmc.h File Reference

MobileC api header file.

```
#include <embedch.h>
```

### Data Structures

- struct [agency\\_s](#)  
*The agency handle.*
- struct [MCAgencyOptions\\_s](#)  
*User modifiable agency options.*

### Defines

- #define [EXPORTMC](#)
- #define [MC\\_Wait](#)(arg1) MC\_MainLoop(arg1)

### Typedefs

- typedef enum [error\\_code\\_e](#) error\_code\_t
- typedef enum [MC\\_SteerCommand\\_e](#) MC\_SteerCommand\_t
- typedef struct [agency\\_s](#) agency\_t
- typedef [agency\\_t](#) \* [agency\\_p](#)
- typedef [agency\\_p](#) [MCAgency\\_t](#)
- typedef struct [MCAgencyOptions\\_s](#) MCAgencyOptions\_t
- typedef struct [agent\\_s](#) agent\_t
- typedef [agent\\_t](#) \* [MCAgent\\_t](#)
- typedef [agent\\_t](#) \* [agent\\_p](#)

### Enumerations

- enum [error\\_code\\_e](#) {  
[MC\\_SUCCESS](#) = 0, [MC\\_ERR](#), [MC\\_ERR\\_CONNECT](#), [MC\\_ERR\\_PARSE](#),  
[MC\\_ERR\\_EMPTY](#), [MC\\_ERR\\_INVALID](#), [MC\\_ERR\\_INVALID\\_ARGS](#), [MC\\_ERR\\_NOT\\_FOUND](#),  
[MC\\_ERR\\_MEMORY](#), [MC\\_ERR\\_SEND](#), [MC\\_WARN\\_DUPLICATE](#), [MC\\_SUCCESS](#) = 0,  
[ERR](#), [MC\\_ERR\\_CONNECT](#), [MC\\_ERR\\_PARSE](#), [MC\\_ERR\\_EMPTY](#),  
[MC\\_ERR\\_INVALID](#), [MC\\_ERR\\_INVALID\\_ARGS](#), [MC\\_ERR\\_NOT\\_FOUND](#), [MC\\_ERR\\_MEMORY](#),  
[MC\\_ERR\\_SEND](#), [MC\\_WARN\\_DUPLICATE](#) }
- enum [MC\\_ThreadIndex\\_e](#) {  
[MC\\_THREAD\\_DF](#) = 0, [MC\\_THREAD\\_AMS](#), [MC\\_THREAD\\_ACC](#), [MC\\_THREAD\\_CP](#),  
[MC\\_THREAD\\_AGENT](#), [MC\\_THREAD\\_ALL](#) }

*MobileC Module indices.*

- enum `MC_SteerCommand_e` { `MC_RUN` = 0, `MC_SUSPEND`, `MC_RESTART`, `MC_STOP` }  
*Available commands for MC\_Steer.*
- enum `MC_Signal_e` {  
`MC_NO_SIGNAL` = 0x00, `MC_RECV_CONNECTION` = 0x01, `MC_RECV_MESSAGE` = 0x02,  
`MC_RECV_AGENT` = 0x04,  
`MC_RECV_RETURN` = 0x08, `MC_EXEC_AGENT` = 0x0F, `MC_ALL_SIGNALS` = 0x10 }  
*MobileC system signals.*
- enum `MC_AgentType_e` { `MC_NONE` = -1, `MC_REMOTE_AGENT` = 0, `MC_LOCAL_AGENT`,  
`MC_RETURN_AGENT` }
- enum `MC_AgentStatus_e` {  
`MC_NO_STATUS` = -1, `MC_WAIT_CH` = 0, `MC_WAIT_MESSGSEND`, `MC_AGENT_ACTIVE`,  
`MC_AGENT_NEUTRAL`, `MC_AGENT_SUSPENDED`, `MC_WAIT_FINISHED` }  
*An agent's current execution state.*

## Functions

- EXPORTMC `int MC_AclDestroy` (struct `fipa_acl_message_s` \*message)  
*Destroy a FIPA ACL message.*
- EXPORTMC struct `fipa_acl_message_s` \* `MC_AclNew` (void)  
*Allocate a new ACL Message.*
- EXPORTMC `int MC_AclPost` (MC\_Agent\_t agent, struct `fipa_acl_message_s` \*message)  
*Post ACL message to agent.*
- EXPORTMC struct `fipa_acl_message_s` \* `MC_AclReply` (struct `fipa_acl_message_s` \*acl\_message)  
*Reply to an ACL message.*
- EXPORTMC struct `fipa_acl_message_s` \* `MC_AclRetrieve` (MC\_Agent\_t agent)  
*Retrieve an ACL message.*
- EXPORTMC `int MC_AclSend` (MC\_Agency\_t attr, struct `fipa_acl_message_s` \*acl)  
*Send a composed ACL Message.*
- EXPORTMC struct `fipa_acl_message_s` \* `MC_AclWaitRetrieve` (MC\_Agent\_t agent)  
*Wait for and retrieve an ACL message.*
- EXPORTMC `int MC_AclSetPerformative` (struct `fipa_acl_message_s` \*acl, enum `fipa_performative_e` performative)
- EXPORTMC `int MC_AclSetSender` (struct `fipa_acl_message_s` \*acl, const char \*name, const char \*address)
- EXPORTMC `int MC_AclAddReceiver` (struct `fipa_acl_message_s` \*acl, const char \*name, const char \*address)

- EXPORTMC [int MC\\_AclAddReplyTo](#) (struct [fipa\\_acl\\_message\\_s](#) \*acl, const char \*name, const char \*address)
- EXPORTMC [int MC\\_AclSetContent](#) (struct [fipa\\_acl\\_message\\_s](#) \*acl, const char \*content)
- EXPORTMC [int MC\\_AddAgent](#) ([MCAgency\\_t](#) attr, [MCAgent\\_t](#) agent)  
*Add an agent to the agency 'attr'.*
- const void \* [MC\\_AgentVariableRetrieve](#) ([MCAgent\\_t](#) agent, const char \*var\_name, [int](#) task\_num)  
*Retrieve a pointer to a previously saved variable.*
- [int MC\\_AgentVariableSave](#) ([MCAgent\\_t](#) agent, const char \*var\_name)  
*Mark an agent variable for saving.*
- EXPORTMC [int MC\\_BarrierDelete](#) ([MCAgency\\_t](#) attr, [int](#) id)  
*Find and delete an initialized MobileC Barrier.*
- EXPORTMC [int MC\\_BarrierInit](#) ([MCAgency\\_t](#) attr, [int](#) id, [int](#) num\_procs)  
*Initialize a MobileC Barrier.*
- EXPORTMC [int MC\\_ChInitializeOptions](#) ([MCAgency\\_t](#) attr, [ChOptions\\_t](#) \*options)  
*Use custom ChOptions\_t type for internal Ch interpreter.*
- EXPORTMC [int MC\\_CallAgentFunc](#) ([MCAgent\\_t](#) agent, const char \*funcName, void \*returnVal, void \*varg)  
*Calls a function defined in an agent.*
- [MCAgent\\_t MC\\_ComposeAgent](#) (const char \*name, const char \*home, const char \*owner, const char \*code, const char \*return\_var\_name, const char \*server, [int](#) persistent)  
*Compose a new agent dynamically without using a prewritten XML file.*
- EXPORTMC [int MC\\_CondBroadcast](#) ([MCAgency\\_t](#) attr, [int](#) id)  
*Wakes up all agents/threads waiting on a condition variable.*
- EXPORTMC [int MC\\_CondSignal](#) ([MCAgency\\_t](#) attr, [int](#) id)  
*Wakes up at least one thread waiting on a condition variable.*
- EXPORTMC [int MC\\_CondReset](#) ([MCAgency\\_t](#) attr, [int](#) id)  
*Reset a previously signalled MobileC condition variable.*
- EXPORTMC [int MC\\_CondWait](#) ([MCAgency\\_t](#) attr, [int](#) id)  
*Wait on a MobileC synchronization variable.*
- [int MC\\_CopyAgent](#) ([MCAgent\\_t](#) \*agent\_out, const [MCAgent\\_t](#) agent\_in)  
*Performs a deep-copy of an agent structure.*
- EXPORTMC [int MC\\_DeleteAgent](#) ([MCAgent\\_t](#) agent)  
*Stop and remove an agent.*
- EXPORTMC [int MC\\_End](#) ([MCAgency\\_t](#) attr)  
*End an agency.*

- `int MC_DestroyServiceSearchResult` (char \*\*agentName, char \*\*serviceName, int \*agentID, int numResult)  
*Free memory allocated by a Service Search operation.*
- EXPORTMC `MCAgent_t MC_FindAgentByName` (MCAgency\_t attr, const char \*name)  
*Find an agent by its name.*
- EXPORTMC `MCAgent_t MC_FindAgentByID` (MCAgency\_t attr, int ID)  
*Find an agent by its id.*
- EXPORTMC `void * MC_GetAgentExecEngine` (MCAgent\_t agent)  
*Retrieve an agent's Ch interpreter.*
- EXPORTMC `int MC_GetAgentID` (MCAgent\_t agent)  
*Retrieve an agent's id.*
- EXPORTMC `char * MC_GetAgentName` (MCAgent\_t agent)
- EXPORTMC `int MC_GetAgentNumTasks` (MCAgent\_t agent)  
*Retrive the number of tasks an agent has.*
- EXPORTMC `int MC_GetAgentReturnData` (MCAgent\_t agent, int task\_num, void \*\*data, int \*dim, int \*\*extent)  
*Get an agent's return data.*
- EXPORTMC `int MC_GetAgentStatus` (MCAgent\_t agent)  
*Get an agent's current status.*
- EXPORTMC `enum MC_AgentType_e MC_GetAgentType` (MCAgent\_t agent)  
*Get an agent's type.*
- EXPORTMC `char * MC_GetAgentXMLString` (MCAgent\_t agent)  
*Get an agent's xml string.*
- EXPORTMC `int MC_HaltAgency` (MCAgency\_t agency)  
*Halt an agency: Do not process new entries in queues.*
- EXPORTMC `MCAgency_t MC_Initialize` (int port, MCAgencyOptions\_t \*options)  
*Initialize and start a MobileC agency.*
- EXPORTMC `int MC_InitializeAgencyOptions` (struct MCAgencyOptions\_s \*options)  
*Initialize MobileC options.*
- EXPORTMC `int MC_MainLoop` (MCAgency\_t attr)  
*Wait indefinitely.*
- EXPORTMC `int MC_LoadAgentFromFile` (MCAgency\_t attr, const char \*filename)  
*Load an agent from a file into an agency.*
- EXPORTMC `int MC_MutexLock` (MCAgency\_t attr, int id)  
*Locks a MobileC synchronization variable as a mutex.*



- EXPORTMC [int MC\\_MutexUnlock](#) ([MCAgency\\_t](#) attr, [int](#) id)
- EXPORTMC [int MC\\_RegisterService](#) ([MCAgency\\_t](#) agency, [MCAgent\\_t](#) agent, [int](#) agentID, const char \*agentName, char \*\*serviceNames, [int](#) numServices)  
*Register a new service with the Directory Facilitator.*
- EXPORTMC [int MC\\_ResumeAgency](#) ([MCAgency\\_t](#) agency)  
*Resumes a halted agency.*
- EXPORTMC [MCAgent\\_t MC\\_RetrieveAgent](#) ([MCAgency\\_t](#) attr)  
*Retrieves the oldest agent from an agency.*
- EXPORTMC [int MC\\_SemaphorePost](#) ([MCAgency\\_t](#) attr, [int](#) id)  
*Post to a MobileC synchronization variable semaphore.*
- EXPORTMC [int MC\\_SemaphoreWait](#) ([MCAgency\\_t](#) attr, [int](#) id)  
*Decreases a MobileC synchronization variable semaphore count by one.*
- EXPORTMC [int MC\\_SetDefaultAgentStatus](#) ([MCAgency\\_t](#) agency, enum [MC\\_AgentStatus\\_e](#) status)  
*Sets default incoming agent status.*
- EXPORTMC [int MC\\_SetThreadOn](#) ([MCAgencyOptions\\_t](#) \*options, enum [MC\\_ThreadIndex\\_e](#) index)  
*Sets a MobileC thread to "on" status.*
- EXPORTMC [int MC\\_SetThreadsAllOn](#) ([MCAgencyOptions\\_t](#) \*options)  
*Set all Mobile-C threads on.*
- EXPORTMC [int MC\\_SetThreadOff](#) ([MCAgencyOptions\\_t](#) \*options, enum [MC\\_ThreadIndex\\_e](#) index)  
*Sets a MobileC thread to "off" status.*
- EXPORTMC [int MC\\_SetThreadsAllOff](#) ([MCAgencyOptions\\_t](#) \*options)  
*Set all MobileC threads to 'off' status.*
- EXPORTMC [int MC\\_PrintAgentCode](#) ([MCAgent\\_t](#) agent)  
*Prints an agents code to stdout.*
- EXPORTMC char \* [MC\\_RetrieveAgentCode](#) ([MCAgent\\_t](#) agent)  
*Retrieves an agent's Ch code.*
- EXPORTMC [int MC\\_ResetSignal](#) ([MCAgency\\_t](#) attr)  
*Reset a MobileC signal.*
- EXPORTMC [int MC\\_SearchForService](#) ([MCAgency\\_t](#) attr, const char \*searchString, char \*\*\*agentNames, char \*\*\*serviceNames, [int](#) \*\*agentIDs, [int](#) \*numResults)  
*Search the directory facilitator for a service.*
- EXPORTMC [int MC\\_SendAgentMigrationMessage](#) ([MCAgency\\_t](#) attr, const char \*message, const char \*hostname, [int](#) port)

*Sends an agent migration message.*

- EXPORTMC [int MC\\_SendAgentMigrationMessageFile](#) (MCAgency\_t attr, const char \*filename, const char \*hostname, [int port](#))

*Sends an agent migration message.*

- EXPORTMC [int MC\\_SetAgentStatus](#) (MCAgent\_t agent, [int status](#))

*Set an agent's status.*

- EXPORTMC [int MC\\_Steer](#) (MCAgency\_t attr, [int\(\\*funcptr\)\(void \\*data\)](#), void \*arg)

*Set up a steerable algorithm.*

- EXPORTMC [enum MC\\_SteerCommand\\_e MC\\_SteerControl](#) (void)

*The MobileC user-algorithm steering function.*

- EXPORTMC [int MC\\_SyncDelete](#) (MCAgency\_t attr, [int id](#))

*Deletes a previously initialized synchronization variable.*

- EXPORTMC [int MC\\_SyncInit](#) (MCAgency\_t attr, [int id](#))

*Initializes a new MobileC synchronization variable.*

- EXPORTMC [int MC\\_TerminateAgent](#) (MCAgent\_t agent)

*Halt a running agent.*

- EXPORTMC [int MC\\_WaitAgent](#) (MCAgency\_t attr)

*Wait indefinitely.*

- EXPORTMC [MCAgent\\_t MC\\_WaitRetrieveAgent](#) (MCAgency\_t attr)

*Wait and retrieve an agent.*

- EXPORTMC [int MC\\_WaitSignal](#) (MCAgency\_t attr, [int signals](#))

*Wait for a MobileC signal.*

### 8.37.1 Detailed Description

MobileC api header file.

Definition in file [libmc.h](#).

### 8.37.2 Define Documentation

#### 8.37.2.1 #define EXPORTMC

Definition at line 138 of file libmc.h.

#### 8.37.2.2 #define MC\_Wait(arg1) MC\_MainLoop(arg1)

Definition at line 143 of file libmc.h.

### 8.37.3 Typedef Documentation

#### 8.37.3.1 typedef agency\_t\* agency\_p

Definition at line 229 of file libmc.h.

#### 8.37.3.2 typedef struct agency\_s agency\_t

#### 8.37.3.3 typedef agent\_t\* agent\_p

Definition at line 249 of file libmc.h.

#### 8.37.3.4 typedef struct agent\_s agent\_t

Definition at line 247 of file libmc.h.

#### 8.37.3.5 typedef enum error\_code\_e error\_code\_t

#### 8.37.3.6 typedef enum MC\_SteerCommand\_e MC\_SteerCommand\_t

#### 8.37.3.7 typedef agency\_p MCAgency\_t

Definition at line 230 of file libmc.h.

#### 8.37.3.8 typedef struct MCAgencyOptions\_s MCAgencyOptions\_t

#### 8.37.3.9 typedef agent\_t\* MCAgent\_t

Definition at line 248 of file libmc.h.

### 8.37.4 Enumeration Type Documentation

#### 8.37.4.1 enum error\_code\_e

Enumerator:

*MC\_SUCCESS*  
*MC\_ERR*  
*MC\_ERR\_CONNECT*  
*MC\_ERR\_PARSE*  
*MC\_ERR\_EMPTY*  
*MC\_ERR\_INVALID*  
*MC\_ERR\_INVALID\_ARGS*  
*MC\_ERR\_NOT\_FOUND*  
*MC\_ERR\_MEMORY*  
*MC\_ERR\_SEND*  
*MC\_WARN\_DUPLICATE*

*MC\_SUCCESS*  
*ERR*  
*MC\_ERR\_CONNECT*  
*MC\_ERR\_PARSE*  
*MC\_ERR\_EMPTY*  
*MC\_ERR\_INVALID*  
*MC\_ERR\_INVALID\_ARGS*  
*MC\_ERR\_NOT\_FOUND*  
*MC\_ERR\_MEMORY*  
*MC\_ERR\_SEND*  
*MC\_WARN\_DUPLICATE*

Definition at line 157 of file libmc.h.

#### 8.37.4.2 enum MC\_AgentStatus\_e

An agent's current execution state.

**Enumerator:**

*MC\_NO\_STATUS*  
*MC\_WAIT\_CH* Waiting to be started  
*MC\_WAIT\_MESSGSEND* Finished, waiting to migrate  
*MC\_AGENT\_ACTIVE* Running  
*MC\_AGENT\_NEUTRAL* Not running, but do not flush  
*MC\_AGENT\_SUSPENDED* Unused  
*MC\_WAIT\_FINISHED* Finished, waiting to be flushed

Definition at line 257 of file libmc.h.

#### 8.37.4.3 enum MC\_AgentType\_e

**Enumerator:**

*MC\_NONE*  
*MC\_REMOTE\_AGENT*  
*MC\_LOCAL\_AGENT*  
*MC\_RETURN\_AGENT*

Definition at line 252 of file libmc.h.

#### 8.37.4.4 enum MC\_Signal\_e

MobileC system signals.

**Note:**

Each signal is activated after the corresponding action. i.e. The 'MC\_RECV\_MESSAGE' signal is activated after a message is received.

**See also:**

[MC\\_WaitSignal\(\)](#), [MC\\_ResetSignal](#)

**Enumerator:**

*MC\_NO\_SIGNAL* No signal has been received  
*MC\_RECV\_CONNECTION* A Connection has been received  
*MC\_RECV\_MESSAGE* A Message has been received from a connection  
*MC\_RECV\_AGENT* An normal agent has been parsed from a message  
*MC\_RECV\_RETURN* A return agent has been parsed from a message  
*MC\_EXEC\_AGENT* A normal agent has been executed.  
*MC\_ALL\_SIGNALS* Catch any of the above signals

Definition at line 202 of file libmc.h.

#### 8.37.4.5 enum MC\_SteerCommand\_e

Available commands for MC\_Steer.

**Enumerator:**

*MC\_RUN* Continue the algorithm  
*MC\_SUSPEND* Suspend/pause the algorithm  
*MC\_RESTART* Restart the algorithm from the beginning  
*MC\_STOP* Stop the algorithm

Definition at line 187 of file libmc.h.

#### 8.37.4.6 enum MC\_ThreadIndex\_e

MobileC Module indices.

**Enumerator:**

*MC\_THREAD\_DF* Directory Facilitator  
*MC\_THREAD\_AMS* Agent Managment system  
*MC\_THREAD\_ACC* Agency communications  
*MC\_THREAD\_CP* Command Prompt  
*MC\_THREAD\_AGENT* Agent threads  
*MC\_THREAD\_ALL*

Definition at line 175 of file libmc.h.

## 8.37.5 Function Documentation

### 8.37.5.1 EXPORTMC int MC\_AclAddReceiver (struct fipa\_acl\_message\_s \* *acl*, const char \* *name*, const char \* *address*)

Definition at line 229 of file libmc.c.

References fipa\_agent\_identifier\_s::addresses, fipa\_agent\_identifier\_New(), fipa\_agent\_identifier\_set\_New(), fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_url\_New(), fipa\_url\_sequence\_New(), fipa\_agent\_identifier\_s::name, fipa\_url\_sequence\_s::num, fipa\_agent\_identifier\_set\_s::num, fipa\_acl\_message\_s::receiver, fipa\_acl\_message\_s::receiver\_num, fipa\_url\_s::str, and fipa\_url\_sequence\_s::urls.

Referenced by MC\_AclAddReceiver\_chdl().

### 8.37.5.2 EXPORTMC int MC\_AclAddReplyTo (struct fipa\_acl\_message\_s \* *acl*, const char \* *name*, const char \* *address*)

Definition at line 265 of file libmc.c.

References fipa\_agent\_identifier\_s::addresses, fipa\_agent\_identifier\_New(), fipa\_agent\_identifier\_set\_New(), fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_url\_New(), fipa\_url\_sequence\_New(), fipa\_agent\_identifier\_s::name, fipa\_url\_sequence\_s::num, fipa\_agent\_identifier\_set\_s::num, fipa\_acl\_message\_s::reply\_to, fipa\_url\_s::str, and fipa\_url\_sequence\_s::urls.

Referenced by MC\_AclAddReplyTo\_chdl().

### 8.37.5.3 EXPORTMC int MC\_AclDestroy (struct fipa\_acl\_message\_s \* *message*)

Destroy a FIPA ACL message.

#### Parameters:

*message* The ACL message to destroy

#### Returns:

0 on success, error code on failure.

Definition at line 72 of file libmc.c.

References fipa\_acl\_message\_Destroy().

Referenced by MC\_AclDestroy\_chdl().

### 8.37.5.4 EXPORTMC struct fipa\_acl\_message\_s\* MC\_AclNew (void) [read]

Allocate a new ACL Message.

#### Returns:

A newly allocated and empty ACL message.

Definition at line 78 of file libmc.c.

References fipa\_acl\_message\_New().

Referenced by MC\_AclNew\_chdl().

**8.37.5.5 EXPORTMC int MC\_AclPost (MCAgent\_t *agent*, struct fipa\_acl\_message\_s \* *message*)**

Post ACL message to agent.

**Parameters:**

*agent* The agent to post the message to

*message* The message to post

**Returns:**

0 if successful, or error\_code\_t type.

Definition at line 83 of file libmc.c.

References agent\_mailbox\_Post(), and agent\_s::mailbox.

Referenced by MC\_AclPost\_chdl(), and MC\_AclSend().

**8.37.5.6 EXPORTMC struct fipa\_acl\_message\_s\* MC\_AclReply (struct fipa\_acl\_message\_s \* *acl\_message*)** [read]

Reply to an ACL message.

**Parameters:**

*acl\_message* The incoming acl message to reply to

**Returns:**

A newly allocated ACL message

**Note:**

This function simply generates a new ACL message with the 'receiver' field automatically set to the 'sender' field of the incoming message.

Definition at line 89 of file libmc.c.

References fipa\_Reply().

Referenced by MC\_AclReply\_chdl().

**8.37.5.7 EXPORTMC struct fipa\_acl\_message\_s\* MC\_AclRetrieve (MCAgent\_t *agent*)** [read]

Retrieve an ACL message.

**Parameters:**

*agent* Agent to retrieve message from.

**Returns:**

an ACL message struct on success or NULL on failure

Definition at line 95 of file libmc.c.

References agent\_mailbox\_Retrieve(), and agent\_s::mailbox.

Referenced by MC\_AclRetrieve\_chdl().

### 8.37.5.8 EXPORTMC int MC\_AclSend (MCAgency\_t *attr*, struct fipa\_acl\_message\_s \* *acl*)

Send a composed ACL Message.

#### Parameters:

*attr* An initialized and running MobileC agency

*acl* An allocated and fully composed ACL message.

#### Returns:

0 if successful, error code on failure.

Definition at line 101 of file libmc.c.

References fipa\_agent\_identifier\_s::addresses, mtp\_http\_s::content, mtp\_http\_content\_s::content\_type, mtp\_http\_content\_s::data, dynstring\_Destroy(), FIPA\_ACL, fipa\_acl\_Compose(), fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_envelope\_Compose(), mtp\_http\_s::host, http\_to\_hostport(), MC\_AclPost(), MC\_FindAgentByName(), dynstring\_s::message, message\_Destroy(), mtp\_http\_s::message\_parts, message\_Send(), message\_s::message\_type, mtp\_http\_CreateMessage(), mtp\_http\_Destroy(), mtp\_http\_New(), fipa\_agent\_identifier\_s::name, fipa\_url\_sequence\_s::num, fipa\_agent\_identifier\_set\_s::num, port, fipa\_acl\_message\_s::receiver, fipa\_url\_s::str, mtp\_http\_s::target, message\_s::target, and fipa\_url\_sequence\_s::urls.

Referenced by MC\_AclSend\_chdl().

### 8.37.5.9 EXPORTMC int MC\_AclSetContent (struct fipa\_acl\_message\_s \* *acl*, const char \* *content*)

Definition at line 300 of file libmc.c.

References fipa\_string\_s::content, fipa\_acl\_message\_s::content, fipa\_string\_Destroy(), and fipa\_string\_New().

Referenced by MC\_AclSetContent\_chdl().

### 8.37.5.10 EXPORTMC int MC\_AclSetPerformative (struct fipa\_acl\_message\_s \* *acl*, enum fipa\_performative\_e *performative*)

Definition at line 198 of file libmc.c.

References fipa\_acl\_message\_s::performative.

Referenced by MC\_AclSetPerformative\_chdl().

### 8.37.5.11 EXPORTMC int MC\_AclSetSender (struct fipa\_acl\_message\_s \* *acl*, const char \* *name*, const char \* *address*)

Definition at line 206 of file libmc.c.

References fipa\_agent\_identifier\_s::addresses, fipa\_agent\_identifier\_Destroy(), fipa\_agent\_identifier\_New(), fipa\_url\_New(), fipa\_url\_sequence\_New(), fipa\_agent\_identifier\_s::name, fipa\_url\_sequence\_s::num, fipa\_acl\_message\_s::sender, fipa\_url\_s::str, and fipa\_url\_sequence\_s::urls.

Referenced by MC\_AclSetSender\_chdl().



### 8.37.5.12 EXPORTMC struct fipa\_acl\_message\_s\* MC\_AclWaitRetrieve (MCAgent\_t *agent*) [read]

Wait for and retrieve an ACL message.

#### Parameters:

*agent* Agent to retrieve message from.

#### Returns:

an ACL message struct on success or NULL on failure

Definition at line 191 of file libmc.c.

References agent\_mailbox\_WaitRetrieve(), and agent\_s::mailbox.

Referenced by MC\_AclWaitRetrieve\_chdl().

### 8.37.5.13 EXPORTMC int MC\_AddAgent (MCAgency\_t *attr*, MCAgent\_t *agent*)

Add an agent to the agency 'attr'.

#### Parameters:

*attr* a MobileC agency

*agent* An initialized MobileC agent

#### Returns:

0 if successful, or error\_code\_t type

Definition at line 317 of file libmc.c.

References mc\_platform\_s::agent\_queue, mc\_platform\_s::ams, COND\_SIGNAL, agency\_s::mc\_platform, agent\_s::mc\_platform, MUTEX\_LOCK, and MUTEX\_UNLOCK.

Referenced by MC\_AddAgent\_chdl().

### 8.37.5.14 const void\* MC\_AgentVariableRetrieve (MCAgent\_t *agent*, const char \* *var\_name*, int *task\_num*)

Retrieve a pointer to a previously saved variable.

#### Parameters:

*agent* A MobileC agent.

*var\_name* The name of the saved variable that has previously been saved.

*task\_num* The previous completed task from which to retrieve the saved variable.

#### Returns:

A pointer to the data on success or NULL on failure.

### 8.37.6 Examples

The following example demonstrates usage of [MC\\_AgentVariableRetrieve\(\)](#) from agent space.

Definition at line 330 of file libmc.c.

References `agent_task_s::agent_variable_list`, `interpreter_variable_data_s::data`, `agent_s::datastate`, `agent_datastate_s::task_progress`, and `agent_datastate_s::tasks`.

Referenced by `MC_AgentVariableRetrieve_chdl()`.

#### 8.37.6.1 `int MC_AgentVariableSave (MCAgent_t agent, const char * var_name)`

Mark an agent variable for saving.

##### Parameters:

*agent* A MobileC agent.

*var\_name* The name of the variable to mark for saving.

##### Returns:

0 on success, non-zero on failure.

##### See also:

`test1.xml`

### 8.37.7 Examples

See `agent_saved_variables_example/test1.xml` for an example of usage of this api function.

### 8.37.8 Examples

The following example demonstrates usage of [MC\\_AgentVariableSave\(\)](#) from agent space.

Definition at line 348 of file libmc.c.

References `agent_s::datastate`, `MC_ERR_MEMORY`, `agent_task_s::num_saved_variables`, `agent_task_s::saved_variables`, `agent_datastate_s::task_progress`, and `agent_datastate_s::tasks`.

Referenced by `MC_AgentVariableSave_chdl()`.

#### 8.37.8.1 `EXPORTMC int MC_BarrierDelete (MCAgency_t attr, int id)`

Find and delete an initialized MobileC Barrier.

##### Parameters:

*attr* A running MobileC agency

*id* The id of the barrier node to delete

**Returns:**

returns 0 on success, error if the node is not found or other failure.

Definition at line 409 of file libmc.c.

References mc\_platform\_s::barrier\_queue, barrier\_queue\_Delete(), and agency\_s::mc\_platform.

Referenced by MC\_BarrierDelete\_chdl().

### 8.37.8.2 EXPORTMC int MC\_BarrierInit (MCAgency\_t attr, int id, int num\_procs)

Initialize a MobileC Barrier.

**Parameters:**

*attr* A running MobileC agency

*id* The requested barrier id

*num\_procs* The number of agents/threads/processes that will wait on the barrier

**Returns:**

The allocated barrier id. May differ from the requested id if it is already in use.

## 8.37.9 Examples

The following example demonstrates an agent which sets up an MC\_Barrier.

Definition at line 395 of file libmc.c.

References barrier\_node\_Initialize(), mc\_platform\_s::barrier\_queue, barrier\_queue\_Add(), barrier\_queue\_Get(), MC\_ERR, agency\_s::mc\_platform, MC\_SUCCESS, and node.

Referenced by MC\_BarrierInit\_chdl().

### 8.37.9.1 EXPORTMC int MC\_CallAgentFunc (MCAgent\_t agent, const char \*funcName, void \*returnVal, void \*varg)

Calls a function defined in an agent.

**Parameters:**

*agent* An initialized and executed MobileC agent

*funcName* The name of the function to call

*returnVal* The agent function's return value

*varg* The agent functions argument

**Note:**

The agent function must be of the form 'void\* func(void\* arg);'

**Returns:**

0 if successful, error\_code\_t type on failure

### 8.37.10 Example

Definition at line 415 of file libmc.c.

References agent\_s::agent\_interp, MUTEX\_LOCK, MUTEX\_UNLOCK, and agent\_s::run\_lock.

Referenced by MC\_CallAgentFunc\_chdl().

#### 8.37.10.1 EXPORTMC int MC\_ChInitializeOptions (MCAgency\_t *attr*, ChOptions\_t \* *options*)

Use custom ChOptions\_t type for internal Ch interpreter.

##### Parameters:

*attr* A running MobileC agency

*options* Initialized Ch options structure

##### Returns:

0 on success, error\_code\_t type on failure

Definition at line 433 of file libmc.c.

References mc\_platform\_s::interp\_options, and agency\_s::mc\_platform.

#### 8.37.10.2 MCAgent\_t MC\_ComposeAgent (const char \* *name*, const char \* *home*, const char \* *owner*, const char \* *code*, const char \* *return\_var\_name*, const char \* *server*, int *persistent*)

Compose a new agent dynamically without using a prewritten XML file.

##### Parameters:

*name* The desired name of the new agent.

*home* The home of the new agent.

*owner* The owner of the new agent.

*code* The agent code

*return\_var\_name* The name of the agent's return variable. Set to "no-return" if no return variable is desired.

*server* The target destination of the agent.

*persistant* A flag indicating whether or not the agent should be persistent. A value of '1' indicates persistence, while a value of '0' indicates default non-persistent behaviour.

##### Returns:

This function returns a valid MCAgent\_t structure on success or NULL on failure.

Definition at line 451 of file libmc.c.

References agent\_datastate\_s::agent\_code, agent\_datastate\_s::agent\_code\_ids, agent\_datastate\_s::agent\_codes, agent\_datastate\_New(), agent\_New(), agent\_s::agent\_status, agent\_task\_New(), agent\_s::agent\_type, agent\_s::datastate, agent\_s::home, MC\_LOCAL\_AGENT, MC\_WAIT\_MESSGSEND, agent\_s::name, agent\_datastate\_s::number\_of\_tasks, agent\_s::orphan, agent\_s::owner, agent\_datastate\_s::persistent, agent\_task\_s::server\_name, agent\_datastate\_s::tasks, and agent\_task\_s::var\_name.

Referenced by MC\_ComposeAgent\_chdl().

### 8.37.10.3 EXPORTMC int MC\_CondBroadcast (MCAgency\_t attr, int id)

Wakes up all agents/threads waiting on a condition variable.

#### Parameters:

*attr* A MobileC agency

*id* Synchronization variable id to broadcast to

#### See also:

[MC\\_SyncInit\(\)](#), [MC\\_CondSignal\(\)](#)

#### Returns:

0 on success, error\_code\_t type on failure

Definition at line 521 of file libmc.c.

References syncListNode\_s::cond, COND\_BROADCAST, syncListNode\_s::lock, MC\_ERR\_NOT\_FOUND, agency\_s::mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, syncListNode\_s::signalled, mc\_platform\_s::syncList, and syncListFind().

Referenced by MC\_CondBroadcast\_chdl().

### 8.37.10.4 EXPORTMC int MC\_CondReset (MCAgency\_t attr, int id)

Reset a previously signalled MobileC condition variable.

#### Parameters:

*attr* A MobileC Agency

*id* The synchronization variable id to reset

#### See also:

[MC\\_SyncInit\(\)](#)

#### Returns:

0 on success, error\_code\_t type on failure

Definition at line 573 of file libmc.c.

References syncListNode\_s::lock, MC\_ERR\_NOT\_FOUND, agency\_s::mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, syncListNode\_s::signalled, mc\_platform\_s::syncList, and syncListFind().

Referenced by MC\_CondReset\_chdl().

### 8.37.10.5 EXPORTMC int MC\_CondSignal (MCAgency\_t *attr*, int *id*)

Wakes up at least one thread waiting on a condition variable.

**Parameters:**

*attr* A MobileC agency  
*id* synchronization variable id

**See also:**

[MC\\_SyncInit\(\)](#)

**Returns:**

0 on success, error\_code\_t type on failure

### 8.37.11 Example

The following example demonstrates the agent-space version of the function, which is nearly identical to the binary space api function.

Definition at line 536 of file libmc.c.

References syncListNode\_s::cond, COND\_SIGNAL, syncListNode\_s::lock, MC\_ERR\_NOT\_FOUND, agency\_s::mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, syncListNode\_s::signalled, mc\_platform\_s::syncList, and syncListFind().

Referenced by MC\_CondSignal\_chdl().

#### 8.37.11.1 EXPORTMC int MC\_CondWait (MCAgency\_t *attr*, int *id*)

Wait on a MobileC synchronization variable.

**Parameters:**

*attr* A MobileC agency  
*id* a synchronization variable id

**See also:**

[MC\\_SyncInit\(\)](#)

**Returns:**

0 on success, error\_code\_t type on failure

### 8.37.12 Example

The following example demonstrates the agent-space version of this function.

Definition at line 551 of file libmc.c.

References syncListNode\_s::cond, COND\_WAIT, syncListNode\_s::lock, MC\_ERR\_NOT\_FOUND, agency\_s::mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, syncListNode\_s::signalled, mc\_platform\_s::syncList, and syncListFind().

Referenced by MC\_CondWait\_chdl().

#### 8.37.12.1 int MC\_CopyAgent (MCAgent\_t \* *agent\_out*, const MCAgent\_t *agent\_in*)

Performs a deep-copy of an agent structure.

##### Parameters:

*agent\_out* A pointer to the agent to copy to.

*agent\_in* The agent to copy

##### Returns:

0 on success, error\_code\_t type on failure.

Definition at line 591 of file libmc.c.

References agent\_Copy(), and MC\_SUCCESS.

#### 8.37.12.2 EXPORTMC int MC\_DeleteAgent (MCAgent\_t *agent*)

Stop and remove an agent.

##### Parameters:

*agent* An agent in any state (running, waiting, etc)

##### Returns:

0 on success, error\_code\_t type on failure

Definition at line 598 of file libmc.c.

References CHECK\_NULL, MC\_ERR\_INVALID, MC\_SetAgentStatus(), MC\_SUCCESS, MC\_TerminateAgent(), and MC\_WAIT\_FINISHED.

Referenced by MC\_DeleteAgent\_chdl().

#### 8.37.12.3 int MC\_DestroyServiceSearchResult (char \*\* *agentName*, char \*\* *serviceName*, int \* *agentID*, int *numResult*)

Free memory allocated by a Service Search operation.

##### Parameters:

*agentName* agent names returned by a search operation.

*serviceName* service names return by a search operation.

*agentID* list of agent id's returned by a search operation.

***numResult*** The number of hits returned by a search operation.

**Returns:**

0 on success, error code on failure.

Definition at line 612 of file libmc.c.

Referenced by MC\_DestroyServiceSearchResult\_chdl().

#### 8.37.12.4 EXPORTMC int MC\_End (MCAgency\_t *attr*)

End an agency.

**Parameters:**

*attr* A running agency

**Returns:**

0 on success, error\_code\_t type on failure

### 8.37.13 Example

Definition at line 661 of file libmc.c.

References mc\_platform\_s::acc, mc\_platform\_s::ams, mc\_platform\_s::cmd\_prompt, COND\_SIGNAL, mc\_platform\_s::connection\_queue, mc\_platform\_s::df, GET\_THREAD\_MODE, agency\_s::hostName, agency\_s::mc\_platform, mc\_platform\_Destroy(), MC\_THREAD\_ACC, MC\_THREAD\_AMS, MC\_THREAD\_CP, MC\_THREAD\_DF, MUTEX\_LOCK, MUTEX\_UNLOCK, mc\_platform\_s::quit, mc\_platform\_s::quit\_lock, cmd\_prompt\_s::thread, THREAD\_CANCEL, THREAD\_JOIN, and agency\_s::threads.

Referenced by MC\_End\_chdl().

#### 8.37.13.1 EXPORTMC MCAgent\_t MC\_FindAgentByID (MCAgency\_t *attr*, int *ID*)

Find an agent by its id.

**Parameters:**

*attr* the agency to search

*ID* the id to search for

**Returns:**

a valid agent on success, NULL on failure

Definition at line 727 of file libmc.c.

References mc\_platform\_s::agent\_queue, and agency\_s::mc\_platform.

Referenced by MC\_FindAgentByID\_chdl().



### 8.37.13.2 EXPORTMC MCAgent\_t MC\_FindAgentByName (MCAgency\_t *attr*, const char \* *name*)

Find an agent by its name.

#### Parameters:

*attr* a running agency  
*name* name to search for

#### Returns:

a valid agent on success or NULL on failure

### 8.37.14 Example

Definition at line 714 of file libmc.c.

References mc\_platform\_s::agent\_queue, and agency\_s::mc\_platform.

Referenced by MC\_AclSend(), and MC\_FindAgentByName\_chdl().

### 8.37.14.1 EXPORTMC void\* MC\_GetAgentExecEngine (MCAgent\_t *agent*)

Retrieve an agent's Ch interpreter.

#### Parameters:

*agent* a valid agent

#### Returns:

a Ch interpreter of type 'ChInterp\_t' on success, or NULL on failure.

Definition at line 779 of file libmc.c.

References agent\_s::agent\_interp.

### 8.37.14.2 EXPORTMC int MC\_GetAgentID (MCAgent\_t *agent*)

Retrieve an agent's id.

Definition at line 785 of file libmc.c.

References agent\_s::id.

### 8.37.14.3 EXPORTMC char\* MC\_GetAgentName (MCAgent\_t *agent*)

Definition at line 793 of file libmc.c.

References agent\_s::lock, MUTEX\_LOCK, MUTEX\_UNLOCK, and agent\_s::name.

#### 8.37.14.4 EXPORTMC int MC\_GetAgentNumTasks (MCAgent\_t agent)

Retrive the number of tasks an agent has.

#### 8.37.15 Example

Definition at line 874 of file libmc.c.

References agent\_s::datastate, and agent\_datastate\_s::number\_of\_tasks.

#### 8.37.15.1 EXPORTMC int MC\_GetAgentReturnData (MCAgent\_t agent, int task\_num, void \*\* data, int \* dim, int \*\* extent)

Get an agent's return data.

##### Parameters:

*agent* a valid agent

*task\_num* the task for which to retrieve the return data. The task must already be completed.

*data* the return data. May be multi dimensional array.

*dim* the number of dimensions of the return array.

*extent* the extent of each one of the array dimensions.

#### 8.37.16 Example

This file demonstrates the retrieval of agent return data from an agent

This is the agent which gets the data

Definition at line 811 of file libmc.c.

References agent\_task\_s::agent\_return\_data, interpreter\_variable\_data\_s::array\_dim, interpreter\_variable\_data\_s::array\_extent, CH\_DATATYPE\_SIZE, interpreter\_variable\_data\_s::data\_type, agent\_s::datastate, agent\_datastate\_s::number\_of\_tasks, size, and agent\_datastate\_s::tasks.

#### 8.37.16.1 EXPORTMC int MC\_GetAgentStatus (MCAgent\_t agent)

Get an agent's current status.

##### Returns:

returns type 'enum MC\_AgentStatus\_e'

Definition at line 757 of file libmc.c.

References agent\_s::agent\_status, agent\_s::lock, MUTEX\_LOCK, and MUTEX\_UNLOCK.

**8.37.16.2 EXPORTMC enum MC\_AgentType\_e MC\_GetAgentType (MCAgent\_t *agent*)**

Get an agent's type.

**Returns:**

returns type 'enum MC\_AgentType\_e'

Definition at line 880 of file libmc.c.

References agent\_s::agent\_type.

**8.37.16.3 EXPORTMC char\* MC\_GetAgentXMLString (MCAgent\_t *agent*)**

Get an agent's xml string.

**Returns:**

a malloc'd character string containing the agent's xml code

Definition at line 767 of file libmc.c.

References agent\_s::datastate, mxmlSaveAllocString(), and agent\_datastate\_s::xml\_agent\_root.

**8.37.16.4 EXPORTMC int MC\_HaltAgency (MCAgency\_t *agency*)**

Halt an agency: Do not process new entries in queues.

**Parameters:**

*agency* A handle to a running MobileC agency.

**Returns:**

0 on success, non-zero on failure.

Definition at line 926 of file libmc.c.

References mc\_platform\_s::giant, mc\_platform\_s::giant\_lock, agency\_s::mc\_platform, MUTEX\_LOCK, and MUTEX\_UNLOCK.

Referenced by MC\_GetAllAgents(), and MC\_HaltAgency\_chdl().

**8.37.16.5 EXPORTMC MCAgency\_t MC\_Initialize (int *port*, MCAgencyOptions\_t \* *options*)**

Initialize and start a MobileC agency.

**Parameters:**

*port* the TCP port the agency should bind to

*options* initialized MobileC options or NULL for default options

**Returns:**

a handle to a running MobileC agency or NULL on failure

### 8.37.17 Example

Definition at line 935 of file libmc.c.

References CHECK\_NULL, agency\_s::client, MCAgencyOptions\_s::default\_agent\_status, agency\_s::default\_agentstatus, MCAgencyOptions\_s::enable\_security, agency\_s::enable\_security, HOST\_NAME\_MAX, agency\_s::hostName, MC\_InitializeAgencyOptions(), agency\_s::mc\_platform, mc\_platform\_Initialize(), MC\_THREAD\_ALL, agency\_s::portno, agency\_s::server, MCAgencyOptions\_s::stack\_size, agency\_s::stack\_size, MCAgencyOptions\_s::threads, and agency\_s::threads.

#### 8.37.17.1 EXPORTMC int MC\_InitializeAgencyOptions (struct MCAgencyOptions\_s \* *options*)

Initialize MobileC options.

##### Parameters:

*options* options to initialize.

##### Returns:

0 on success, error\_code\_t on failure

##### Note:

MobileC options should be initialized with this function before any of its members are modified.

### 8.37.18 Example

Definition at line 990 of file libmc.c.

References MCAgencyOptions\_s::default\_agent\_status, MCAgencyOptions\_s::enable\_security, MC\_THREAD\_ALL, MC\_WAIT\_CH, MCAgencyOptions\_s::modified, MCAgencyOptions\_s::stack\_size, and MCAgencyOptions\_s::threads.

Referenced by MC\_Initialize().

#### 8.37.18.1 EXPORTMC int MC\_LoadAgentFromFile (MCAgency\_t *attr*, const char \* *filename*)

Load an agent from a file into an agency.

##### Parameters:

*agency* A valid and running Mobile-C agency

*filename* Filename containing the agent to load

##### Returns:

0 on success, non-zero on failure.

Definition at line 1007 of file libmc.c.

References agency\_s::mc\_platform, message\_Destroy(), message\_InitializeFromString(), message\_New(), mc\_platform\_s::message\_queue, MXML\_DESCEND, mxmlFindElement(), mxmlLoadString(), message\_s::to\_address, message\_s::xml\_payload, and message\_s::xml\_root.

### 8.37.18.2 EXPORTMC int MC\_MainLoop (MCAgency\_t *attr*)

Wait indefinitely.

**Note:**

This function is intended to block the calling thread forever.

Definition at line 1632 of file libmc.c.

References mc\_platform\_s::ams, and agency\_s::mc\_platform.

### 8.37.18.3 EXPORTMC int MC\_MutexLock (MCAgency\_t *attr*, int *id*)

Locks a MobileC synchronization variable as a mutex.

**Parameters:**

*attr* a MobileC agency handle

*id* the synchronization variable id to lock

**Returns:**

0 on success, error\_code\_t type on failure

## 8.37.19 Example

Consider the following agents, which use the agent-space version of this api function. Note that the 'sleep' agent is sent first, followed by the 'wake' agent.

Definition at line 1077 of file libmc.c.

References syncListNode\_s::lock, agency\_s::mc\_platform, MUTEX\_LOCK, mc\_platform\_s::syncList, and syncListFind().

### 8.37.19.1 EXPORTMC int MC\_MutexUnlock (MCAgency\_t *attr*, int *id*)

Definition at line 1089 of file libmc.c.

References syncListNode\_s::lock, agency\_s::mc\_platform, MUTEX\_UNLOCK, mc\_platform\_s::syncList, and syncListFind().

### 8.37.19.2 EXPORTMC int MC\_PrintAgentCode (MCAgent\_t *agent*)

Prints an agents code to stdout.

**Returns:**

0 on success, error\_code\_t on failure

Definition at line 1101 of file libmc.c.

References agent\_datastate\_s::agent\_code, agent\_s::datastate, agent\_s::lock, MUTEX\_LOCK, MUTEX\_UNLOCK, agent\_datastate\_s::number\_of\_tasks, and agent\_datastate\_s::task\_progress.

### 8.37.19.3 EXPORTMC int MC\_RegisterService (MCAgency\_t *agency*, MCAgent\_t *agent*, int *agentID*, const char \* *agentName*, char \*\* *serviceNames*, int *numServices*)

Register a new service with the Directory Facilitator.

#### Parameters:

*agency* a MobileC agency handle  
*agent* (OPTIONAL: See note) a MobileC agent  
*agentID* (OPTIONAL: See note) a MobileC agent id  
*agentName* (OPTIONAL: See note) a MobileC agent name  
*serviceNames* an array of character strings of service names  
*numServices* the number of services described in 'serviceNames'

#### Returns:

0 on success, error\_code\_t type on failure

#### Note:

Three of the input arguments are optional. The function expects as input the arguments 'agent XOR (agentID AND agentName)'.

## 8.37.20 Example

Definition at line 1117 of file libmc.c.

References CHECK\_NULL, mc\_platform\_s::df, df\_AddRequest(), df\_request\_list\_node\_New(), agent\_s::id, MC\_ERR\_INVALID\_ARGS, MC\_ERR\_MEMORY, agency\_s::mc\_platform, MUTEX\_INIT, MUTEX\_T, and agent\_s::name.

### 8.37.20.1 EXPORTMC int MC\_ResetSignal (MCAgency\_t *attr*)

Reset a MobileC signal.

#### Returns:

0 on success, error\_code\_t on failure

#### See also:

[MC\\_WaitSignal\(\)](#)

Definition at line 1251 of file libmc.c.

References COND\_SIGNAL, mc\_platform\_s::giant, mc\_platform\_s::giant\_cond, mc\_platform\_s::giant\_lock, MC\_NO\_SIGNAL, agency\_s::mc\_platform, mc\_platform\_s::MC\_signal, MUTEX\_LOCK, and MUTEX\_UNLOCK.

### 8.37.20.2 EXPORTMC int MC\_ResumeAgency (MCAgency\_t *agency*)

Resumes a halted agency.

#### Parameters:

*agency* An agency previously halted with the [MC\\_HaltAgency\(\)](#) function.

#### Returns:

0 on success, non-zero on failure.

Definition at line 1196 of file libmc.c.

References `mc_platform_s::giant`, `mc_platform_s::giant_lock`, `agency_s::mc_platform`, `MUTEX_LOCK`, and `MUTEX_UNLOCK`.

Referenced by `MC_GetAllAgents()`.

### 8.37.20.3 EXPORTMC MCAgent\_t MC\_RetrieveAgent (MCAgency\_t *attr*)

Retrieves the oldest agent from an agency.

#### Returns:

a valid agent or NULL on failure

Definition at line 1205 of file libmc.c.

References `mc_platform_s::agent_queue`, `agent_s::agent_status`, `ListSearch()`, `MC_AGENT_NEUTRAL`, `agency_s::mc_platform`, `MUTEX_LOCK`, and `MUTEX_UNLOCK`.

### 8.37.20.4 EXPORTMC char\* MC\_RetrieveAgentCode (MCAgent\_t *agent*)

Retrieves an agent's Ch code.

#### Returns:

a malloc'd character string on success, NULL on failure

Definition at line 1235 of file libmc.c.

References `agent_datastate_s::agent_code`, `agent_s::datastate`, `agent_s::lock`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, and `agent_datastate_s::task_progress`.

### 8.37.20.5 EXPORTMC int MC\_SearchForService (MCAgency\_t *attr*, const char \* *searchString*, char \*\*\* *agentNames*, char \*\*\* *serviceNames*, int \*\* *agentIDs*, int \* *numResults*)

Search the directory facilitator for a service.

#### Returns:

0 on success, `error_code_t` on failure

**Parameters:**

*attr* (input) a MobileC agency handle  
*searchString* (input) substring to search services for  
*agentNames* (return) array of agent names with matching services  
*serviceNames* (return) array of matching service names  
*agentIDs* (return) array of matching agent IDs  
*numResults* (return) number of matching results

**8.37.21 Example**

Definition at line 1262 of file libmc.c.

References CHECK\_NULL, COND\_SLEEP\_ACTION, mc\_platform\_s::df, df\_AddRequest(), df\_request\_list\_node\_Destroy(), df\_request\_list\_node\_New(), df\_request\_search\_Destroy(), df\_request\_search\_New(), MC\_ERR\_MEMORY, agency\_s::mc\_platform, MC\_SUCCESS, and search.

**8.37.21.1 EXPORTMC int MC\_SemaphorePost (MCAgency\_t attr, int id)**

Post to a MobileC synchronization variable semaphore.

**Parameters:**

*attr* a MobileC agency handle  
*id* the synchronization variable id to post to

**Returns:**

0 on success, error\_code\_t type on failure

**8.37.22 Example**

Definition at line 1312 of file libmc.c.

References agency\_s::mc\_platform, syncListNode\_s::sem, SEMAPHORE\_POST, mc\_platform\_s::syncList, and syncListFind().

**8.37.22.1 EXPORTMC int MC\_SemaphoreWait (MCAgency\_t attr, int id)**

Decreases a MobileC synchronization variable semaphore count by one.

**Parameters:**

*attr* a MobileC agency handle  
*id* synchronization variable id to wait on

**Returns:**

0 on MC\_SUCCESS, error\_code\_t type of failure



**Note:**

If the semaphore count is already zero, this function will block until another thread posts to the semaphore.

**8.37.23 Example**

Definition at line 1324 of file libmc.c.

References `agency_s::mc_platform`, `syncListNode_s::sem`, `SEMAPHORE_WAIT`, `mc_platform_s::syncList`, and `syncListFind()`.

**8.37.23.1 EXPORTMC int MC\_SendAgentMigrationMessage (MCAgency\_t *attr*, const char \* *message*, const char \* *hostname*, int *port*)**

Sends an agent migration message.

**Parameters:**

*attr* a MobileC agency handle  
*message* a valid MobileC xml agent migration message  
*hostname* host to send the message to  
*port* port to send the message to

Definition at line 1346 of file libmc.c.

References `MC_ERR`, `agency_s::mc_platform`, `message_Destroy()`, `message_InitializeFromString()`, `message_New()`, and `mc_platform_s::message_queue`.

**8.37.23.2 EXPORTMC int MC\_SendAgentMigrationMessageFile (MCAgency\_t *attr*, const char \* *filename*, const char \* *hostname*, int *port*)**

Sends an agent migration message.

**Parameters:**

*attr* a MobileC agency handle  
*filename* file containing a valid MobileC xml agent migration message  
*hostname* hostname to send the agent to  
*port* port to send the agent to

Definition at line 1377 of file libmc.c.

References `agency_s::mc_platform`, `message_Destroy()`, `message_InitializeFromString()`, `message_New()`, and `mc_platform_s::message_queue`.

**8.37.23.3 EXPORTMC int MC\_SetAgentStatus (MCAgent\_t *agent*, int *status*)**

Set an agent's status.

**Parameters:**

*agent* a MobileC agent

*status* agent status of type 'enum MC\_AgentStatus\_e'

**Returns:**

0 on success, or error\_code\_t on failure

Definition at line 1463 of file libmc.c.

References agent\_s::agent\_status, mc\_platform\_s::ams, COND\_SIGNAL, agent\_s::lock, agent\_s::mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, and agent\_s::orphan.

Referenced by MC\_DeleteAgent().

#### 8.37.23.4 EXPORTMC int MC\_SetDefaultAgentStatus (MCAgency\_t *agency*, enum MC\_AgentStatus\_e *status*)

Sets default incoming agent status.

**Parameters:**

*agency* a MobileC agency handle

*status* the status to set all incoming agents

**Returns:**

0 on success, error\_type\_t on failure

**Note:**

using this function will override any status the incoming agent attempts to set for itself.

Definition at line 1478 of file libmc.c.

References mc\_platform\_s::default\_agentstatus, and agency\_s::mc\_platform.

#### 8.37.23.5 EXPORTMC int MC\_SetThreadOff (MCAgencyOptions\_t \* *options*, enum MC\_ThreadIndex\_e *index*)

Sets a MobileC thread to "off" status.

**Parameters:**

*options* MobileC options previously initialized with [MC\\_InitializeAgencyOptions\(\)](#)

*index* the thread to set

**Returns:**

0 on success, error\_code\_t on failure

**Note:**

This function must be called before [MC\\_Initialize\(\)](#). Once an agency is started with MC\_Initialize, the MC\_SetThread functions will have no effect.

Definition at line 1505 of file libmc.c.

References SET\_THREAD\_OFF, and MCAgencyOptions\_s::threads.

### 8.37.23.6 EXPORTMC int MC\_SetThreadOn (MCAgencyOptions\_t \* *options*, enum MC\_ThreadIndex\_e *index*)

Sets a MobileC thread to "on" status.

#### Parameters:

*options* MobileC options previously initialized with [MC\\_InitializeAgencyOptions\(\)](#)

*index* the thread to set

#### Returns:

0 on success, error\_code\_t on failure

#### Note:

This function must be called before [MC\\_Initialize\(\)](#). Once an agency is started with MC\_Initialize, the MC\_SetThread functions will have no effect.

Definition at line 1488 of file libmc.c.

References SET\_THREAD\_ON, and MCAgencyOptions\_s::threads.

### 8.37.23.7 EXPORTMC int MC\_SetThreadsAllOff (MCAgencyOptions\_t \* *options*)

Set all MobileC threads to 'off' status.

#### Parameters:

*options* a MobileC options structure initialized with with the [MC\\_InitializeAgencyOptions\(\)](#) function.

#### Returns:

0 on success, error code on failure.

Definition at line 1512 of file libmc.c.

References MC\_THREAD\_ALL, SET\_THREAD\_OFF, and MCAgencyOptions\_s::threads.

### 8.37.23.8 EXPORTMC int MC\_SetThreadsAllOn (MCAgencyOptions\_t \* *options*)

Set all Mobile-C threads on.

#### Parameters:

*options* MobileC options structure, initialized with [MC\\_InitializeAgencyOptions\(\)](#)

#### Returns:

0 on success, error code on failure.

Definition at line 1495 of file libmc.c.

References MC\_THREAD\_ALL, SET\_THREAD\_ON, and MCAgencyOptions\_s::threads.

**8.37.23.9 EXPORTMC int MC\_Steer (MCAgency\_t *attr*, int(\*) (void \**data*) *funcptr*, void \* *arg*)**

Set up a steerable algorithm.

**Parameters:**

*attr* a MobileC agency handle  
*funcptr* a function pointer to the algorithm  
*arg* an argument for the algorithm function

**Returns:**

0 on success, error\_code\_t on failure

**Note:**

The algorithm function must contain a call to MC\_SteerControl in order for the algorithm to be steerable.

**8.37.24 Example**

Definition at line 1522 of file libmc.c.

References agency\_s::mc\_platform, MC\_RESTART, MC\_RUN, mc\_platform\_s::MC\_steer\_command, mc\_platform\_s::MC\_steer\_lock, MUTEX\_LOCK, and MUTEX\_UNLOCK.

**8.37.24.1 EXPORTMC enum MC\_SteerCommand\_e MC\_SteerControl (void)**

The MobileC user-algorithm steering function.

**Returns:**

The current steering command

**Note:**

This function belongs inside a user's steerable algorithm.

**See also:**

[MC\\_Steer\(\)](#)

Definition at line 1541 of file libmc.c.

References COND\_WAIT, mc\_platform\_s::MC\_steer\_command, mc\_platform\_s::MC\_steer\_cond, mc\_platform\_s::MC\_steer\_lock, MC\_SUSPEND, MUTEX\_LOCK, and MUTEX\_UNLOCK.

**8.37.24.2 EXPORTMC int MC\_SyncDelete (MCAgency\_t *attr*, int *id*)**

Deletes a previously initialized synchronization variable.

**Parameters:**

*attr* a MobileC agency handle

*id* the sync variable id to delete

**Returns:**

0 on success, or `error_code_t` on failure

Definition at line 1557 of file `libmc.c`.

References `syncList_s::giant_lock`, `syncListNode_s::lock`, `MC_ERR_NOT_FOUND`, `agency_s::mc_platform`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `mc_platform_s::syncList`, `syncListFind()`, `syncListNodeDestroy()`, and `syncListRemove()`.

**8.37.24.3 EXPORTMC int MC\_SyncInit (MCAgency\_t attr, int id)**

Initializes a new MobileC synchronization variable.

**Parameters:**

*attr* a MobileC agency handle

*id* the requested sync variable id

**Returns:**

new sync variable's id. May be different than the requested id.

**Note:**

Each synchronization variable may be used as a mutex, condition variable, or semaphore. However, it should only be used as one type of synchronization variable per instance, or undefined behaviour may result.

**See also:**

[MC\\_MutexLock\(\)](#), [MC\\_MutexUnlock\(\)](#), [MC\\_CondWait\(\)](#), [MC\\_CondSignal\(\)](#), [MC\\_CondBroadcast](#), [MC\\_SemaphorePost\(\)](#), [MC\\_SemaphoreWait\(\)](#)

Definition at line 1587 of file `libmc.c`.

References `syncList_s::giant_lock`, `syncListNode_s::id`, `agency_s::mc_platform`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `node`, `mc_platform_s::syncList`, `syncListAddNode()`, `syncListFind()`, and `syncListNodeNew()`.

**8.37.24.4 EXPORTMC int MC\_TerminateAgent (MCAgent\_t agent)**

Halt a running agent.

**Returns:**

0 on success, `error_code_t` on failure

Definition at line 1612 of file `libmc.c`.

References `agent_s::agent_interp`.

Referenced by `MC_DeleteAgent()`.

### 8.37.24.5 EXPORTMC int MC\_WaitAgent (MCAgency\_t *attr*)

Wait indefinitely.

**Note:**

This function is intended to block the calling thread forever. Wait for an agent arrival event  
This function blocks until an agent arrival signal is triggered, at which point it unblocks.

Definition at line 1639 of file libmc.c.

References mc\_platform\_s::agent\_queue, COND\_WAIT, mc\_platform, agency\_s::mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, and size.

### 8.37.24.6 EXPORTMC MCAgent\_t MC\_WaitRetrieveAgent (MCAgency\_t *attr*)

Wait and retrieve an agent.

**Returns:**

a valid MobileC agent on success, or NULL on failure

**Note:**

This function blocks until the arrival of an agent. The agent is retrieved after it is initialized, but before it is executed.

Definition at line 1659 of file libmc.c.

References mc\_platform\_s::agent\_queue, ListSearch(), agency\_s::mc\_platform, MC\_RECV\_AGENT, MC\_WaitSignal(), MUTEX\_LOCK, and MUTEX\_UNLOCK.

### 8.37.24.7 EXPORTMC int MC\_WaitSignal (MCAgency\_t *attr*, int *signals*)

Wait for a MobileC signal.

**Parameters:**

*attr* a MobileC agency handle

*signals* a flag of signals to wait for, of type 'enum MC\_Signal\_e'

**Returns:**

0 on success, error\_code\_t on failure

**Note:**

the parameter 'signals' may be something like 'MC\_RECV\_MESSAGE | MC\_RECV\_AGENT', etc.

## 8.37.25 Example

Definition at line 1677 of file libmc.c.

References COND\_WAIT, mc\_platform\_s::giant, mc\_platform\_s::giant\_lock, agency\_s::mc\_platform, mc\_platform\_s::MC\_signal, mc\_platform\_s::MC\_signal\_cond, mc\_platform\_s::MC\_signal\_lock, MUTEX\_LOCK, and MUTEX\_UNLOCK.

Referenced by MC\_WaitRetrieveAgent().

## 8.38 /home/dko/projects/mobilec/trunk/src/include/macros.h File Reference

```
#include <pthread.h>
#include <semaphore.h>
#include "config.h"
```

### Defines

- #define [GET\\_THREAD\\_MODE](#)(a, b) ( (a & (1<<b)) / (1<<b) )
- #define [SET\\_THREAD\\_ON](#)(a, b) a = (a | (1<<b))
- #define [SET\\_THREAD\\_OFF](#)(a, b) a = (a & ~(1<<b))
- #define [STRUCT](#)(name, members)
- #define [PTHREAD\\_STACK\\_SIZE](#) 131072
- #define [THREAD\\_T](#) pthread\_t
- #define [THREAD\\_CREATE](#)(thread\_handle, function, arg)
- #define [THREAD\\_CANCEL](#)(thread\_handle) pthread\_cancel( thread\_handle )
- #define [THREAD\\_JOIN](#)(thread\_handle) pthread\_join( thread\_handle, NULL )
- #define [MUTEX\\_T](#) pthread\_mutex\_t
- #define [MUTEX\\_INIT](#)(mutex) pthread\_mutex\_init(mutex, NULL)
- #define [MUTEX\\_DESTROY](#)(mutex) pthread\_mutex\_destroy(mutex)
- #define [MUTEX\\_LOCK](#)(mutex)
- #define [MUTEX\\_UNLOCK](#)(mutex) pthread\_mutex\_unlock( mutex )
- #define [MUTEX\\_NEW](#)(mutex)
- #define [COND\\_T](#) pthread\_cond\_t
- #define [COND\\_INIT](#)(cond) pthread\_cond\_init(cond, NULL)
- #define [COND\\_DESTROY](#)(cond) pthread\_cond\_destroy(cond)
- #define [COND\\_WAIT](#)(cond, mutex) pthread\_cond\_wait(cond, mutex )
- #define [COND\\_SLEEP](#)(cond, mutex, test)
- #define [COND\\_RESET](#)(cond, mutex) pthread\_mutex\_unlock( mutex );
- #define [COND\\_SLEEP\\_ACTION](#)(cond, mutex, action)
- #define [SIGNAL](#)(cond, mutex, action)
- #define [COND\\_BROADCAST](#)(cond) pthread\_cond\_broadcast( cond )
- #define [COND\\_SIGNAL](#)(cond) pthread\_cond\_signal( cond )
- #define [SEMAPHORE\\_T](#) sem\_t
- #define [SEMAPHORE\\_INIT](#)(sem) sem\_init(sem, 0, 0)
- #define [SEMAPHORE\\_DESTROY](#)(sem) sem\_destroy(sem)
- #define [SEMAPHORE\\_WAIT](#)(sem) sem\_wait(sem)
- #define [SEMAPHORE\\_POST](#)(sem) sem\_post(sem)
- #define [RWLOCK\\_T](#) mc\_rwlock\_t
- #define [RWLOCK\\_INIT](#)(rwlock) mc\_rwlock\_init(rwlock)
- #define [RWLOCK\\_DESTROY](#)(rwlock) mc\_rwlock\_destroy(rwlock)
- #define [RWLOCK\\_RDLOCK](#)(rwlock) mc\_rwlock\_rdlock(rwlock)
- #define [RWLOCK\\_RDUNLOCK](#)(rwlock) mc\_rwlock\_rdunlock(rwlock)
- #define [RWLOCK\\_WRLOCK](#)(rwlock) mc\_rwlock\_wrlock(rwlock)
- #define [RWLOCK\\_WRUNLOCK](#)(rwlock) mc\_rwlock\_wrunlock(rwlock)
- #define [WAKE\\_QUEUE](#)(queue, action)
- #define [SLEEP\\_QUEUE](#)(queue)



- #define [SLEEP\\_RESET](#)(queue) pthread\_mutex\_unlock( queue → thread\_mutex )
- #define [CHECK\\_NULL](#)(var, action)
- #define [WARN](#)(message)
- #define [CH\\_DATATYPE\\_SIZE](#)(type, size)
- #define [CH\\_DATATYPE\\_STRING](#)(type, string)
- #define [CH\\_DATATYPE\\_VALUE\\_STRING](#)(type, string, p)
- #define [CH\\_STRING\\_DATATYPE](#)(string, type)
- #define [CH\\_DATATYPE\\_STR\\_TO\\_VAL](#)(type, string, val)

## 8.38.1 Define Documentation

### 8.38.1.1 #define CH\_DATATYPE\_SIZE(type, size)

**Value:**

```
switch(type) {
    case CH_CHARTYPE:
        size = sizeof(char);
        break;
    case CH_INTTYPE:
        size = sizeof(int);
        break;
    case CH_UINTTYPE:
        size = sizeof(unsigned int);
        break;
    case CH_SHORTTYPE:
        size = sizeof(short);
        break;
    case CH_USHORTTYPE:
        size = sizeof(unsigned short);
        break;
    case CH_FLOATTYPE:
        size = sizeof(float);
        break;
    case CH_DOUBLETTYPE:
        size = sizeof(double);
        break;
    default:
        fprintf(stderr, "Unknown data type: %d at %s:%d",
            type, __FILE__, __LINE__);
        size=0;
}
```

Definition at line 433 of file macros.h.

Referenced by agent\_AddPersistentVariable(), agent\_xml\_compose\_\_create\_row\_nodes(), agent\_xml\_parse\_\_data(), agent\_xml\_parse\_\_fill\_row\_data(), agent\_xml\_parse\_\_row(), interpreter\_variable\_data\_Initialize(), interpreter\_variable\_data\_InitializeFromAgent(), and MC\_GetAgentReturnData().

### 8.38.1.2 #define CH\_DATATYPE\_STR\_TO\_VAL(type, string, val)

**Value:**

```
switch (type) { \
    case CH_INTTYPE: \
        *(int*)val = atoi(string); \
    break; \
    case CH_UINTTYPE: \
```

```

        *(unsigned int*)val = atoi(string); \
break; \
case CH_SHORTTYPE: \
        *(short*)val = (short)atoi(string); /*FIXME*/ \
break; \
case CH_USHORTTYPE: \
        *(unsigned short*)val = (unsigned short)atoi(string); /*FIXME*/ \
break; \
case CH_FLOATTYPE: \
        *(float*)val = strtod(string, NULL); \
break; \
case CH_DOUBLETTYPE: \
        *(double*)val = strtod(string, NULL); \
break; \
default: \
        fprintf(stderr, \
            "Unsupported data type: %d %s:%d\n", \
            type, __FILE__, __LINE__ ); \
}

```

Definition at line 547 of file macros.h.

Referenced by agent\_xml\_parse\_\_data().

### 8.38.1.3 #define CH\_DATATYPE\_STRING(type, string)

**Value:**

```

switch(type) {
    case CH_CHARTYPE:
        strcpy(string, "char");
        break;
    case CH_INTTYPE:
        strcpy(string, "int");
        break;
    case CH_UINTTYPE:
        strcpy(string, "unsigned int");
        break;
    case CH_SHORTTYPE:
        strcpy(string, "short");
        break;
    case CH_USHORTTYPE:
        strcpy(string, "unsigned short");
        break;
    case CH_FLOATTYPE:
        strcpy(string, "float");
        break;
    case CH_DOUBLETTYPE:
        strcpy(string, "double");
        break;
    default:
        fprintf(stderr,
            "Unsupported data type: %d %s:%d\n",
            type, __FILE__, __LINE__ );
}

```

Definition at line 464 of file macros.h.

Referenced by agent\_xml\_compose\_\_data().

### 8.38.1.4 #define CH\_DATATYPE\_VALUE\_STRING(type, string, p)

**Value:**

```

switch(type) {
    case CH_CHARTYPE:
        sprintf(string, "%c", *((char*)p));
        break;
    case CH_INTTYPE:
        sprintf(string, "%d", *((int*)p));
        break;
    case CH_UINTTYPE:
        sprintf(string, "%d", *((unsigned int*)p));
        break;
    case CH_SHORTTYPE:
        sprintf(string, "%d", *((short*)p));
        break;
    case CH_USHORTTYPE:
        sprintf(string, "%d", *((unsigned short*)p));
        break;
    case CH_FLOATTYPE:
        sprintf(string, "%f", *((float*)p));
        break;
    case CH_DOUBLETTYPE:
        sprintf(string, "%f", *((double*)p));
        break;
    default:
        fprintf(stderr,
            "Unsupported data type: %d %s:%d\n",
            type, __FILE__, __LINE__);
}

```

Definition at line 496 of file macros.h.

Referenced by agent\_xml\_compose\_\_create\_row\_nodes(), and agent\_xml\_compose\_\_data().

#### 8.38.1.5 #define CH\_STRING\_DATATYPE(string, type)

**Value:**

```

if (!strcmp(string, "int")) {
    type = CH_INTTYPE;
} else if (!strcmp(string, "float")) {
    type = CH_FLOATTYPE;
} else if (!strcmp(string, "double")) {
    type = CH_DOUBLETTYPE;
} else if (!strcmp(string, "unsigned int")) {
    type = CH_UINTTYPE;
} else if (!strcmp(string, "short")) {
    type = CH_SHORTTYPE;
} else if (!strcmp(string, "unsigned short")) {
    type = CH_USHORTTYPE;
} else if (!strcmp(string, "char")) {
    type = CH_CHARTYPE;
} else {
    fprintf(stderr,
        "Unsupported data type: %d %s:%d\n",
        type, __FILE__, __LINE__);
}

```

Definition at line 525 of file macros.h.

Referenced by agent\_xml\_parse\_\_data().

#### 8.38.1.6 #define CHECK\_NULL(var, action)

**Value:**

```

if ( var == NULL ) {
    fprintf(stderr, "Pointer var is null: expected otherwise.\n");
    fprintf(stderr, "Error occured at %s:%d", __FILE__, __LINE__);
    action;
}

```

Definition at line 416 of file macros.h.

Referenced by agent\_AddPersistentVariable(), agent\_datastate\_New(), agent\_xml\_parse\_\_home(), agent\_xml\_parse\_\_name(), agent\_xml\_parse\_\_owner(), agent\_xml\_parse\_\_task(), ams\_Initialize(), barrier\_node\_Initialize(), barrier\_queue\_New(), df\_request\_list\_New(), df\_request\_list\_node\_New(), df\_request\_search\_New(), fipa\_agent\_identifier\_Parse(), fipa\_word\_Parse(), http\_ParseExpression(), interpreter\_variable\_data\_Initialize(), interpreter\_variable\_data\_InitializeFromAgent(), interpreter\_variable\_data\_New(), listen\_Thread(), MC\_AclSend\_chdl(), MC\_AddAgent\_chdl(), MC\_Barrier\_chdl(), MC\_BarrierDelete\_chdl(), MC\_BarrierInit\_chdl(), MC\_CondBroadcast\_chdl(), MC\_CondReset\_chdl(), MC\_CondSignal\_chdl(), MC\_CondWait\_chdl(), MC\_DeleteAgent(), MC\_DeregisterService\_chdl(), MC\_End\_chdl(), MC\_FindAgentByID\_chdl(), MC\_FindAgentByName\_chdl(), MC\_HaltAgency\_chdl(), MC\_Initialize(), mc\_platform\_Initialize(), MC\_RegisterService(), mc\_rwlock\_init(), MC\_SearchForService(), message\_InitializeFromAgent(), message\_InitializeFromConnection(), message\_InitializeFromString(), message\_New(), message\_queue\_SendOutgoing(), message\_Send(), mtp\_http\_InitializeFromConnection(), mtp\_http\_New(), syncListNodeInit(), syncListNodeNew(), xml\_get\_cdata(), xml\_get\_text(), and xml\_new\_cdata().

#### 8.38.1.7 #define COND\_BROADCAST(cond) pthread\_cond\_broadcast( cond )

Definition at line 205 of file macros.h.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), ams\_Thread(), df\_Thread(), listen\_Thread(), MC\_Barrier(), MC\_CondBroadcast(), and MC\_SendSteerCommand().

#### 8.38.1.8 #define COND\_DESTROY(cond) pthread\_cond\_destroy(cond)

Definition at line 181 of file macros.h.

Referenced by ams\_Destroy(), barrier\_node\_Destroy(), df\_Destroy(), df\_request\_list\_node\_Destroy(), df\_request\_search\_Destroy(), mc\_platform\_Destroy(), mc\_rwlock\_destroy(), and syncListNodeDestroy().

#### 8.38.1.9 #define COND\_INIT(cond) pthread\_cond\_init(cond, NULL)

Definition at line 178 of file macros.h.

Referenced by acc\_Initialize(), ams\_Initialize(), barrier\_node\_Initialize(), df\_Initialize(), df\_request\_list\_New(), df\_request\_list\_node\_New(), df\_request\_search\_New(), mc\_platform\_Initialize(), mc\_rwlock\_init(), syncListNodeInit(), and syncListNodeNew().

#### 8.38.1.10 #define COND\_RESET(cond, mutex) pthread\_mutex\_unlock( mutex );

Definition at line 193 of file macros.h.

#### 8.38.1.11 #define COND\_SIGNAL(cond) pthread\_cond\_signal( cond )

Definition at line 207 of file macros.h.

Referenced by `acc_MessageHandlerThread()`, `agent_RunChScriptThread()`, `MC_AddAgent()`, `MC_CondSignal()`, `MC_End()`, `MC_ResetSignal()`, `mc_rwlock_rdunlock()`, `mc_rwlock_wrunlock()`, and `MC_SetAgentStatus()`.

#### 8.38.1.12 `#define COND_SLEEP(cond, mutex, test)`

**Value:**

```
if (pthread_mutex_lock( mutex )) \
printf("pthread lock error: %s:%d\n", __FILE__, __LINE__); \
if (!test) { \
    pthread_cond_wait( cond, mutex ); \
}
```

Definition at line 187 of file `macros.h`.

#### 8.38.1.13 `#define COND_SLEEP_ACTION(cond, mutex, action)`

**Value:**

```
if (pthread_mutex_lock( mutex )) \
printf("pthread lock error: %s:%d\n", __FILE__, __LINE__); \
action; \
pthread_cond_wait( cond, mutex );
```

Definition at line 195 of file `macros.h`.

Referenced by `MC_SearchForService()`.

#### 8.38.1.14 `#define COND_T pthread_cond_t`

Definition at line 176 of file `macros.h`.

Referenced by `acc_Initialize()`, `ams_Initialize()`, `barrier_node_Initialize()`, `df_Initialize()`, `df_request_list_New()`, `df_request_list_node_New()`, `df_request_search_New()`, `mc_platform_Initialize()`, `mc_rwlock_rdlock_init()`, `syncListNodeInit()`, and `syncListNodeNew()`.

#### 8.38.1.15 `#define COND_WAIT(cond, mutex) pthread_cond_wait(cond, mutex)`

Definition at line 184 of file `macros.h`.

Referenced by `acc_MessageHandlerThread()`, `acc_Thread()`, `agent_mailbox_WaitRetrieve()`, `ams_Thread()`, `df_Thread()`, `MC_Barrier()`, `MC_CondWait()`, `mc_platform_Initialize()`, `mc_rwlock_rdlock()`, `mc_rwlock_wrlock()`, `MC_SteerControl()`, `MC_WaitAgent()`, and `MC_WaitSignal()`.

#### 8.38.1.16 `#define GET_THREAD_MODE(a, b) ( (a & (1<<b)) / (1<<b) )`

Definition at line 110 of file `macros.h`.

Referenced by `MC_End()`, and `mc_platform_Initialize()`.

**8.38.1.17 #define MUTEX\_DESTROY(mutex) pthread\_mutex\_destroy(mutex)**

Definition at line 159 of file macros.h.

Referenced by agent\_Destroy(), agent\_Initialize(), ams\_Destroy(), barrier\_node\_Destroy(), df\_Destroy(), df\_request\_list\_node\_Destroy(), df\_request\_search\_Destroy(), mc\_platform\_Destroy(), mc\_rwlock\_destroy(), and syncListNodeDestroy().

**8.38.1.18 #define MUTEX\_INIT(mutex) pthread\_mutex\_init(mutex, NULL)**

Definition at line 156 of file macros.h.

Referenced by acc\_Initialize(), agent\_Copy(), agent\_Initialize(), agent\_New(), ams\_Initialize(), barrier\_node\_Initialize(), df\_Initialize(), df\_request\_list\_New(), df\_request\_list\_node\_New(), df\_request\_search\_New(), mc\_platform\_Initialize(), MC\_RegisterService(), mc\_rwlock\_init(), syncListInit(), syncListNodeInit(), and syncListNodeNew().

**8.38.1.19 #define MUTEX\_LOCK(mutex)**

**Value:**

```
if (pthread_mutex_lock( mutex )) \
    fprintf(stderr, "pthread lock error: %s:%d\n", __FILE__, __LINE__)
```

Definition at line 162 of file macros.h.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), agent\_Copy(), agent\_Destroy(), agent\_mailbox\_WaitRetrieve(), agent\_RunChScriptThread(), ams\_ManageAgentList(), ams\_Print(), ams\_Thread(), AP\_QUEUE\_SEARCH\_TEMPLATE(), AP\_QUEUE\_STD\_DEFN\_TEMPLATE(), df\_Destroy(), df\_node\_Destroy(), df\_request\_list\_Pop(), df\_SearchForService(), df\_Thread(), interpreter\_variable\_data\_Initialize(), listen\_Thread(), MC\_AddAgent(), MC\_Barrier(), MC\_CallAgentFunc(), MC\_CondBroadcast(), MC\_CondReset(), MC\_CondSignal(), MC\_CondWait(), MC\_End(), MC\_GetAgentName(), MC\_GetAgentStatus(), MC\_GetAllAgents(), MC\_HaltAgency(), MC\_MutexLock(), mc\_platform\_Initialize(), MC\_PrintAgentCode(), MC\_ResetSignal(), MC\_ResumeAgency(), MC\_RetrieveAgent(), MC\_RetrieveAgentCode(), mc\_rwlock\_rdlock(), mc\_rwlock\_rdunlock(), mc\_rwlock\_wrlock(), mc\_rwlock\_wrunlock(), MC\_SendSteerCommand(), MC\_SetAgentStatus(), MC\_Steer(), MC\_SteerControl(), MC\_SyncDelete(), MC\_SyncInit(), MC\_WaitAgent(), MC\_WaitRetrieveAgent(), MC\_WaitSignal(), and message\_queue\_SendOutgoing().

**8.38.1.20 #define MUTEX\_NEW(mutex)**

**Value:**

```
mutex = (pthread_mutex_t*)malloc(sizeof(pthread_mutex_t)); \
    if (mutex == NULL) \
        fprintf(stderr, "Memory Error. %s:%d\n", __FILE__, __LINE__); \
```

Definition at line 167 of file macros.h.

Referenced by agent\_New().

**8.38.1.21 #define MUTEX\_T pthread\_mutex\_t**

Definition at line 154 of file macros.h.

Referenced by `acc_Initialize()`, `agent_Copy()`, `agent_Initialize()`, `ams_Initialize()`, `barrier_node_Initialize()`, `df_Initialize()`, `df_request_list_New()`, `df_request_list_node_New()`, `df_request_search_New()`, `mc_platform_Initialize()`, `MC_RegisterService()`, `mc_rwlock_init()`, `syncListInit()`, `syncListNodeInit()`, and `syncListNodeNew()`.

#### 8.38.1.22 **#define MUTEX\_UNLOCK(mutex) pthread\_mutex\_unlock( mutex )**

Definition at line 165 of file `macros.h`.

Referenced by `acc_MessageHandlerThread()`, `acc_Thread()`, `agent_mailbox_WaitRetrieve()`, `agent_RunChScriptThread()`, `ams_ManageAgentList()`, `ams_Print()`, `ams_Thread()`, `AP_QUEUE_SEARCH_TEMPLATE()`, `AP_QUEUE_STD_DEFN_TEMPLATE()`, `df_request_list_Pop()`, `df_SearchForService()`, `df_Thread()`, `interpreter_variable_data_Initialize()`, `listen_Thread()`, `MC_AddAgent()`, `MC_Barrier()`, `MC_CallAgentFunc()`, `MC_CondBroadcast()`, `MC_CondReset()`, `MC_CondSignal()`, `MC_CondWait()`, `MC_End()`, `MC_GetAgentName()`, `MC_GetAgentStatus()`, `MC_GetAllAgents()`, `MC_HaltAgency()`, `MC_MutexUnlock()`, `mc_platform_Initialize()`, `MC_PrintAgentCode()`, `MC_ResetSignal()`, `MC_ResumeAgency()`, `MC_RetrieveAgent()`, `MC_RetrieveAgentCode()`, `mc_rwlock_rdlock()`, `mc_rwlock_rdunlock()`, `mc_rwlock_wrlock()`, `mc_rwlock_wrunlock()`, `MC_SendSteerCommand()`, `MC_SetAgentStatus()`, `MC_Steer()`, `MC_SteerControl()`, `MC_SyncDelete()`, `MC_SyncInit()`, `MC_WaitAgent()`, `MC_WaitRetrieveAgent()`, `MC_WaitSignal()`, and `message_queue_SendOutgoing()`.

#### 8.38.1.23 **#define PTHREAD\_STACK\_SIZE 131072**

Definition at line 133 of file `macros.h`.

#### 8.38.1.24 **#define RWLOCK\_DESTROY(rwlock) mc\_rwlock\_destroy(rwlock)**

Definition at line 249 of file `macros.h`.

Referenced by `barrier_queue_Destroy()`.

#### 8.38.1.25 **#define RWLOCK\_INIT(rwlock) mc\_rwlock\_init(rwlock)**

Definition at line 241 of file `macros.h`.

Referenced by `barrier_queue_New()`, and `syncListInit()`.

#### 8.38.1.26 **#define RWLOCK\_RDLOCK(rwlock) mc\_rwlock\_rdlock(rwlock)**

Definition at line 267 of file `macros.h`.

Referenced by `barrier_queue_Get()`, and `syncListFind()`.

#### 8.38.1.27 **#define RWLOCK\_RDUNLOCK(rwlock) mc\_rwlock\_rdunlock(rwlock)**

Definition at line 269 of file `macros.h`.

Referenced by `barrier_queue_Get()`, and `syncListFind()`.

#### 8.38.1.28 **#define RWLOCK\_T mc\_rwlock\_t**

Definition at line 234 of file `macros.h`.

Referenced by `barrier_queue_New()`, and `syncListInit()`.

#### **8.38.1.29 #define RWLOCK\_WRLOCK(rwlock) mc\_rwlock\_wrlock(rwlock)**

Definition at line 271 of file `macros.h`.

Referenced by `barrier_queue_Add()`, `barrier_queue_Delete()`, `syncListAddNode()`, `syncListDelete()`, and `syncListRemove()`.

#### **8.38.1.30 #define RWLOCK\_WRUNLOCK(rwlock) mc\_rwlock\_wrunlock(rwlock)**

Definition at line 273 of file `macros.h`.

Referenced by `barrier_queue_Add()`, `barrier_queue_Delete()`, `syncListAddNode()`, `syncListDelete()`, and `syncListRemove()`.

#### **8.38.1.31 #define SEMAPHORE\_DESTROY(sem) sem\_destroy(sem)**

Definition at line 219 of file `macros.h`.

Referenced by `syncListNodeDestroy()`.

#### **8.38.1.32 #define SEMAPHORE\_INIT(sem) sem\_init(sem, 0, 0)**

Definition at line 216 of file `macros.h`.

Referenced by `syncListNodeInit()`, and `syncListNodeNew()`.

#### **8.38.1.33 #define SEMAPHORE\_POST(sem) sem\_post(sem)**

Definition at line 224 of file `macros.h`.

Referenced by `MC_SemaphorePost()`.

#### **8.38.1.34 #define SEMAPHORE\_T sem\_t**

Definition at line 214 of file `macros.h`.

Referenced by `syncListNodeInit()`, and `syncListNodeNew()`.

#### **8.38.1.35 #define SEMAPHORE\_WAIT(sem) sem\_wait(sem)**

Definition at line 222 of file `macros.h`.

Referenced by `MC_SemaphoreWait()`.

#### **8.38.1.36 #define SET\_THREAD\_OFF(a, b) a = (a & ~(1<<b))**

Definition at line 114 of file `macros.h`.

Referenced by `MC_SetThreadOff()`, and `MC_SetThreadsAllOff()`.



**8.38.1.37 #define SET\_THREAD\_ON(a, b) a = (a | (1<<b))**

Definition at line 113 of file macros.h.

Referenced by MC\_SetThreadOn(), and MC\_SetThreadsAllOn().

**8.38.1.38 #define SIGNAL(cond, mutex, action)**

**Value:**

```
pthread_mutex_lock( mutex ); \
action; \
pthread_cond_signal( cond ); \
pthread_mutex_unlock( mutex )
```

Definition at line 200 of file macros.h.

Referenced by agent\_RunChScriptThread(), df\_Add(), and df\_AddRequest().

**8.38.1.39 #define SLEEP\_QUEUE(queue)**

**Value:**

```
if (pthread_mutex_lock( queue->thread_mutex )) \
printf("pthread lock error: %s:%d\n", __FILE__, __LINE__); \
pthread_cond_wait( queue->touched_signal, queue->thread_mutex )
```

Definition at line 287 of file macros.h.

**8.38.1.40 #define SLEEP\_RESET(queue) pthread\_mutex\_unlock( queue → thread\_mutex )**

Definition at line 291 of file macros.h.

**8.38.1.41 #define STRUCT(name, members)**

**Value:**

```
typedef struct name##_s { \
    members \
} name##_t; \
typedef name##_t* name##_p;
```

Definition at line 118 of file macros.h.

**8.38.1.42 #define THREAD\_CANCEL(thread\_handle) pthread\_cancel( thread\_handle )**

Definition at line 143 of file macros.h.

Referenced by MC\_End().

**8.38.1.43 #define THREAD\_CREATE(thread\_handle, function, arg)****Value:**

```
pthread_create( \
    thread_handle, \
    &attr, \
    function, \
    (void*) arg \
)
```

Definition at line 135 of file macros.h.

Referenced by acc\_Start(), agent\_RunChScript(), ams\_Start(), cmd\_prompt\_Start(), and df\_Start().

**8.38.1.44 #define THREAD\_JOIN(thread\_handle) pthread\_join( thread\_handle, NULL )**

Definition at line 146 of file macros.h.

Referenced by MC\_End().

**8.38.1.45 #define THREAD\_T pthread\_t**

Definition at line 134 of file macros.h.

Referenced by agent\_RunChScript().

**8.38.1.46 #define WAKE\_QUEUE(queue, action)****Value:**

```
if (pthread_mutex_trylock( queue->lock ) == 0) { \
    action; \
    pthread_cond_signal( queue->cond); \
    pthread_mutex_unlock( queue->lock); \
}
```

Definition at line 281 of file macros.h.

**8.38.1.47 #define WARN(message)****Value:**

```
fprintf(stderr, "WARNING: "); \
    fprintf(stderr, message ); \
    fprintf(stderr, " %s:%d\n", __FILE__, __LINE__ )
```

Definition at line 423 of file macros.h.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), and message\_InitializeFromAgent().

## 8.39 /home/dko/projects/mobilec/trunk/src/include/mc\_error.h File Reference

### Typedefs

- typedef enum [error\\_code\\_e](#) [error\\_code\\_t](#)

### Enumerations

- enum [error\\_code\\_e](#) {  
    [MC\\_SUCCESS](#) = 0, [MC\\_ERR](#), [MC\\_ERR\\_CONNECT](#), [MC\\_ERR\\_PARSE](#),  
    [MC\\_ERR\\_EMPTY](#), [MC\\_ERR\\_INVALID](#), [MC\\_ERR\\_INVALID\\_ARGS](#), [MC\\_ERR\\_NOT\\_FOUND](#),  
    [MC\\_ERR\\_MEMORY](#), [MC\\_ERR\\_SEND](#), [MC\\_WARN\\_DUPLICATE](#), [MC\\_SUCCESS](#) = 0,  
    [ERR](#), [MC\\_ERR\\_CONNECT](#), [MC\\_ERR\\_PARSE](#), [MC\\_ERR\\_EMPTY](#),  
    [MC\\_ERR\\_INVALID](#), [MC\\_ERR\\_INVALID\\_ARGS](#), [MC\\_ERR\\_NOT\\_FOUND](#), [MC\\_ERR\\_MEMORY](#),  
    [MC\\_ERR\\_SEND](#), [MC\\_WARN\\_DUPLICATE](#) }

#### 8.39.1 Typedef Documentation

##### 8.39.1.1 typedef enum [error\\_code\\_e](#) [error\\_code\\_t](#)

#### 8.39.2 Enumeration Type Documentation

##### 8.39.2.1 enum [error\\_code\\_e](#)

Enumerator:

*[MC\\_SUCCESS](#)*  
*[MC\\_ERR](#)*  
*[MC\\_ERR\\_CONNECT](#)*  
*[MC\\_ERR\\_PARSE](#)*  
*[MC\\_ERR\\_EMPTY](#)*  
*[MC\\_ERR\\_INVALID](#)*  
*[MC\\_ERR\\_INVALID\\_ARGS](#)*  
*[MC\\_ERR\\_NOT\\_FOUND](#)*  
*[MC\\_ERR\\_MEMORY](#)*  
*[MC\\_ERR\\_SEND](#)*  
*[MC\\_WARN\\_DUPLICATE](#)*  
*[MC\\_SUCCESS](#)*  
*[ERR](#)*  
*[MC\\_ERR\\_CONNECT](#)*  
*[MC\\_ERR\\_PARSE](#)*  
*[MC\\_ERR\\_EMPTY](#)*

*MC\_ERR\_INVALID*  
*MC\_ERR\_INVALID\_ARGS*  
*MC\_ERR\_NOT\_FOUND*  
*MC\_ERR\_MEMORY*  
*MC\_ERR\_SEND*  
*MC\_WARN\_DUPLICATE*

Definition at line 41 of file mc\_error.h.

## 8.40 /home/dko/projects/mobilec/trunk/src/include/mc\_platform.h File Reference

```
#include "acc.h"
#include "ams.h"
#include "barrier.h"
#include "cmd_prompt.h"
#include "config.h"
#include "connection.h"
#include "data_structures.h"
#include "df.h"
#include "libmc.h"
#include "../mc_sync/sync_list.h"
#include "../security/asm.h"
```

### Data Structures

- struct [mc\\_platform\\_s](#)

### Typedefs

- typedef struct [mc\\_platform\\_s](#) [mc\\_platform\\_t](#)
- typedef [mc\\_platform\\_t](#) \* [mc\\_platform\\_p](#)

### Functions

- [mc\\_platform\\_p](#) [mc\\_platform\\_Initialize](#) ([MCAgency\\_t](#) agency)
- [int](#) [mc\\_platform\\_Destroy](#) ([mc\\_platform\\_p](#) mc\_platform)

#### 8.40.1 Typedef Documentation

##### 8.40.1.1 typedef [mc\\_platform\\_t](#)\* [mc\\_platform\\_p](#)

Definition at line 112 of file [mc\\_platform.h](#).

##### 8.40.1.2 typedef struct [mc\\_platform\\_s](#) [mc\\_platform\\_t](#)

Definition at line 111 of file [mc\\_platform.h](#).

#### 8.40.2 Function Documentation

##### 8.40.2.1 [int](#) [mc\\_platform\\_Destroy](#) ([mc\\_platform\\_p](#) *mc\_platform*)

Definition at line 235 of file [mc\\_platform.c](#).

References `mc_platform_s::acc`, `acc_Destroy()`, `mc_platform_s::agent_queue`, `mc_platform_s::ams`, `ams_Destroy()`, `mc_platform_s::asm_message_queue`, `mc_platform_s::barrier_queue`, `barrier_queue_Destroy()`, `mc_platform_s::cmd_prompt`, `cmd_prompt_Destroy()`, `COND_DESTROY`, `mc_platform_s::connection_queue`, `mc_platform_s::df`, `df_Destroy()`, `mc_platform_s::giant_cond`, `mc_platform_s::giant_lock`, `mc_platform_s::interp_options`, `mc_platform_s::MC_signal_cond`, `mc_platform_s::MC_signal_lock`, `mc_platform_s::MC_steer_cond`, `mc_platform_s::MC_steer_lock`, `MC_SUCCESS`, `mc_platform_s::MC_sync_cond`, `mc_platform_s::MC_sync_lock`, `mc_platform_s::message_queue`, `MUTEX_DESTROY`, and `mc_platform_s::quit_lock`.

Referenced by `handler_QUIT()`, and `MC_End()`.

#### 8.40.2.2 `mc_platform_p mc_platform_Initialize (MCAgency_t agency)`

Definition at line 47 of file `mc_platform.c`.

References `mc_platform_s::acc`, `acc_Initialize()`, `acc_Start()`, `mc_platform_s::agent_queue`, `mc_platform_s::ams`, `ams_Initialize()`, `ams_Start()`, `mc_platform_s::asm_message_queue`, `mc_platform_s::barrier_queue`, `barrier_queue_New()`, `CHECK_NULL`, `mc_platform_s::cmd_prompt`, `cmd_prompt_Initialize()`, `cmd_prompt_Start()`, `COND_INIT`, `COND_T`, `COND_WAIT`, `mc_platform_s::connection_queue`, `agency_s::default_agentstatus`, `mc_platform_s::default_agentstatus`, `DEFAULT_HOSTNAME_LENGTH`, `mc_platform_s::df`, `df_Initialize()`, `df_Start()`, `agency_s::enable_security`, `mc_platform_s::err`, `GET_THREAD_MODE`, `mc_platform_s::giant`, `mc_platform_s::giant_cond`, `mc_platform_s::giant_lock`, `mc_platform_s::hostname`, `mc_platform_s::interp_options`, `agency_s::last_error`, `MC_ERR_MEMORY`, `MC_NO_SIGNAL`, `mc_platform`, `mc_platform_s::MC_signal`, `mc_platform_s::MC_signal_cond`, `mc_platform_s::MC_signal_lock`, `mc_platform_s::MC_steer_cond`, `mc_platform_s::MC_steer_lock`, `mc_platform_s::MC_sync_cond`, `mc_platform_s::MC_sync_lock`, `MC_THREAD_ACC`, `MC_THREAD_ALL`, `MC_THREAD_AMS`, `MC_THREAD_CP`, `MC_THREAD_DF`, `mc_platform_s::message_queue`, `MUTEX_INIT`, `MUTEX_LOCK`, `MUTEX_T`, `MUTEX_UNLOCK`, `mc_platform_s::port`, `agency_s::portno`, `mc_platform_s::quit`, `mc_platform_s::quit_lock`, `agency_s::stack_size`, `mc_platform_s::stack_size`, `mc_platform_s::syncList`, `syncListInit()`, and `agency_s::threads`.

Referenced by `MC_Initialize()`.

## 8.41 /home/dko/projects/mobilec/trunk/src/include/mc\_rwlock.h File Reference

```
#include "macros.h"
```

### Data Structures

- struct [mc\\_rwlock\\_s](#)

### Typedefs

- typedef struct [mc\\_rwlock\\_s](#) [mc\\_rwlock\\_t](#)
- typedef [mc\\_rwlock\\_t](#) \* [mc\\_rwlock\\_p](#)

### Functions

- [int mc\\_rwlock\\_init](#) ([mc\\_rwlock\\_p](#) *rwlock*)
- [int mc\\_rwlock\\_destroy](#) ([mc\\_rwlock\\_p](#) *rwlock*)
- [int mc\\_rwlock\\_rdlock](#) ([mc\\_rwlock\\_p](#) *rwlock*)
- [int mc\\_rwlock\\_rdunlock](#) ([mc\\_rwlock\\_p](#) *rwlock*)
- [int mc\\_rwlock\\_wrlock](#) ([mc\\_rwlock\\_p](#) *rwlock*)
- [int mc\\_rwlock\\_wrunlock](#) ([mc\\_rwlock\\_p](#) *rwlock*)

#### 8.41.1 Typedef Documentation

##### 8.41.1.1 typedef [mc\\_rwlock\\_t](#)\* [mc\\_rwlock\\_p](#)

Definition at line 46 of file [mc\\_rwlock.h](#).

##### 8.41.1.2 typedef struct [mc\\_rwlock\\_s](#) [mc\\_rwlock\\_t](#)

#### 8.41.2 Function Documentation

##### 8.41.2.1 [int mc\\_rwlock\\_destroy](#) ([mc\\_rwlock\\_p](#) *rwlock*)

Definition at line 61 of file [mc\\_rwlock.c](#).

References [mc\\_rwlock\\_s::cond](#), [COND\\_DESTROY](#), [mc\\_rwlock\\_s::lock](#), and [MUTEX\\_DESTROY](#).

##### 8.41.2.2 [int mc\\_rwlock\\_init](#) ([mc\\_rwlock\\_p](#) *rwlock*)

Definition at line 41 of file [mc\\_rwlock.c](#).

References [CHECK\\_NULL](#), [mc\\_rwlock\\_s::cond](#), [COND\\_INIT](#), [COND\\_T](#), [mc\\_rwlock\\_s::lock](#), [MC\\_ERR\\_MEMORY](#), [MUTEX\\_INIT](#), [MUTEX\\_T](#), [mc\\_rwlock\\_s::num\\_readers](#), [mc\\_rwlock\\_s::write\\_flag](#), and [mc\\_rwlock\\_s::write\\_request](#).

#### 8.41.2.3 `int mc_rwlock_rdlock (mc_rwlock_p rwlock)`

Definition at line 74 of file `mc_rwlock.c`.

References `mc_rwlock_s::cond`, `COND_WAIT`, `mc_rwlock_s::lock`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `mc_rwlock_s::num_readers`, `mc_rwlock_s::write_flag`, and `mc_rwlock_s::write_request`.

#### 8.41.2.4 `int mc_rwlock_rdunlock (mc_rwlock_p rwlock)`

Definition at line 89 of file `mc_rwlock.c`.

References `mc_rwlock_s::cond`, `COND_SIGNAL`, `mc_rwlock_s::lock`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, and `mc_rwlock_s::num_readers`.

#### 8.41.2.5 `int mc_rwlock_wrlock (mc_rwlock_p rwlock)`

Definition at line 102 of file `mc_rwlock.c`.

References `mc_rwlock_s::cond`, `COND_WAIT`, `mc_rwlock_s::lock`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `mc_rwlock_s::num_readers`, `mc_rwlock_s::write_flag`, and `mc_rwlock_s::write_request`.

#### 8.41.2.6 `int mc_rwlock_wrunlock (mc_rwlock_p rwlock)`

Definition at line 121 of file `mc_rwlock.c`.

References `mc_rwlock_s::cond`, `COND_SIGNAL`, `mc_rwlock_s::lock`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, and `mc_rwlock_s::write_flag`.



## 8.42 /home/dko/projects/mobilec/trunk/src/include/message.h File Reference

```
#include "config.h"
#include "mtp_http.h"
#include <mxml.h>
#include <netinet/in.h>
```

### Data Structures

- struct [message\\_s](#)

### Typedefs

- typedef enum [message\\_type\\_e](#) [message\\_type\\_t](#)
- typedef struct [message\\_s](#) [message\\_t](#)
- typedef [message\\_t](#) \* [message\\_p](#)

### Enumerations

- enum [message\\_type\\_e](#) {  
RELAY, REQUEST, SUBSCRIBE, CANCEL,  
N\_UNDRSTD, MOBILE\_AGENT, QUER\_IF, QUER\_REF,  
AGENT\_UPDATE, RETURN\_MSG, FIPA\_ACL, ENCRYPTED\_DATA,  
ENCRYPTION\_INITIALIZE, REQUEST\_ENCRYPTION\_INITIALIZE, NUM\_MESSAGE\_  
TYPE }

### Functions

- [message\\_p](#) [message\\_New](#) (void)
- [message\\_p](#) [message\\_Copy](#) ([message\\_p](#) src)
- int [message\\_InitializeFromAgent](#) (struct [mc\\_platform\\_s](#) \*[mc\\_platform](#), [message\\_p](#) message, struct [agent\\_s](#) \*[agent](#))
- int [message\\_InitializeFromConnection](#) (struct [mc\\_platform\\_s](#) \*[mc\\_platform](#), [message\\_p](#) message, struct [connection\\_s](#) \*[connection](#))
- int [message\\_InitializeFromString](#) (struct [mc\\_platform\\_s](#) \*[mc\\_platform](#), [message\\_p](#) message, const char \*string, const char \*destination\_host, int destination\_port, const char \*target)
- int [message\\_Destroy](#) ([message\\_p](#) message)
- int [message\\_Send](#) ([message\\_p](#) message)
- int [http\\_to\\_hostport](#) (const char \*http\_str, char \*\*host, int \*port, char \*\*target)

#### 8.42.1 Typedef Documentation

##### 8.42.1.1 typedef [message\\_t](#)\* [message\\_p](#)

Definition at line 108 of file [message.h](#).

**8.42.1.2** `typedef struct message_s message_t`

**8.42.1.3** `typedef enum message_type_e message_type_t`

## **8.42.2 Enumeration Type Documentation**

**8.42.2.1** `enum message_type_e`

Enumerator:

*RELAY*  
*REQUEST*  
*SUBSCRIBE*  
*CANCEL*  
*N\_UNDRSTD*  
*MOBILE\_AGENT*  
*QUER\_IF*  
*QUER\_REF*  
*AGENT\_UPDATE*  
*RETURN\_MSG*  
*FIPA\_ACL*  
*ENCRYPTED\_DATA*  
*ENCRYPTION\_INITIALIZE*  
*REQUEST\_ENCRYPTION\_INITIALIZE*  
*NUM\_MESSAGE\_TYPE*

Definition at line 47 of file message.h.

## **8.42.3 Function Documentation**

**8.42.3.1** `int http_to_hostport (const char * http_str, char ** host, int * port, char ** target)`

Definition at line 505 of file message.c.

References MC\_ERR\_PARSE.

Referenced by MC\_AclSend().

**8.42.3.2** `message_p message_Copy (message_p src)`

Definition at line 310 of file message.c.

**8.42.3.3** `int message_Destroy (message_p message)`

Definition at line 604 of file message.c.

References message\_s::addr, message\_s::agent\_xml\_flag, message\_s::from\_address, MC\_SUCCESS, message\_s::message\_body, mxmDelete(), message\_s::target, message\_s::to\_address, message\_s::update\_name, and message\_s::xml\_root.

Referenced by `acc_MessageHandlerThread()`, `acc_Thread()`, `ams_ManageAgentList()`, `MC_AclSend()`, `MC_LoadAgentFromFile()`, `MC_SendAgentMigrationMessage()`, `MC_SendAgentMigrationMessageFile()`, `message_InitializeFromConnection()`, and `message_queue_SendOutgoing()`.

#### 8.42.3.4 `int message_InitializeFromAgent (struct mc_platform_s * mc_platform, message_p message, struct agent_s * agent)`

Referenced by `ams_ManageAgentList()`.

#### 8.42.3.5 `int message_InitializeFromConnection (struct mc_platform_s * mc_platform, message_p message, struct connection_s * connection)`

#### 8.42.3.6 `int message_InitializeFromString (struct mc_platform_s * mc_platform, message_p message, const char * string, const char * destination_host, int destination_port, const char * target)`

Referenced by `MC_LoadAgentFromFile()`, `MC_SendAgentMigrationMessage()`, and `MC_SendAgentMigrationMessageFile()`.

#### 8.42.3.7 `message_p message_New (void)`

Definition at line 286 of file `message.c`.

References `message_s::addr`, `message_s::agent_xml_flag`, `CHECK_NULL`, `message_s::connect_id`, `message_s::from_address`, `message_s::http_type`, `message_s::isHTTP`, `message_s::message_body`, `message_s::message_id`, `message_s::message_type`, `message_s::target`, `message_s::to_address`, `message_s::update_name`, `message_s::update_num`, `message_s::xml_payload`, and `message_s::xml_root`.

Referenced by `acc_Thread()`, `ams_ManageAgentList()`, `MC_LoadAgentFromFile()`, `MC_SendAgentMigrationMessage()`, `MC_SendAgentMigrationMessageFile()`, and `mtp_http_CreateMessage()`.

#### 8.42.3.8 `int message_Send (message_p message)`

Definition at line 642 of file `message.c`.

References `CHECK_NULL`, `MC_ERR`, `MC_ERR_CONNECT`, `MC_ERR_SEND`, `message_s::message_body`, `mtp_http_ComposeMessage()`, `mtp_http_New()`, `mtp_http_Parse()`, `port`, `send`, `SOCKET_INPUT_SIZE`, `strtok_r`, and `message_s::to_address`.

Referenced by `acc_MessageHandlerThread()`, `MC_AclSend()`, and `message_queue_SendOutgoing()`.

## 8.43 /home/dko/projects/mobilec/trunk/src/include/mtp\_http.h File Reference

### Data Structures

- struct [mtp\\_http\\_content\\_s](#)
- struct [mtp\\_http\\_s](#)

### Defines

- #define [SOCKET\\_INPUT\\_SIZE](#) 4096

### Typedefs

- typedef struct [mtp\\_http\\_content\\_s](#) [mtp\\_http\\_content\\_t](#)
- typedef struct [mtp\\_http\\_s](#) [mtp\\_http\\_t](#)
- typedef [mtp\\_http\\_t](#) \* [mtp\\_http\\_p](#)

### Enumerations

- enum [http\\_status\\_code\\_e](#) {  
    [CONTINUE](#) = 100, [SWITCHING\\_PROTOCOLS](#), [PROCESSING](#), [OK](#) = 200,  
    [CREATED](#), [ACCEPTED](#), [NON\\_AUTHORITATIVE\\_INFORMATION](#), [NO\\_CONTENT](#),  
    [RESET\\_CONTENT](#), [PARTIAL\\_CONTENT](#), [MULTI\\_STATUS](#), [BAD\\_REQUEST](#) = 400,  
    [UNAUTHORIZED](#), [PAYMENT\\_REQUIRED](#), [FORBIDDEN](#), [NOT\\_FOUND](#),  
    [METHOD\\_NOT\\_ALLOWED](#), [NOT\\_ACCEPTABLE](#), [PROXY\\_AUTHENTICATION\\_REQUIRED](#), [REQUEST\\_TIMEOUT](#),  
    [CONFLICT](#), [GONE](#), [LENGTH\\_REQUIRED](#), [PRECONDITION\\_FAILED](#),  
    [REQUEST\\_ENTITY\\_TOO\\_LARGE](#), [REQUEST\\_URI\\_TOO\\_LONG](#), [UNSUPPORTED\\_MEDIA\\_TYPE](#), [REQUESTED\\_RANGE\\_NOT\\_SATISFIABLE](#),  
    [EXPECTATION\\_FAILED](#), [UNPROCESSABLE\\_ENTITY](#), [LOCKED](#), [FAILED\\_DEPENDANCY](#),  
    [UNORDERED\\_COLLECTION](#), [UPGRADE\\_REQUIRED](#), [RETRY\\_WITH](#) }  
    *http return status codes*
- enum [http\\_performative\\_e](#) {  
    [HTTP\\_PERFORMATIVE\\_UNDEF](#) = -1, [HTTP\\_PERFORMATIVE\\_ZERO](#) = 0, [HTTP\\_HEAD](#),  
    [HTTP\\_GET](#),  
    [HTTP\\_POST](#), [HTTP\\_PUT](#), [HTTP\\_DELETE](#), [HTTP\\_TRACE](#),  
    [HTTP\\_OPTIONS](#), [HTTP\\_CONNECT](#), [HTTP\\_RESPONSE](#), [HTTP\\_NUM\\_PERFORMATIVES](#) }  
    *http 'verbs'*

## Functions

- const char \* [http\\_GetExpression](#) (const char \*string, char \*\*expr)  
*Parse an html expression.*
- int [http\\_ParseExpression](#) (const char \*expression\_string, char \*\*name, char \*\*value)  
*Parse an expression into its name and value.*
- const char \* [http\\_ParseHeader](#) (mtp\_http\_p http, const char \*string)
- const char \* [http\\_GetToken](#) (const char \*string, char \*\*token)
- int [mtp\\_http\\_Destroy](#) (mtp\_http\_p http)
- int [mtp\\_http\\_InitializeFromConnection](#) (struct [mtp\\_http\\_s](#) \*http, struct [connection\\_s](#) \*connection)
- [mtp\\_http\\_p](#) [mtp\\_http\\_New](#) (void)
- int [mtp\\_http\\_Parse](#) (struct [mtp\\_http\\_s](#) \*http, const char \*string)
- int [mtp\\_http\\_ComposeMessage](#) (struct [message\\_s](#) \*message)
- struct [message\\_s](#) \* [mtp\\_http\\_CreateMessage](#) (mtp\_http\_t \*mtp\_http, char \*hostname, int port)

### 8.43.1 Define Documentation

#### 8.43.1.1 #define SOCKET\_INPUT\_SIZE 4096

Definition at line 144 of file mtp\_http.h.

Referenced by [message\\_InitializeFromConnection\(\)](#), [message\\_Send\(\)](#), and [mtp\\_http\\_InitializeFromConnection\(\)](#).

### 8.43.2 Typedef Documentation

#### 8.43.2.1 typedef struct mtp\_http\_content\_s mtp\_http\_content\_t

#### 8.43.2.2 typedef mtp\_http\_t\* mtp\_http\_p

Definition at line 142 of file mtp\_http.h.

#### 8.43.2.3 typedef struct mtp\_http\_s mtp\_http\_t

### 8.43.3 Enumeration Type Documentation

#### 8.43.3.1 enum http\_performative\_e

http 'verbs'

Enumerator:

*HTTP\_PERFORMATIVE\_UNDEF*  
*HTTP\_PERFORMATIVE\_ZERO*  
*HTTP\_HEAD*  
*HTTP\_GET*  
*HTTP\_POST*

*HTTP\_PUT*  
*HTTP\_DELETE*  
*HTTP\_TRACE*  
*HTTP\_OPTIONS*  
*HTTP\_CONNECT*  
*HTTP\_RESPONSE*  
*HTTP\_NUM\_PERFORMATIVES*

Definition at line 90 of file mtp\_http.h.

#### 8.43.3.2 enum http\_status\_code\_e

http return status codes

**Enumerator:**

*CONTINUE*  
*SWITCHING\_PROTOCOLS*  
*PROCESSING*  
*OK*  
*CREATED*  
*ACCEPTED*  
*NON\_AUTHORITATIVE\_INFORMATION*  
*NO\_CONTENT*  
*RESET\_CONTENT*  
*PARTIAL\_CONTENT*  
*MULTI\_STATUS*  
*BAD\_REQUEST*  
*UNAUTHORIZED*  
*PAYMENT\_REQUIRED*  
*FORBIDDEN*  
*NOT\_FOUND*  
*METHOD\_NOT\_ALLOWED*  
*NOT\_ACCEPTABLE*  
*PROXY\_AUTHENTICATION\_REQUIRED*  
*REQUEST\_TIMEOUT*  
*CONFLICT*  
*GONE*  
*LENGTH\_REQUIRED*  
*PRECONDITION\_FAILED*  
*REQUEST\_ENTITY\_TOO\_LARGE*  
*REQUEST\_URI\_TOO\_LONG*  
*UNSUPPORTED\_MEDIA\_TYPE*

***REQUESTED\_RANGE\_NOT\_SATISFIABLE***  
***EXPECTATION\_FAILED***  
***UNPROCESSABLE\_ENTITY***  
***LOCKED***  
***FAILED\_DEPENDANCY***  
***UNORDERED\_COLLECTION***  
***UPGRADE\_REQUIRED***  
***RETRY\_WITH***

Definition at line 45 of file mtp\_http.h.

### 8.43.4 Function Documentation

#### 8.43.4.1 **const char\* http\_GetExpression (const char \* *string*, char \*\* *expr*)**

Parse an html expression.

**Parameters:**

***string*** (input) The html block of text: Will parse the first expression pointed to by 'string'.  
***expr*** (output) The allocated expression

**Returns:**

A pointer to the next expression segment of the string block, or NULL.

Definition at line 210 of file mtp\_http.c.

Referenced by mtp\_http\_Parse().

#### 8.43.4.2 **const char\* http\_GetToken (const char \* *string*, char \*\* *token*)**

Definition at line 633 of file mtp\_http.c.

Referenced by http\_ParseHeader().

#### 8.43.4.3 **int http\_ParseExpression (const char \* *expression\_string*, char \*\* *name*, char \*\* *value*)**

Parse an expression into its name and value.

**Parameters:**

***expression\_string*** (input) The expression  
***name*** (output) An allocated name string or NULL  
***value*** (output) An allocated value string or NULL

**Returns:**

error\_code\_t type

**Note:**

an http expression is something like 'Date: Mon, 23 May 2005 22:38:34 GMT'  
 ' where 'Date' is the name and the remainder of the string is the value

Definition at line 269 of file mtp\_http.c.

References CHECK\_NULL, MC\_ERR\_PARSE, and MC\_SUCCESS.

Referenced by mtp\_http\_Parse().

**8.43.4.4 const char\* http\_ParseHeader (mtp\_http\_p *http*, const char \* *string*)**

Definition at line 544 of file mtp\_http.c.

References HTTP\_CONNECT, HTTP\_DELETE, HTTP\_GET, http\_GetToken(), HTTP\_HEAD, HTTP\_OPTIONS, mtp\_http\_s::http\_performative, HTTP\_PERFORMATIVE\_UNDEF, HTTP\_POST, HTTP\_PUT, HTTP\_RESPONSE, HTTP\_TRACE, mtp\_http\_s::response\_code, mtp\_http\_s::response\_string, and mtp\_http\_s::target.

Referenced by mtp\_http\_Parse().

**8.43.4.5 int mtp\_http\_ComposeMessage (struct message\_s \* *message*)**

Referenced by message\_Send().

**8.43.4.6 struct message\_s\* mtp\_http\_CreateMessage (mtp\_http\_t \* *mtp\_http*, char \* *hostname*, int *port*)** [read]

Definition at line 718 of file mtp\_http.c.

References mtp\_http\_s::content, mtp\_http\_content\_s::content\_type, mtp\_http\_content\_s::data, dynstring\_Append(), dynstring\_Destroy(), dynstring\_New(), mtp\_http\_s::host, message\_s::isHTTP, dynstring\_s::len, dynstring\_s::message, message\_s::message\_body, message\_New(), mtp\_http\_s::message\_parts, PACKAGE\_VERSION, mtp\_http\_s::target, and message\_s::to\_address.

Referenced by MC\_AclSend().

**8.43.4.7 int mtp\_http\_Destroy (mtp\_http\_p *http*)**

Definition at line 53 of file mtp\_http.c.

References mtp\_http\_s::accept\_ranges, mtp\_http\_s::boundary, mtp\_http\_s::connection, mtp\_http\_s::content, mtp\_http\_s::content\_length, mtp\_http\_content\_s::content\_type, mtp\_http\_s::content\_type, mtp\_http\_content\_s::data, mtp\_http\_s::date, mtp\_http\_s::host, mtp\_http\_s::http\_version, mtp\_http\_s::message\_parts, mtp\_http\_s::return\_code, SAFE\_FREE, mtp\_http\_s::server, mtp\_http\_s::target, and mtp\_http\_s::user\_agent.

Referenced by acc\_Thread(), and MC\_AclSend().

**8.43.4.8 int mtp\_http\_InitializeFromConnection (struct mtp\_http\_s \* *http*, struct connection\_s \* *connection*)**

Referenced by acc\_Thread().



#### 8.43.4.9 mtp\_http\_p mtp\_http\_New (void)

Definition at line 85 of file mtp\_http.c.

References CHECK\_NULL, and mtp\_http\_s::content.

Referenced by acc\_Thread(), MC\_AclSend(), and message\_Send().

#### 8.43.4.10 int mtp\_http\_Parse (struct mtp\_http\_s \* *http*, const char \* *string*)

Definition at line 329 of file mtp\_http.c.

References mtp\_http\_s::boundary, mtp\_http\_s::content, mtp\_http\_content\_s::content\_type, mtp\_http\_s::content\_type, mtp\_http\_content\_s::data, http\_GetExpression(), HTTP\_PARSE\_EXPR, http\_ParseExpression(), http\_ParseHeader(), mtp\_http\_s::http\_performative, HTTP\_POST, HTTP\_PUT, HTTP\_RESPONSE, MC\_SUCCESS, mtp\_http\_s::message\_parts, and SAFE\_FREE.

Referenced by message\_Send(), and mtp\_http\_InitializeFromConnection().

## 8.44 /home/dko/projects/mobilec/trunk/src/include/xml\_compose.h File Reference

```
#include "agent.h"
```

### Functions

- [mxml\\_node\\_t \\* agent\\_xml\\_compose \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_gaf\\_message \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_message \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_mobile\\_agent \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_agent\\_data \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_name \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_owner \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_home \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_tasks \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_task \(agent\\_p agent, int index\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_data \(agent\\_p agent, int index, interpreter\\_variable\\_data\\_t \\*interp\\_variable\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_agent\\_code \(agent\\_p agent, int index\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_row \(interpreter\\_variable\\_data\\_t \\*interp\\_variable, int index\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_create\\_row\\_nodes \(void \\*data, int index, int \\*extent, ChType\\_t type, int dim, int extent\\_index\)](#)

### 8.44.1 Function Documentation

#### 8.44.1.1 [mxml\\_node\\_t\\* agent\\_xml\\_compose \(agent\\_p agent\)](#)

Definition at line 41 of file `xml_compose.c`.

References `agent_xml_compose__gaf_message()`, `MXML_ADD_AFTER`, `MXML_ADD_TO_PARENT`, `MXML_NO_CALLBACK`, `mxmlAdd()`, `mxmlLoadString()`, and `node`.

Referenced by `message_InitializeFromAgent()`.

#### 8.44.1.2 [mxml\\_node\\_t\\* agent\\_xml\\_compose\\_\\_agent\\_code \(agent\\_p agent, int index\)](#)

Definition at line 501 of file `xml_compose.c`.

References `agent_datastate_s::agent_code_ids`, `agent_datastate_s::agent_codes`, `agent_s::datastate`, `MXML_NO_PARENT`, `mxmlElementSetAttr()`, `mxmlNewElement()`, `node`, and `xml_new_cdata()`.

Referenced by `agent_xml_compose__tasks()`.

#### 8.44.1.3 [mxml\\_node\\_t\\* agent\\_xml\\_compose\\_\\_agent\\_data \(agent\\_p agent\)](#)

Definition at line 145 of file `xml_compose.c`.

References `agent_xml_compose__home()`, `agent_xml_compose__name()`, `agent_xml_compose__owner()`, `agent_xml_compose__tasks()`, `MXML_ADD_AFTER`, `mxmlAdd()`, `mxmlNewElement()`, and `node`.

Referenced by `agent_xml_compose__mobile_agent()`.

**8.44.1.4 mxml\_node\_t\* agent\_xml\_compose\_\_create\_row\_nodes (void \* *data*, int *index*, int \* *extent*, ChType\_t *type*, int *dim*, int *extent\_index*)**

Definition at line 552 of file xml\_compose.c.

References agent\_xml\_compose\_\_create\_row\_nodes(), CH\_DATATYPE\_SIZE, CH\_DATATYPE\_VALUE\_STRING, MXML\_ADD\_AFTER, MXML\_ADD\_TO\_PARENT, MXML\_NO\_PARENT, mxmlAdd(), mxmlElementSetAttr(), mxmlNewElement(), mxmlNewText(), node, and size.

Referenced by agent\_xml\_compose\_\_create\_row\_nodes(), and agent\_xml\_compose\_\_row().

**8.44.1.5 mxml\_node\_t\* agent\_xml\_compose\_\_data (agent\_p *agent*, int *index*, interpreter\_variable\_data\_t \* *interp\_variable*)**

Definition at line 427 of file xml\_compose.c.

References agent\_xml\_compose\_\_row(), interpreter\_variable\_data\_s::array\_dim, CH\_DATATYPE\_STRING, CH\_DATATYPE\_VALUE\_STRING, interpreter\_variable\_data\_s::data, interpreter\_variable\_data\_s::data\_type, MXML\_ADD\_AFTER, mxmlAdd(), mxmlElementSetAttr(), mxmlNewElement(), interpreter\_variable\_data\_s::name, and node.

Referenced by agent\_xml\_compose\_\_task().

**8.44.1.6 mxml\_node\_t\* agent\_xml\_compose\_\_gaf\_message (agent\_p *agent*)**

Definition at line 61 of file xml\_compose.c.

References agent\_xml\_compose\_\_message(), MXML\_ADD\_AFTER, mxmlAdd(), mxmlNewElement(), and node.

Referenced by agent\_xml\_compose().

**8.44.1.7 mxml\_node\_t\* agent\_xml\_compose\_\_home (agent\_p *agent*)**

Definition at line 237 of file xml\_compose.c.

References agent\_s::home, mxmlNewElement(), mxmlNewText(), and node.

Referenced by agent\_xml\_compose\_\_agent\_data().

**8.44.1.8 mxml\_node\_t\* agent\_xml\_compose\_\_message (agent\_p *agent*)**

Definition at line 80 of file xml\_compose.c.

References agent\_s::agent\_type, agent\_xml\_compose\_\_mobile\_agent(), MC\_LOCAL\_AGENT, MC\_REMOTE\_AGENT, MC\_RETURN\_AGENT, MXML\_ADD\_AFTER, mxmlAdd(), mxmlElementSetAttr(), mxmlNewElement(), and node.

Referenced by agent\_xml\_compose\_\_gaf\_message().

**8.44.1.9 mxml\_node\_t\* agent\_xml\_compose\_\_mobile\_agent (agent\_p *agent*)**

Definition at line 124 of file xml\_compose.c.

References agent\_xml\_compose\_\_agent\_data(), MXML\_ADD\_AFTER, mxmlAdd(), mxmlNewElement(), and node.

Referenced by `agent_xml_compose__message()`.

#### **8.44.1.10 `mxml_node_t* agent_xml_compose__name (agent_p agent)`**

Definition at line 205 of file `xml_compose.c`.

References `mxmlNewElement()`, `mxmlNewText()`, `agent_s::name`, and `node`.

Referenced by `agent_xml_compose__agent_data()`.

#### **8.44.1.11 `mxml_node_t* agent_xml_compose__owner (agent_p agent)`**

Definition at line 221 of file `xml_compose.c`.

References `mxmlNewElement()`, `mxmlNewText()`, `node`, and `agent_s::owner`.

Referenced by `agent_xml_compose__agent_data()`.

#### **8.44.1.12 `mxml_node_t* agent_xml_compose__row (interpreter_variable_data_t * interp_variable, int index)`**

Definition at line 529 of file `xml_compose.c`.

References `agent_xml_compose__create_row_nodes()`, `interpreter_variable_data_s::array_dim`, `interpreter_variable_data_s::array_extent`, `interpreter_variable_data_s::data`, `interpreter_variable_data_s::data_type`, and `node`.

Referenced by `agent_xml_compose__data()`.

#### **8.44.1.13 `mxml_node_t* agent_xml_compose__task (agent_p agent, int index)`**

Definition at line 310 of file `xml_compose.c`.

References `agent_task_s::agent_return_data`, `agent_task_s::agent_variable_list`, `agent_xml_compose__data()`, `agent_task_s::code_id`, `agent_s::datastate`, `MXML_ADD_AFTER`, `mxmlAdd()`, `mxmlElementSetAttr()`, `mxmlNewElement()`, `node`, `agent_task_s::persistent`, `agent_datastate_s::persistent`, `agent_task_s::server_name`, `agent_task_s::task_completed`, `agent_datastate_s::tasks`, and `agent_task_s::var_name`.

Referenced by `agent_xml_compose__tasks()`.

#### **8.44.1.14 `mxml_node_t* agent_xml_compose__tasks (agent_p agent)`**

Definition at line 253 of file `xml_compose.c`.

References `agent_xml_compose__agent_code()`, `agent_xml_compose__task()`, `agent_s::datastate`, `MXML_ADD_AFTER`, `mxmlAdd()`, `mxmlElementSetAttr()`, `mxmlNewElement()`, `node`, `agent_datastate_s::number_of_tasks`, and `agent_datastate_s::task_progress`.

Referenced by `agent_xml_compose__agent_data()`.

## 8.45 /home/dko/projects/mobilec/trunk/src/include/xml\_helper.h File Reference

```
#include <mxml.h>
```

### Functions

- [mxml\\_node\\_t \\* xml\\_find\\_sibling](#) (const [mxml\\_node\\_t](#) \**node*, const char \**sibling\_name*)
- [char \\* xml\\_get\\_cdata](#) (const [mxml\\_node\\_t](#) \**node*)
- [mxml\\_node\\_t \\* xml\\_get\\_child](#) (const [mxml\\_node\\_t](#) \**node*, const char \**child\_name*, [int](#) *descend*)
- [mxml\\_node\\_t \\* xml\\_get\\_deep\\_child](#) (const [mxml\\_node\\_t](#) \**parent*, const char \*\**child\_path*)
- [mxml\\_node\\_t \\* xml\\_get\\_next\\_element](#) (const [mxml\\_node\\_t](#) \**node*)
- [char \\* xml\\_get\\_text](#) (const [mxml\\_node\\_t](#) \**node*)
- [const char \\* xml\\_get\\_element\\_name](#) (const [mxml\\_node\\_t](#) \**node*)
- [mxml\\_node\\_t \\* xml\\_new\\_cdata](#) ([mxml\\_node\\_t](#) \**parent*, const char \**text*)
- [const char \\* whitespace\\_cb](#) ([mxml\\_node\\_t](#) \**node*, [int](#) *where*)

### 8.45.1 Function Documentation

#### 8.45.1.1 [const char \\* whitespace\\_cb](#) ([mxml\\_node\\_t](#) \* *node*, [int](#) *where*)

Definition at line 571 of file `testmxml.c`.

References `mxml_node_s::child`, `mxml_value_u::element`, `MXML_WS_AFTER_CLOSE`, `MXML_WS_AFTER_OPEN`, `MXML_WS_BEFORE_CLOSE`, `MXML_WS_BEFORE_OPEN`, `mxml_value_s::name`, `mxml_node_s::parent`, and `mxml_node_s::value`.

Referenced by `main()`.

#### 8.45.1.2 [mxml\\_node\\_t\\* xml\\_find\\_sibling](#) (const [mxml\\_node\\_t](#) \* *node*, const char \* *sibling\_name*)

Definition at line 49 of file `xml_helper.c`.

References `MXML_NO_DESCEND`, `mxmlFindElement()`, `mxml_node_s::parent`, and `xml_get_element_name()`.

#### 8.45.1.3 [char\\* xml\\_get\\_cdata](#) (const [mxml\\_node\\_t](#) \* *node*)

Definition at line 71 of file `xml_helper.c`.

References `CHECK_NULL`, `MXML_ELEMENT`, `mxml_node_s::type`, and `xml_get_element_name()`.

Referenced by `xml_get_text()`.

#### 8.45.1.4 [mxml\\_node\\_t\\* xml\\_get\\_child](#) (const [mxml\\_node\\_t](#) \* *node*, const char \* *child\_name*, [int](#) *descend*)

Definition at line 104 of file `xml_helper.c`.

References `mxmlFindElement()`.

Referenced by `agent_return_xml_parse()`, `agent_xml_parse__agent_data()`, `agent_xml_parse__data()`, `agent_xml_parse__mobile_agent()`, and `xml_get_deep_child()`.

**8.45.1.5** `mxml_node_t* xml_get_deep_child (const mxml_node_t * parent, const char ** child_path)`

Definition at line 123 of file `xml_helper.c`.

References `MXML_NO_DESCEND`, `node`, and `xml_get_child()`.

**8.45.1.6** `const char* xml_get_element_name (const mxml_node_t * node)`

Definition at line 217 of file `xml_helper.c`.

References `mxml_value_u::element`, `MXML_ELEMENT`, `mxml_value_s::name`, `mxml_node_s::type`, and `mxml_node_s::value`.

Referenced by `agent_xml_parse__data()`, `agent_xml_parse__mobile_agent()`, `agent_xml_parse__row()`, `xml_find_sibling()`, and `xml_get_cdata()`.

**8.45.1.7** `mxml_node_t* xml_get_next_element (const mxml_node_t * node)`

Definition at line 137 of file `xml_helper.c`.

References `MXML_ELEMENT`, `mxml_node_s::next`, and `mxml_node_s::type`.

**8.45.1.8** `char* xml_get_text (const mxml_node_t * node)`

Definition at line 155 of file `xml_helper.c`.

References `CHECK_NULL`, `mxml_node_s::child`, `mxml_value_u::element`, `MXML_ELEMENT`, `MXML_TEXT`, `mxml_value_s::name`, `mxml_node_s::next`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::type`, `mxml_node_s::value`, and `xml_get_cdata()`.

Referenced by `agent_xml_parse__agent_code()`, `agent_xml_parse__home()`, `agent_xml_parse__name()`, and `agent_xml_parse__owner()`.

**8.45.1.9** `mxml_node_t* xml_new_cdata (mxml_node_t * parent, const char * text)`

Definition at line 230 of file `xml_helper.c`.

References `CHECK_NULL`, `mxmlNewElement()`, and `node`.

Referenced by `agent_xml_compose__agent_code()`.

## 8.46 /home/dko/projects/mobilec/trunk/src/include/xml\_parser.h File Reference

```
#include <mxml.h>
#include "macros.h"
#include "agent.h"
```

### Functions

- [STRUCT](#) (xml\_parser, const [mxml\\_node\\_t](#) \*root; const [mxml\\_node\\_t](#) \*node;)
- [error\\_code\\_t agent\\_xml\\_parse](#) ([agent\\_p](#) agent)
- [error\\_code\\_t agent\\_xml\\_parse\\_\\_mobile\\_agent](#) ([agent\\_p](#) agent, [xml\\_parser\\_p](#) xml\_parser)
- [error\\_code\\_t agent\\_xml\\_parse\\_\\_agent\\_data](#) ([agent\\_p](#) agent, [xml\\_parser\\_p](#) xml\_parser)
- [error\\_code\\_t agent\\_xml\\_parse\\_\\_name](#) ([agent\\_p](#) agent, [xml\\_parser\\_p](#) xml\_parser)
- [error\\_code\\_t agent\\_xml\\_parse\\_\\_owner](#) ([agent\\_p](#) agent, [xml\\_parser\\_p](#) xml\_parser)
- [error\\_code\\_t agent\\_xml\\_parse\\_\\_home](#) ([agent\\_p](#) agent, [xml\\_parser\\_p](#) xml\_parser)
- [error\\_code\\_t agent\\_xml\\_parse\\_\\_tasks](#) ([agent\\_p](#) agent, [xml\\_parser\\_p](#) xml\_parser)
- [error\\_code\\_t agent\\_xml\\_parse\\_\\_task](#) ([agent\\_p](#) agent, [xml\\_parser\\_p](#) xml\_parser, [int](#) index)
- [error\\_code\\_t agent\\_xml\\_parse\\_\\_data](#) ([agent\\_p](#) agent, [xml\\_parser\\_p](#) xml\_parser, [int](#) index)
- [error\\_code\\_t agent\\_xml\\_parse\\_\\_row](#) ([interpreter\\_variable\\_data\\_t](#) \*interp\_variable, [xml\\_parser\\_p](#) xml\_parser, [int](#) index)
- void [agent\\_xml\\_parse\\_\\_fill\\_row\\_data](#) (void \*data, [ChType\\_t](#) type, [int](#) \*extent, const [mxml\\_node\\_t](#) \*node, [int](#) \*index)
- [error\\_code\\_t agent\\_xml\\_parse\\_\\_agent\\_code](#) ([agent\\_p](#) agent, [int](#) index, [xml\\_parser\\_p](#) xml\_parser)
- [error\\_code\\_t message\\_xml\\_parse](#) ([message\\_p](#) message)
- [error\\_code\\_t message\\_xml\\_parse\\_\\_message](#) ([message\\_p](#) message, [xml\\_parser\\_p](#) xml\_parser)

### 8.46.1 Function Documentation

#### 8.46.1.1 [error\\_code\\_t agent\\_xml\\_parse](#) ([agent\\_p](#) agent)

Definition at line 52 of file `xml_parser.c`.

References `agent_xml_parse__mobile_agent()`, `agent_s::datastate`, `MC_SUCCESS`, and `agent_datastate_s::xml_agent_root`.

Referenced by `agent_initialize()`.

#### 8.46.1.2 [error\\_code\\_t agent\\_xml\\_parse\\_\\_agent\\_code](#) ([agent\\_p](#) agent, [int](#) index, [xml\\_parser\\_p](#) xml\_parser)

Definition at line 782 of file `xml_parser.c`.

References `agent_datastate_s::agent_code`, `agent_datastate_s::agent_code_ids`, `agent_datastate_s::agent_codes`, `agent_task_s::code_id`, `agent_s::datastate`, `MC_SUCCESS`, `mxmlElementGetAttr()`, `agent_datastate_s::number_of_tasks`, `agent_datastate_s::task_progress`, `agent_datastate_s::tasks`, and `xml_get_text()`.

Referenced by `agent_xml_parse__tasks()`.

**8.46.1.3 error\_code\_t agent\_xml\_parse\_\_agent\_data (agent\_p agent, xml\_parser\_p xml\_parser)**

Definition at line 93 of file xml\_parser.c.

References agent\_xml\_parse\_\_home(), agent\_xml\_parse\_\_name(), agent\_xml\_parse\_\_owner(), agent\_xml\_parse\_\_tasks(), MC\_ERR\_PARSE, MC\_SUCCESS, and xml\_get\_child().

Referenced by agent\_xml\_parse\_\_mobile\_agent().

**8.46.1.4 error\_code\_t agent\_xml\_parse\_\_data (agent\_p agent, xml\_parser\_p xml\_parser, int index)**

Definition at line 457 of file xml\_parser.c.

References agent\_task\_s::agent\_return\_data, agent\_task\_s::agent\_variable\_list, agent\_xml\_parse\_\_row(), interpreter\_variable\_data\_s::array\_dim, CH\_DATATYPE\_SIZE, CH\_DATATYPE\_STR\_TO\_VAL, CH\_STRING\_DATATYPE, interpreter\_variable\_data\_s::data, interpreter\_variable\_data\_s::data\_type, agent\_s::datastate, interpreter\_variable\_data\_New(), MC\_ERR\_PARSE, MC\_SUCCESS, mxmlElementGetAttr(), interpreter\_variable\_data\_s::name, mxml\_node\_s::parent, agent\_task\_s::persistent, agent\_datastate\_s::tasks, xml\_get\_child(), and xml\_get\_element\_name().

Referenced by agent\_xml\_parse\_\_task().

**8.46.1.5 void agent\_xml\_parse\_\_fill\_row\_data (void \* data, ChType\_t type, int \* extent, const mxml\_node\_t \* node, int \* index)**

Definition at line 666 of file xml\_parser.c.

References agent\_xml\_parse\_\_fill\_row\_data(), CH\_DATATYPE\_SIZE, mxml\_node\_s::child, MXML\_DESCEND\_FIRST, MXML\_ELEMENT, MXML\_TEXT, mxmlFindElement(), mxml\_text\_s::string, strtok\_r, mxml\_value\_u::text, mxml\_node\_s::type, and mxml\_node\_s::value.

Referenced by agent\_xml\_parse\_\_fill\_row\_data(), and agent\_xml\_parse\_\_row().

**8.46.1.6 error\_code\_t agent\_xml\_parse\_\_home (agent\_p agent, xml\_parser\_p xml\_parser)**

Definition at line 201 of file xml\_parser.c.

References CHECK\_NULL, agent\_s::home, MC\_SUCCESS, and xml\_get\_text().

Referenced by agent\_return\_xml\_parse(), and agent\_xml\_parse\_\_agent\_data().

**8.46.1.7 error\_code\_t agent\_xml\_parse\_\_mobile\_agent (agent\_p agent, xml\_parser\_p xml\_parser)**

Definition at line 65 of file xml\_parser.c.

References agent\_xml\_parse\_\_agent\_data(), MC\_ERR\_PARSE, xml\_get\_child(), and xml\_get\_element\_name().

Referenced by agent\_xml\_parse().

**8.46.1.8 error\_code\_t agent\_xml\_parse\_\_name (agent\_p agent, xml\_parser\_p xml\_parser)**

Definition at line 148 of file xml\_parser.c.

References CHECK\_NULL, MC\_ERR\_PARSE, MC\_SUCCESS, agent\_s::name, and xml\_get\_text().



Referenced by agent\_return\_xml\_parse(), and agent\_xml\_parse\_\_agent\_data().

#### 8.46.1.9 error\_code\_t agent\_xml\_parse\_\_owner (agent\_p agent, xml\_parser\_p xml\_parser)

Definition at line 174 of file xml\_parser.c.

References CHECK\_NULL, MC\_SUCCESS, agent\_s::owner, and xml\_get\_text().

Referenced by agent\_return\_xml\_parse(), and agent\_xml\_parse\_\_agent\_data().

#### 8.46.1.10 error\_code\_t agent\_xml\_parse\_\_row (interpreter\_variable\_data\_t \* interp\_variable, xml\_parser\_p xml\_parser, int index)

Definition at line 602 of file xml\_parser.c.

References agent\_xml\_parse\_\_fill\_row\_data(), interpreter\_variable\_data\_s::array\_dim, interpreter\_variable\_data\_s::array\_extent, CH\_DATATYPE\_SIZE, interpreter\_variable\_data\_s::data, interpreter\_variable\_data\_s::data\_type, MC\_SUCCESS, and xml\_get\_element\_name().

Referenced by agent\_xml\_parse\_\_data().

#### 8.46.1.11 error\_code\_t agent\_xml\_parse\_\_task (agent\_p agent, xml\_parser\_p xml\_parser, int index)

Definition at line 365 of file xml\_parser.c.

References agent\_xml\_parse\_\_data(), CHECK\_NULL, agent\_task\_s::code\_id, agent\_s::datastate, MC\_ERR\_PARSE, MC\_SUCCESS, MXML\_DESCEND\_FIRST, MXML\_NO\_DESCEND, mxmElementGetAttr(), mxmFindElement(), agent\_task\_s::server\_name, agent\_task\_s::task\_completed, agent\_datastate\_s::tasks, and agent\_task\_s::var\_name.

Referenced by agent\_xml\_parse\_\_tasks().

#### 8.46.1.12 error\_code\_t agent\_xml\_parse\_\_tasks (agent\_p agent, xml\_parser\_p xml\_parser)

Definition at line 227 of file xml\_parser.c.

References agent\_datastate\_s::agent\_code, agent\_datastate\_s::agent\_code\_ids, agent\_datastate\_s::agent\_codes, agent\_task\_New(), agent\_xml\_parse\_\_agent\_code(), agent\_xml\_parse\_\_task(), agent\_s::datastate, MC\_ERR\_PARSE, MXML\_DESCEND, MXML\_DESCEND\_FIRST, MXML\_NO\_DESCEND, mxmElementGetAttr(), mxmFindElement(), agent\_datastate\_s::number\_of\_tasks, agent\_datastate\_s::task\_progress, and agent\_datastate\_s::tasks.

Referenced by agent\_return\_xml\_parse(), and agent\_xml\_parse\_\_agent\_data().

#### 8.46.1.13 error\_code\_t message\_xml\_parse (message\_p message)

Definition at line 858 of file xml\_parser.c.

Referenced by acc\_Thread(), and message\_InitializeFromConnection().

**8.46.1.14** `error_code_t message_xml_parse__message (message_p message, xml_parser_p xml_parser)`

Definition at line 913 of file `xml_parser.c`.

**8.46.1.15** `STRUCT (xml_parser, const mxml_node_t *root;const mxml_node_t *node;)`

## 8.47 /home/dko/projects/mobilec/trunk/src/libmc.c File Reference

```
#include <unistd.h>
#include <pthread.h>
#include <embedch.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/time.h>
#include "include/libmc.h"
#include "include/macros.h"
#include "include/mc_platform.h"
#include "include/message.h"
#include "include/data_structures.h"
#include "include/fipa_acl_envelope.h"
#include "include/fipa_acl.h"
#include "include/agent.h"
#include "include/agent_task.h"
```

### Defines

- #define [HOST\\_NAME\\_MAX](#) 255

### Functions

- [int MC\\_AclDestroy](#) (struct [fipa\\_acl\\_message\\_s](#) \*message)  
*Destroy a FIPA ACL message.*
- EXPORTMC [fipa\\_acl\\_message\\_t](#) \* [MC\\_AclNew](#) (void)  
*Allocate a new ACL Message.*
- EXPORTMC [int MC\\_AclPost](#) ([MCAgent\\_t](#) agent, struct [fipa\\_acl\\_message\\_s](#) \*message)  
*Post ACL message to agent.*
- EXPORTMC [fipa\\_acl\\_message\\_t](#) \* [MC\\_AclReply](#) ([fipa\\_acl\\_message\\_t](#) \*acl\_message)  
*Reply to an ACL message.*
- EXPORTMC [fipa\\_acl\\_message\\_t](#) \* [MC\\_AclRetrieve](#) ([MCAgent\\_t](#) agent)  
*Retrieve an ACL message.*
- EXPORTMC [int MC\\_AclSend](#) ([MCAgency\\_t](#) attr, [fipa\\_acl\\_message\\_t](#) \*acl)  
*Send a composed ACL Message.*
- EXPORTMC [fipa\\_acl\\_message\\_t](#) \* [MC\\_AclWaitRetrieve](#) ([MCAgent\\_t](#) agent)  
*Wait for and retrieve an ACL message.*

- `int MC_AclSetPerformative (fipa_acl_message_t *acl, enum fipa_performative_e performative)`
- `int MC_AclSetSender (fipa_acl_message_t *acl, const char *name, const char *address)`
- `int MC_AclAddReceiver (fipa_acl_message_t *acl, const char *name, const char *address)`
- `int MC_AclAddReplyTo (fipa_acl_message_t *acl, const char *name, const char *address)`
- `int MC_AclSetContent (fipa_acl_message_t *acl, const char *content)`
- `EXPORTMC int MC_AddAgent (MCAgency_t attr, MCAgent_t agent)`  
*Add an agent to the agency 'attr'.*
- `const void * MC_AgentVariableRetrieve (MCAgent_t agent, const char *var_name, int task_num)`  
*Retrieve a pointer to a previously saved variable.*
- `int MC_AgentVariableSave (MCAgent_t agent, const char *var_name)`  
*Mark an agent variable for saving.*
- `int MC_Barrier (MCAgency_t attr, int id)`
- `EXPORTMC int MC_BarrierInit (MCAgency_t attr, int id, int num_procs)`  
*Initialize a MobileC Barrier.*
- `EXPORTMC int MC_BarrierDelete (MCAgency_t attr, int id)`  
*Find and delete an initialized MobileC Barrier.*
- `EXPORTMC int MC_CallAgentFunc (MCAgent_t agent, const char *funcName, void *returnVal, void *varg)`  
*Calls a function defined in an agent.*
- `EXPORTMC int MC_ChInitializeOptions (MCAgency_t attr, ChOptions_t *options)`  
*Use custom ChOptions\_t type for internal Ch interpreter.*
- `MCAgent_t MC_ComposeAgent (const char *name, const char *home, const char *owner, const char *code, const char *return_var_name, const char *server, int persistent)`  
*Compose a new agent dynamically without using a prewritten XML file.*
- `EXPORTMC int MC_CondBroadcast (MCAgency_t attr, int id)`  
*Wakes up all agents/threads waiting on a condition variable.*
- `EXPORTMC int MC_CondSignal (MCAgency_t attr, int id)`  
*Wakes up at least one thread waiting on a condition variable.*
- `EXPORTMC int MC_CondWait (MCAgency_t attr, int id)`  
*Wait on a MobileC synchronization variable.*
- `EXPORTMC int MC_CondReset (MCAgency_t attr, int id)`  
*Reset a previously signalled MobileC condition variable.*
- `int MC_CopyAgent (MCAgent_t *agent_out, const MCAgent_t agent_in)`  
*Performs a deep-copy of an agent structure.*
- `EXPORTMC int MC_DeleteAgent (MCAgent_t agent)`  
*Stop and remove an agent.*

- `int MC_DestroyServiceSearchResult` (char \*\*agentName, char \*\*serviceName, `int` \*agentID, `int` numResult)  
*Free memory allocated by a Service Search operation.*
- `int MC_DeregisterService` (MCAgency\_t agency, `int` agentID, const char \*serviceName)
- EXPORTMC `int MC_End` (MCAgency\_t agency)  
*End an agency.*
- EXPORTMC `MCAgent_t MC_FindAgentByName` (MCAgency\_t attr, const char \*name)  
*Find an agent by its name.*
- EXPORTMC `MCAgent_t MC_FindAgentByID` (MCAgency\_t attr, `int` ID)  
*Find an agent by its id.*
- `time_t MC_GetAgentArrivalTime` (MCAgent\_t agent)
- EXPORTMC `int MC_GetAgentStatus` (MCAgent\_t agent)  
*Get an agent's current status.*
- EXPORTMC char \* `MC_GetAgentXMLString` (MCAgent\_t agent)  
*Get an agent's xml string.*
- EXPORTMC void \* `MC_GetAgentExecEngine` (MCAgent\_t agent)  
*Retrieve an agent's Ch interpreter.*
- EXPORTMC `int MC_GetAgentID` (MCAgent\_t agent)  
*Retrieve an agent's id.*
- EXPORTMC char \* `MC_GetAgentName` (MCAgent\_t agent)
- EXPORTMC `int MC_GetAgentReturnData` (MCAgent\_t agent, `int` task\_num, void \*\*data, `int` \*dim, `int` \*\*extent)  
*Get an agent's return data.*
- EXPORTMC `int MC_GetAgentNumTasks` (MCAgent\_t agent)  
*Retrive the number of tasks an agent has.*
- EXPORTMC enum `MC_AgentType_e MC_GetAgentType` (MCAgent\_t agent)  
*Get an agent's type.*
- `int MC_GetAllAgents` (MCAgency\_t attr, MCAgent\_t \*\*agents, `int` \*num\_agents)
- EXPORTMC `int MC_HaltAgency` (MCAgency\_t attr)  
*Halt an agency: Do not process new entries in queues.*
- EXPORTMC `MCAgency_t MC_Initialize` (`int` port, MCAgencyOptions\_t \*options)  
*Initialize and start a MobileC agency.*
- EXPORTMC `int MC_InitializeAgencyOptions` (struct MCAgencyOptions\_s \*options)  
*Initialize MobileC options.*
- EXPORTMC `int MC_LoadAgentFromFile` (MCAgency\_t attr, const char \*filename)

*Load an agent from a file into an agency.*

- EXPORTMC [int MC\\_MutexLock](#) ([MCAgency\\_t](#) attr, [int](#) id)  
*Locks a MobileC synchronization variable as a mutex.*
- EXPORTMC [int MC\\_MutexUnlock](#) ([MCAgency\\_t](#) attr, [int](#) id)
- EXPORTMC [int MC\\_PrintAgentCode](#) ([MCAgent\\_t](#) agent)  
*Prints an agents code to stdout.*
- EXPORTMC [int MC\\_RegisterService](#) ([MCAgency\\_t](#) agency, [MCAgent\\_t](#) agent, [int](#) agentID, const char \*agentName, char \*\*serviceNames, [int](#) numServices)  
*Register a new service with the Directory Facilitator.*
- EXPORTMC [int MC\\_ResumeAgency](#) ([MCAgency\\_t](#) attr)  
*Resumes a halted agency.*
- EXPORTMC [MCAgent\\_t MC\\_RetrieveAgent](#) ([MCAgency\\_t](#) attr)  
*Retrieves the oldest agent from an agency.*
- EXPORTMC char \* [MC\\_RetrieveAgentCode](#) ([MCAgent\\_t](#) agent)  
*Retrieves an agent's Ch code.*
- EXPORTMC [int MC\\_ResetSignal](#) ([MCAgency\\_t](#) attr)  
*Reset a MobileC signal.*
- EXPORTMC [int MC\\_SearchForService](#) ([MCAgency\\_t](#) attr, const char \*searchString, char \*\*\*agentNames, char \*\*\*serviceNames, [int](#) \*\*agentIDs, [int](#) \*numResults)  
*Search the directory facilitator for a service.*
- EXPORTMC [int MC\\_SemaphorePost](#) ([MCAgency\\_t](#) attr, [int](#) id)  
*Post to a MobileC synchronization variable semaphore.*
- EXPORTMC [int MC\\_SemaphoreWait](#) ([MCAgency\\_t](#) attr, [int](#) id)  
*Decreases a MobileC synchronization variable semaphore count by one.*
- [int MC\\_SendCh](#) ([MCAgency\\_t](#) attr, const char \*filename, const char \*remotehost, [int](#) port)
- EXPORTMC [int MC\\_SendAgentMigrationMessage](#) ([MCAgency\\_t](#) attr, const char \*string, const char \*hostname, [int](#) port)  
*Sends an agent migration message.*
- EXPORTMC [int MC\\_SendAgentMigrationMessageFile](#) ([MCAgency\\_t](#) attr, const char \*filename, const char \*hostname, [int](#) port)  
*Sends an agent migration message.*
- EXPORTMC [int MC\\_SendSteerCommand](#) ([MCAgency\\_t](#) attr, enum [MC\\_SteerCommand\\_e](#) cmd)
- [int MC\\_SetAgentStatus](#) ([MCAgent\\_t](#) agent, [int](#) status)  
*Set an agent's status.*
- [int MC\\_SetDefaultAgentStatus](#) ([MCAgency\\_t](#) agency, enum [MC\\_AgentStatus\\_e](#) status)  
*Sets default incoming agent status.*

- EXPORTMC [int MC\\_SetThreadOn](#) (MCAgencyOptions\_t \*options, enum [MC\\_ThreadIndex\\_e](#) index)  
*Sets a MobileC thread to "on" status.*
- EXPORTMC [int MC\\_SetThreadsAllOn](#) (MCAgencyOptions\_t \*options)  
*Set all Mobile-C threads on.*
- EXPORTMC [int MC\\_SetThreadOff](#) (MCAgencyOptions\_t \*options, enum [MC\\_ThreadIndex\\_e](#) index)  
*Sets a MobileC thread to "off" status.*
- EXPORTMC [int MC\\_SetThreadsAllOff](#) (MCAgencyOptions\_t \*options)  
*Set all MobileC threads to 'off' status.*
- EXPORTMC [int MC\\_Steer](#) (MCAgency\_t attr, int(\*funcptr)(void \*data), void \*arg)  
*Set up a steerable algorithm.*
- EXPORTMC enum [MC\\_SteerCommand\\_e](#) [MC\\_SteerControl](#) (void)  
*The MobileC user-algorithm steering function.*
- EXPORTMC [int MC\\_SyncDelete](#) (MCAgency\_t attr, int id)  
*Deletes a previously initialized synchronization variable.*
- EXPORTMC [int MC\\_SyncInit](#) (MCAgency\_t attr, int id)  
*Initializes a new MobileC synchronization variable.*
- EXPORTMC [int MC\\_TerminateAgent](#) (MCAgent\_t agent)  
*Halt a running agent.*
- [int MC\\_MainLoop](#) (MCAgency\_t attr)  
*Wait indefinitely.*
- EXPORTMC [int MC\\_WaitAgent](#) (MCAgency\_t attr)  
*Wait indefinitely.*
- EXPORTMC [MCAgent\\_t MC\\_WaitRetrieveAgent](#) (MCAgency\_t attr)  
*Wait and retrieve an agent.*
- EXPORTMC [int MC\\_WaitSignal](#) (MCAgency\_t attr, int signals)  
*Wait for a MobileC signal.*
- [int MC\\_AclDestroy\\_chdl](#) (void \*varg)
- void \* [MC\\_AclNew\\_chdl](#) (void \*varg)
- [int MC\\_AclPost\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclReply\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclRetrieve\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AclSend\\_chdl](#) (void \*varg)
- EXPORTCH void \* [MC\\_AclWaitRetrieve\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AclSetPerformative\\_chdl](#) (void \*varg)

- EXPORTCH [int MC\\_AclSetSender\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AclAddReceiver\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AclAddReplyTo\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AclSetContent\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AddAgent\\_chdl](#) (void \*varg)
- EXPORTCH [const void \\* MC\\_AgentVariableRetrieve\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_AgentVariableSave\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_CallAgentFunc\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_Barrier\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_BarrierDelete\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_BarrierInit\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_CondBroadcast\\_chdl](#) (void \*varg)
- EXPORTCH [MCAgent\\_t MC\\_ComposeAgent\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_CondSignal\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_CondReset\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_CondWait\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_DeleteAgent\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_DestroyServiceSearchResult\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_DeregisterService\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_End\\_chdl](#) (void \*varg)
- EXPORTCH [MCAgent\\_t MC\\_FindAgentByID\\_chdl](#) (void \*varg)
- EXPORTCH [MCAgent\\_t MC\\_FindAgentByName\\_chdl](#) (void \*varg)
- EXPORTCH [time\\_t MC\\_GetAgentArrivalTime\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_GetAgentID\\_chdl](#) (void \*varg)
- EXPORTCH [char \\* MC\\_GetAgentName\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_GetAgentNumTasks\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_GetAgentStatus\\_chdl](#) (void \*varg)
- EXPORTCH [char \\* MC\\_GetAgentXMLString\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_GetTimeOfDay\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_HaltAgency\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_MutexLock\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_MutexUnlock\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_PrintAgentCode\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_RegisterService\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_ResumeAgency\\_chdl](#) (void \*varg)
- EXPORTCH [MCAgent\\_t MC\\_RetrieveAgent\\_chdl](#) (void \*varg)
- EXPORTCH [char \\* MC\\_RetrieveAgentCode\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SearchForService\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SemaphorePost\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SemaphoreWait\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SendAgentMigrationMessage\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SendAgentMigrationMessageFile\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SendSteerCommand\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SetAgentStatus\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SetDefaultAgentStatus\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SyncDelete\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_SyncInit\\_chdl](#) (void \*varg)
- EXPORTCH [int MC\\_TerminateAgent\\_chdl](#) (void \*varg)



## Variables

- [mc\\_platform\\_p g\\_mc\\_platform](#)

### 8.47.1 Define Documentation

#### 8.47.1.1 #define HOST\_NAME\_MAX 255

Definition at line 61 of file libmc.c.

Referenced by MC\_Initialize().

### 8.47.2 Function Documentation

#### 8.47.2.1 int MC\_AclAddReceiver (fipa\_acl\_message\_t \* *acl*, const char \* *name*, const char \* *address*)

Definition at line 229 of file libmc.c.

References fipa\_agent\_identifier\_s::addresses, fipa\_agent\_identifier\_New(), fipa\_agent\_identifier\_set\_New(), fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_url\_New(), fipa\_url\_sequence\_New(), fipa\_agent\_identifier\_s::name, fipa\_url\_sequence\_s::num, fipa\_agent\_identifier\_set\_s::num, fipa\_acl\_message\_s::receiver, fipa\_acl\_message\_s::receiver\_num, fipa\_url\_s::str, and fipa\_url\_sequence\_s::urls.

Referenced by MC\_AclAddReceiver\_chdl().

#### 8.47.2.2 EXPORTCH int MC\_AclAddReceiver\_chdl (void \* *varg*)

Definition at line 1851 of file libmc.c.

References MC\_AclAddReceiver().

Referenced by agent\_RunChScriptThread().

#### 8.47.2.3 int MC\_AclAddReplyTo (fipa\_acl\_message\_t \* *acl*, const char \* *name*, const char \* *address*)

Definition at line 265 of file libmc.c.

References fipa\_agent\_identifier\_s::addresses, fipa\_agent\_identifier\_New(), fipa\_agent\_identifier\_set\_New(), fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_url\_New(), fipa\_url\_sequence\_New(), fipa\_agent\_identifier\_s::name, fipa\_url\_sequence\_s::num, fipa\_agent\_identifier\_set\_s::num, fipa\_acl\_message\_s::reply\_to, fipa\_url\_s::str, and fipa\_url\_sequence\_s::urls.

Referenced by MC\_AclAddReplyTo\_chdl().

#### 8.47.2.4 EXPORTCH int MC\_AclAddReplyTo\_chdl (void \* *varg*)

Definition at line 1871 of file libmc.c.

References MC\_AclAddReplyTo().

Referenced by agent\_RunChScriptThread().

**8.47.2.5 int MC\_AclDestroy (struct fipa\_acl\_message\_s \* *message*)**

Destroy a FIPA ACL message.

**Parameters:**

*message* The ACL message to destroy

**Returns:**

0 on success, error code on failure.

Definition at line 72 of file libmc.c.

References fipa\_acl\_message\_Destroy().

Referenced by MC\_AclDestroy\_chdl().

**8.47.2.6 int MC\_AclDestroy\_chdl (void \* *varg*)**

Definition at line 1698 of file libmc.c.

References MC\_AclDestroy().

Referenced by agent\_RunChScriptThread().

**8.47.2.7 EXPORTMC fipa\_acl\_message\_t\* MC\_AclNew (void) [read]**

Allocate a new ACL Message.

**Returns:**

A newly allocated and empty ACL message.

Definition at line 78 of file libmc.c.

References fipa\_acl\_message\_New().

Referenced by MC\_AclNew\_chdl().

**8.47.2.8 void\* MC\_AclNew\_chdl (void \* *varg*)**

Definition at line 1713 of file libmc.c.

References MC\_AclNew().

Referenced by agent\_RunChScriptThread().

**8.47.2.9 EXPORTMC int MC\_AclPost (MCAgent\_t *agent*, struct fipa\_acl\_message\_s \* *message*)**

Post ACL message to agent.

**Parameters:**

*agent* The agent to post the message to

*message* The message to post

**Returns:**

0 if successful, or `error_code_t` type.

Definition at line 83 of file `libmc.c`.

References `agent_mailbox_Post()`, and `agent_s::mailbox`.

Referenced by `MC_AclPost_chdl()`, and `MC_AclSend()`.

**8.47.2.10 int MC\_AclPost\_chdl (void \* *varg*)**

Definition at line 1721 of file `libmc.c`.

References `MC_AclPost()`.

Referenced by `agent_RunChScriptThread()`.

**8.47.2.11 EXPORTMC fipa\_acl\_message\_t\* MC\_AclReply (struct fipa\_acl\_message\_s \* *acl\_message*)** [read]

Reply to an ACL message.

**Parameters:**

*acl\_message* The incoming `acl_message` to reply to

**Returns:**

A newly allocated ACL message

**Note:**

This function simply generates a new ACL message with the 'receiver' field automatically set to the 'sender' field of the incoming message.

Definition at line 89 of file `libmc.c`.

References `fipa_Reply()`.

Referenced by `MC_AclReply_chdl()`.

**8.47.2.12 EXPORTCH void\* MC\_AclReply\_chdl (void \* *varg*)**

Definition at line 1739 of file `libmc.c`.

References `MC_AclReply()`.

Referenced by `agent_RunChScriptThread()`.

**8.47.2.13 EXPORTMC fipa\_acl\_message\_t\* MC\_AclRetrieve (MCAgent\_t *agent*)** [read]

Retrieve an ACL message.

**Parameters:**

*agent* Agent to retrieve message from.

**Returns:**

an ACL message struct on success or NULL on failure

Definition at line 95 of file libmc.c.

References agent\_mailbox\_Retrieve(), and agent\_s::mailbox.

Referenced by MC\_AclRetrieve\_chdl().

**8.47.2.14 EXPORTCH void\* MC\_AclRetrieve\_chdl (void \* varg)**

Definition at line 1755 of file libmc.c.

References MC\_AclRetrieve().

Referenced by agent\_RunChScriptThread().

**8.47.2.15 EXPORTMC int MC\_AclSend (MCAgency\_t attr, struct fipa\_acl\_message\_s \* acl)**

Send a composed ACL Message.

**Parameters:**

*attr* An initialized and running MobileC agency

*acl* An allocated and fully composed ACL message.

**Returns:**

0 if successful, error code on failure.

Definition at line 101 of file libmc.c.

References fipa\_agent\_identifier\_s::addresses, mtp\_http\_s::content, mtp\_http\_content\_s::content\_type, mtp\_http\_content\_s::data, dynstring\_Destroy(), FIPA\_ACL, fipa\_acl\_Compose(), fipa\_agent\_identifier\_set\_s::fipa\_agent\_identifiers, fipa\_envelope\_Compose(), mtp\_http\_s::host, http\_to\_hostport(), MC\_AclPost(), MC\_FindAgentByName(), dynstring\_s::message, message\_Destroy(), mtp\_http\_s::message\_parts, message\_Send(), message\_s::message\_type, mtp\_http\_CreateMessage(), mtp\_http\_Destroy(), mtp\_http\_New(), fipa\_agent\_identifier\_s::name, fipa\_url\_sequence\_s::num, fipa\_agent\_identifier\_set\_s::num, port, fipa\_acl\_message\_s::receiver, fipa\_url\_s::str, mtp\_http\_s::target, message\_s::target, and fipa\_url\_sequence\_s::urls.

Referenced by MC\_AclSend\_chdl().

**8.47.2.16 EXPORTCH int MC\_AclSend\_chdl (void \* varg)**

Definition at line 1771 of file libmc.c.

References CHECK\_NULL, MC\_AclSend(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

**8.47.2.17 int MC\_AclSetContent (fipa\_acl\_message\_t \* acl, const char \* content)**

Definition at line 300 of file libmc.c.

References `fipa_string_s::content`, `fipa_acl_message_s::content`, `fipa_string_Destroy()`, and `fipa_string_New()`.

Referenced by `MC_AclSetContent_chdl()`.

#### 8.47.2.18 EXPORTCH int MC\_AclSetContent\_chdl (void \* *varg*)

Definition at line 1891 of file `libmc.c`.

References `fipa_acl_message_s::content`, and `MC_AclSetContent()`.

Referenced by `agent_RunChScriptThread()`.

#### 8.47.2.19 int MC\_AclSetPerformative (fipa\_acl\_message\_t \* *acl*, enum fipa\_performative\_e *performative*)

Definition at line 198 of file `libmc.c`.

References `fipa_acl_message_s::performative`.

Referenced by `MC_AclSetPerformative_chdl()`.

#### 8.47.2.20 EXPORTCH int MC\_AclSetPerformative\_chdl (void \* *varg*)

Definition at line 1813 of file `libmc.c`.

References `MC_AclSetPerformative()`, and `fipa_acl_message_s::performative`.

Referenced by `agent_RunChScriptThread()`.

#### 8.47.2.21 int MC\_AclSetSender (fipa\_acl\_message\_t \* *acl*, const char \* *name*, const char \* *address*)

Definition at line 206 of file `libmc.c`.

References `fipa_agent_identifier_s::addresses`, `fipa_agent_identifier_Destroy()`, `fipa_agent_identifier_New()`, `fipa_url_New()`, `fipa_url_sequence_New()`, `fipa_agent_identifier_s::name`, `fipa_url_sequence_s::num`, `fipa_acl_message_s::sender`, `fipa_url_s::str`, and `fipa_url_sequence_s::urls`.

Referenced by `MC_AclSetSender_chdl()`.

#### 8.47.2.22 EXPORTCH int MC\_AclSetSender\_chdl (void \* *varg*)

Definition at line 1831 of file `libmc.c`.

References `MC_AclSetSender()`.

Referenced by `agent_RunChScriptThread()`.

#### 8.47.2.23 EXPORTMC fipa\_acl\_message\_t\* MC\_AclWaitRetrieve (MCAgent\_t *agent*) [read]

Wait for and retrieve an ACL message.

##### Parameters:

*agent* Agent to retrieve message from.

**Returns:**

an ACL message struct on success or NULL on failure

Definition at line 191 of file libmc.c.

References agent\_mailbox\_WaitRetrieve(), and agent\_s::mailbox.

Referenced by MC\_AclWaitRetrieve\_chdl().

**8.47.2.24 EXPORTCH void\* MC\_AclWaitRetrieve\_chdl (void \* *varg*)**

Definition at line 1795 of file libmc.c.

References MC\_AclWaitRetrieve().

Referenced by agent\_RunChScriptThread().

**8.47.2.25 EXPORTMC int MC\_AddAgent (MCAgency\_t *attr*, MCAgent\_t *agent*)**

Add an agent to the agency '*attr*'.

**Parameters:**

*attr* a MobileC agency

*agent* An initialized MobileC agent

**Returns:**

0 if successful, or error\_code\_t type

Definition at line 317 of file libmc.c.

References mc\_platform\_s::agent\_queue, mc\_platform\_s::ams, COND\_SIGNAL, agency\_s::mc\_platform, agent\_s::mc\_platform, MUTEX\_LOCK, and MUTEX\_UNLOCK.

Referenced by MC\_AddAgent\_chdl().

**8.47.2.26 EXPORTCH int MC\_AddAgent\_chdl (void \* *varg*)**

Definition at line 1911 of file libmc.c.

References CHECK\_NULL, MC\_AddAgent(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

**8.47.2.27 const void\* MC\_AgentVariableRetrieve (MCAgent\_t *agent*, const char \* *var\_name*, int *task\_num*)**

Retrieve a pointer to a previously saved variable.

**Parameters:**

*agent* A MobileC agent.

*var\_name* The name of the saved variable that has previously been saved.

*task\_num* The previous completed task from which to retrieve the saved variable.

**Returns:**

A pointer to the data on success or NULL on failure.

### 8.47.3 Examples

The following example demonstrates usage of [MC\\_AgentVariableRetrieve\(\)](#) from agent space.

Definition at line 330 of file libmc.c.

References `agent_task_s::agent_variable_list`, `interpreter_variable_data_s::data`, `agent_s::datastate`, `agent_datastate_s::task_progress`, and `agent_datastate_s::tasks`.

Referenced by `MC_AgentVariableRetrieve_chdl()`.

#### 8.47.3.1 EXPORTCH const void\* MC\_AgentVariableRetrieve\_chdl (void \* *varg*)

Definition at line 1935 of file libmc.c.

References `MC_AgentVariableRetrieve()`.

Referenced by `agent_RunChScriptThread()`.

#### 8.47.3.2 int MC\_AgentVariableSave (MCAgent\_t *agent*, const char \* *var\_name*)

Mark an agent variable for saving.

**Parameters:**

*agent* A MobileC agent.

*var\_name* The name of the variable to mark for saving.

**Returns:**

0 on success, non-zero on failure.

**See also:**

`test1.xml`

### 8.47.4 Examples

See `agent_saved_variables_example/test1.xml` for an example of usage of this api function.

### 8.47.5 Examples

The following example demonstrates usage of [MC\\_AgentVariableSave\(\)](#) from agent space.

Definition at line 348 of file libmc.c.

References `agent_s::datastate`, `MC_ERR_MEMORY`, `agent_task_s::num_saved_variables`, `agent_task_s::saved_variables`, `agent_datastate_s::task_progress`, and `agent_datastate_s::tasks`.

Referenced by `MC_AgentVariableSave_chdl()`.

#### 8.47.5.1 EXPORTCH int MC\_AgentVariableSave\_chdl (void \* *varg*)

Definition at line 1959 of file libmc.c.

References `MC_AgentVariableSave()`.

Referenced by `agent_RunChScriptThread()`.

#### 8.47.5.2 int MC\_Barrier (MCAgency\_t *attr*, int *id*)

Definition at line 369 of file libmc.c.

References `mc_platform_s::barrier_queue`, `barrier_queue_Get()`, `barrier_node_s::cond`, `COND_BROADCAST`, `COND_WAIT`, `barrier_node_s::lock`, `MC_ERR_NOT_FOUND`, `agency_s::mc_platform`, `MC_SUCCESS`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `node`, `barrier_node_s::num_registered`, and `barrier_node_s::num_waiting`.

Referenced by `MC_Barrier_chdl()`.

#### 8.47.5.3 EXPORTCH int MC\_Barrier\_chdl (void \* *varg*)

Definition at line 2011 of file libmc.c.

References `CHECK_NULL`, `MC_Barrier()`, and `agency_s::mc_platform`.

Referenced by `agent_RunChScriptThread()`.

#### 8.47.5.4 EXPORTMC int MC\_BarrierDelete (MCAgency\_t *attr*, int *id*)

Find and delete an initialized MobileC Barrier.

##### Parameters:

- attr* A running MobileC agency
- id* The id of the barrier node to delete

##### Returns:

returns 0 on success, error if the node is not found or other failure.

Definition at line 409 of file libmc.c.

References `mc_platform_s::barrier_queue`, `barrier_queue_Delete()`, and `agency_s::mc_platform`.

Referenced by `MC_BarrierDelete_chdl()`.

#### 8.47.5.5 EXPORTCH int MC\_BarrierDelete\_chdl (void \* *varg*)

Definition at line 2034 of file libmc.c.



References CHECK\_NULL, MC\_BarrierDelete(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

#### 8.47.5.6 EXPORTMC int MC\_BarrierInit (MCAgency\_t *attr*, int *id*, int *num\_procs*)

Initialize a MobileC Barrier.

##### Parameters:

*attr* A running MobileC agency

*id* The requested barrier id

*num\_procs* The number of agents/threads/processes that will wait on the barrier

##### Returns:

The allocated barrier id. May differ from the requested id if it is already in use.

### 8.47.6 Examples

The following example demonstrates an agent which sets up an MC\_Barrier.

Definition at line 395 of file libmc.c.

References barrier\_node\_Initialize(), mc\_platform\_s::barrier\_queue, barrier\_queue\_Add(), barrier\_queue\_Get(), MC\_ERR, agency\_s::mc\_platform, MC\_SUCCESS, and node.

Referenced by MC\_BarrierInit\_chdl().

#### 8.47.6.1 EXPORTCH int MC\_BarrierInit\_chdl (void \* *varg*)

Definition at line 2057 of file libmc.c.

References CHECK\_NULL, MC\_BarrierInit(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

#### 8.47.6.2 EXPORTMC int MC\_CallAgentFunc (MCAgent\_t *agent*, const char \* *funcName*, void \* *returnVal*, void \* *varg*)

Calls a function defined in an agent.

##### Parameters:

*agent* An initialized and executed MobileC agent

*funcName* The name of the function to call

*returnVal* The agent function's return value

*varg* The agent functions argument

##### Note:

The agent function must be of the form 'void\* func(void\* arg);'

**Returns:**

0 if successful, `error_code_t` type on failure

**8.47.7 Example**

Definition at line 415 of file `libmc.c`.

References `agent_s::agent_interp`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, and `agent_s::run_lock`.

Referenced by `MC_CallAgentFunc_chdl()`.

**8.47.7.1 EXPORTCH int MC\_CallAgentFunc\_chdl (void \* *varg*)**

Definition at line 1981 of file `libmc.c`.

References `MC_CallAgentFunc()`.

Referenced by `agent_RunChScriptThread()`.

**8.47.7.2 EXPORTMC int MC\_ChInitializeOptions (MCAgency\_t *attr*, ChOptions\_t \* *options*)**

Use custom `ChOptions_t` type for internal Ch interpreter.

**Parameters:**

*attr* A running MobileC agency

*options* Initialized Ch options structure

**Returns:**

0 on success, `error_code_t` type on failure

Definition at line 433 of file `libmc.c`.

References `mc_platform_s::interp_options`, and `agency_s::mc_platform`.

**8.47.7.3 MCAgent\_t MC\_ComposeAgent (const char \* *name*, const char \* *home*, const char \* *owner*, const char \* *code*, const char \* *return\_var\_name*, const char \* *server*, int *persistent*)**

Compose a new agent dynamically without using a prewritten XML file.

**Parameters:**

*name* The desired name of the new agent.

*home* The home of the new agent.

*owner* The owner of the new agent.

*code* The agent code

*return\_var\_name* The name of the agent's return variable. Set to "no-return" if no return variable is desired.

*server* The target destination of the agent.

***persistent*** A flag indicating whether or not the agent should be persistent. A value of '1' indicates persistence, while a value of '0' indicates default non-persistent behaviour.

#### Returns:

This function returns a valid MCAgent\_t structure on success or NULL on failure.

Definition at line 451 of file libmc.c.

References agent\_datastate\_s::agent\_code, agent\_datastate\_s::agent\_code\_ids, agent\_datastate\_s::agent\_codes, agent\_datastate\_New(), agent\_New(), agent\_s::agent\_status, agent\_task\_New(), agent\_s::agent\_type, agent\_s::datastate, agent\_s::home, MC\_LOCAL\_AGENT, MC\_WAIT\_MESSGSEND, agent\_s::name, agent\_datastate\_s::number\_of\_tasks, agent\_s::orphan, agent\_s::owner, agent\_datastate\_s::persistent, agent\_task\_s::server\_name, agent\_datastate\_s::tasks, and agent\_task\_s::var\_name.

Referenced by MC\_ComposeAgent\_chdl().

#### 8.47.7.4 EXPORTCH MCAgent\_t MC\_ComposeAgent\_chdl (void \* varg)

Definition at line 2105 of file libmc.c.

References MC\_ComposeAgent().

Referenced by agent\_RunChScriptThread().

#### 8.47.7.5 EXPORTMC int MC\_CondBroadcast (MCAgency\_t attr, int id)

Wakes up all agents/threads waiting on a condition variable.

#### Parameters:

***attr*** A MobileC agency

***id*** Synchronization variable id to broadcast to

#### See also:

[MC\\_SyncInit\(\)](#), [MC\\_CondSignal\(\)](#)

#### Returns:

0 on success, error\_code\_t type on failure

Definition at line 521 of file libmc.c.

References syncListNode\_s::cond, COND\_BROADCAST, syncListNode\_s::lock, MC\_ERR\_NOT\_FOUND, agency\_s::mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, syncListNode\_s::signalled, mc\_platform\_s::syncList, and syncListFind().

Referenced by MC\_CondBroadcast\_chdl().

#### 8.47.7.6 EXPORTCH int MC\_CondBroadcast\_chdl (void \* varg)

Definition at line 2082 of file libmc.c.

References CHECK\_NULL, MC\_CondBroadcast(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

#### 8.47.7.7 EXPORTMC int MC\_CondReset (MCAgency\_t *attr*, int *id*)

Reset a previously signalled MobileC condition variable.

##### Parameters:

*attr* A MobileC Agency

*id* The synchronization variable id to reset

##### See also:

[MC\\_SyncInit\(\)](#)

##### Returns:

0 on success, error\_code\_t type on failure

Definition at line 573 of file libmc.c.

References syncListNode\_s::lock, MC\_ERR\_NOT\_FOUND, agency\_s::mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, syncListNode\_s::signalled, mc\_platform\_s::syncList, and syncListFind().

Referenced by MC\_CondReset\_chdl().

#### 8.47.7.8 EXPORTCH int MC\_CondReset\_chdl (void \* *varg*)

Definition at line 2168 of file libmc.c.

References CHECK\_NULL, MC\_CondReset(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

#### 8.47.7.9 EXPORTMC int MC\_CondSignal (MCAgency\_t *attr*, int *id*)

Wakes up at least one thread waiting on a condition variable.

##### Parameters:

*attr* A MobileC agency

*id* synchronization variable id

##### See also:

[MC\\_SyncInit\(\)](#)

##### Returns:

0 on success, error\_code\_t type on failure

### 8.47.8 Example

The following example demonstrates the agent-space version of the function, which is nearly identical to the binary space api function.

Definition at line 536 of file libmc.c.

References `syncListNode_s::cond`, `COND_SIGNAL`, `syncListNode_s::lock`, `MC_ERR_NOT_FOUND`, `agency_s::mc_platform`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `syncListNode_s::signalled`, `mc_platform_s::syncList`, and `syncListFind()`.

Referenced by `MC_CondSignal_chdl()`.

#### 8.47.8.1 EXPORTCH int MC\_CondSignal\_chdl (void \* *varg*)

Definition at line 2145 of file libmc.c.

References `CHECK_NULL`, `MC_CondSignal()`, and `agency_s::mc_platform`.

Referenced by `agent_RunChScriptThread()`.

#### 8.47.8.2 EXPORTMC int MC\_CondWait (MCAgency\_t *attr*, int *id*)

Wait on a MobileC synchronization variable.

##### Parameters:

*attr* A MobileC agency

*id* a synchronization variable id

##### See also:

[MC\\_SyncInit\(\)](#)

##### Returns:

0 on success, `error_code_t` type on failure

### 8.47.9 Example

The following example demonstrates the agent-space version of this function.

Definition at line 551 of file libmc.c.

References `syncListNode_s::cond`, `COND_WAIT`, `syncListNode_s::lock`, `MC_ERR_NOT_FOUND`, `agency_s::mc_platform`, `MUTEX_LOCK`, `MUTEX_UNLOCK`, `syncListNode_s::signalled`, `mc_platform_s::syncList`, and `syncListFind()`.

Referenced by `MC_CondWait_chdl()`.

#### 8.47.9.1 EXPORTCH int MC\_CondWait\_chdl (void \* *varg*)

Definition at line 2191 of file libmc.c.

References `CHECK_NULL`, `MC_CondWait()`, and `agency_s::mc_platform`.

Referenced by `agent_RunChScriptThread()`.

**8.47.9.2 int MC\_CopyAgent (MCAgent\_t \* *agent\_out*, const MCAgent\_t *agent\_in*)**

Performs a deep-copy of an agent structure.

**Parameters:**

*agent\_out* A pointer to the agent to copy to.

*agent\_in* The agent to copy

**Returns:**

0 on success, error\_code\_t type on failure.

Definition at line 591 of file libmc.c.

References agent\_Copy(), and MC\_SUCCESS.

**8.47.9.3 EXPORTMC int MC\_DeleteAgent (MCAgent\_t *agent*)**

Stop and remove an agent.

**Parameters:**

*agent* An agent in any state (running, waiting, etc)

**Returns:**

0 on success, error\_code\_t type on failure

Definition at line 598 of file libmc.c.

References CHECK\_NULL, MC\_ERR\_INVALID, MC\_SetAgentStatus(), MC\_SUCCESS, MC\_TerminateAgent(), and MC\_WAIT\_FINISHED.

Referenced by MC\_DeleteAgent\_chdl().

**8.47.9.4 EXPORTCH int MC\_DeleteAgent\_chdl (void \* *varg*)**

Definition at line 2213 of file libmc.c.

References MC\_DeleteAgent().

Referenced by agent\_RunChScriptThread().

**8.47.9.5 int MC\_DeregisterService (MCAgency\_t *agency*, int *agentID*, const char \* *serviceName*)**

Definition at line 632 of file libmc.c.

References mc\_platform\_s::df, df\_AddRequest(), df\_request\_list\_node\_New(), and agency\_s::mc\_platform.

Referenced by MC\_DeregisterService\_chdl().

**8.47.9.6 EXPORTCH int MC\_DeregisterService\_chdl (void \* *varg*)**

Definition at line 2255 of file libmc.c.

References CHECK\_NULL, MC\_DeregisterService(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

**8.47.9.7 int MC\_DestroyServiceSearchResult (char \*\* *agentName*, char \*\* *serviceName*, int \* *agentID*, int *numResult*)**

Free memory allocated by a Service Search operation.

**Parameters:**

*agentName* agent names returned by a search operation.

*serviceName* service names return by a search operation.

*agentID* list of agent id's returned by a search operation.

*numResult* The number of hits returned by a search operation.

**Returns:**

0 on success, error code on failure.

Definition at line 612 of file libmc.c.

Referenced by MC\_DestroyServiceSearchResult\_chdl().

**8.47.9.8 EXPORTCH int MC\_DestroyServiceSearchResult\_chdl (void \* *varg*)**

Definition at line 2228 of file libmc.c.

References MC\_DestroyServiceSearchResult().

Referenced by agent\_RunChScriptThread().

**8.47.9.9 EXPORTMC int MC\_End (MCAgency\_t *attr*)**

End an agency.

**Parameters:**

*attr* A running agency

**Returns:**

0 on success, error\_code\_t type on failure

**8.47.10 Example**

Definition at line 661 of file libmc.c.

References mc\_platform\_s::acc, mc\_platform\_s::ams, mc\_platform\_s::cmd\_prompt, COND\_SIGNAL, mc\_platform\_s::connection\_queue, mc\_platform\_s::df, GET\_THREAD\_MODE, agency\_s::hostName, agency\_s::mc\_platform, mc\_platform\_Destroy(), MC\_THREAD\_ACC, MC\_THREAD\_AMS, MC\_THREAD\_CP, MC\_THREAD\_DF, MUTEX\_LOCK, MUTEX\_UNLOCK, mc\_platform\_s::quit, mc\_platform\_s::quit\_lock, cmd\_prompt\_s::thread, THREAD\_CANCEL, THREAD\_JOIN, and agency\_s::threads.

Referenced by MC\_End\_chdl().

#### 8.47.10.1 EXPORTCH int MC\_End\_chdl (void \* *varg*)

Definition at line 2282 of file libmc.c.

References CHECK\_NULL, MC\_End(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

#### 8.47.10.2 EXPORTMC MCAgent\_t MC\_FindAgentByID (MCAgency\_t *attr*, int *ID*)

Find an agent by its id.

##### Parameters:

*attr* the agency to search

*ID* the id to search for

##### Returns:

a valid agent on success, NULL on failure

Definition at line 727 of file libmc.c.

References mc\_platform\_s::agent\_queue, and agency\_s::mc\_platform.

Referenced by MC\_FindAgentByID\_chdl().

#### 8.47.10.3 EXPORTCH MCAgent\_t MC\_FindAgentByID\_chdl (void \* *varg*)

Definition at line 2299 of file libmc.c.

References CHECK\_NULL, MC\_FindAgentByID(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

#### 8.47.10.4 EXPORTMC MCAgent\_t MC\_FindAgentByName (MCAgency\_t *attr*, const char \* *name*)

Find an agent by its name.

##### Parameters:

*attr* a running agency

*name* name to search for



**Returns:**

a valid agent on success or NULL on failure

**8.47.11 Example**

Definition at line 714 of file libmc.c.

References mc\_platform\_s::agent\_queue, and agency\_s::mc\_platform.

Referenced by MC\_AclSend(), and MC\_FindAgentByName\_chdl().

**8.47.11.1 EXPORTCH MCAgent\_t MC\_FindAgentByName\_chdl (void \* varg)**

Definition at line 2322 of file libmc.c.

References CHECK\_NULL, MC\_FindAgentByName(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

**8.47.11.2 time\_t MC\_GetAgentArrivalTime (MCAgent\_t agent)**

Definition at line 743 of file libmc.c.

References agent\_s::arrival\_time.

Referenced by MC\_GetAgentArrivalTime\_chdl().

**8.47.11.3 EXPORTCH time\_t MC\_GetAgentArrivalTime\_chdl (void \* varg)**

Definition at line 2348 of file libmc.c.

References MC\_GetAgentArrivalTime().

**8.47.11.4 EXPORTMC void\* MC\_GetAgentExecEngine (MCAgent\_t agent)**

Retrieve an agent's Ch interpreter.

**Parameters:**

*agent* a valid agent

**Returns:**

a Ch interpreter of type 'ChInterp\_t' on success, or NULL on failure.

Definition at line 779 of file libmc.c.

References agent\_s::agent\_interp.

**8.47.11.5 EXPORTMC int MC\_GetAgentID (MCAgent\_t *agent*)**

Retrieve an agent's id.

Definition at line 785 of file libmc.c.

References agent\_s::id.

**8.47.11.6 EXPORTCH int MC\_GetAgentID\_chdl (void \* *varg*)**

Definition at line 2367 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.47.11.7 EXPORTMC char\* MC\_GetAgentName (MCAgent\_t *agent*)**

Definition at line 793 of file libmc.c.

References agent\_s::lock, MUTEX\_LOCK, MUTEX\_UNLOCK, and agent\_s::name.

**8.47.11.8 EXPORTCH char\* MC\_GetAgentName\_chdl (void \* *varg*)**

Definition at line 2383 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.47.11.9 EXPORTMC int MC\_GetAgentNumTasks (MCAgent\_t *agent*)**

Retrive the number of tasks an agent has.

**8.47.12 Example**

Definition at line 874 of file libmc.c.

References agent\_s::datastate, and agent\_datastate\_s::number\_of\_tasks.

**8.47.12.1 EXPORTCH int MC\_GetAgentNumTasks\_chdl (void \* *varg*)**

Definition at line 2399 of file libmc.c.

**8.47.12.2 EXPORTMC int MC\_GetAgentReturnData (MCAgent\_t *agent*, int *task\_num*, void \*\* *data*, int \* *dim*, int \*\* *extent*)**

Get an agent's return data.

**Parameters:**

*agent* a valid agent

*task\_num* the task for which to retrieve the return data. The task must already be completed.

*data* the return data. May be multi dimensional array.

*dim* the number of dimensions of the return array.

*extent* the extent of each one of the array dimensions.

### 8.47.13 Example

This file demonstrates the retrieval of agent return data from an agent

This is the agent which gets the data

Definition at line 811 of file libmc.c.

References agent\_task\_s::agent\_return\_data, interpreter\_variable\_data\_s::array\_dim, interpreter\_variable\_data\_s::array\_extent, CH\_DATATYPE\_SIZE, interpreter\_variable\_data\_s::data\_type, agent\_s::datastate, agent\_datastate\_s::number\_of\_tasks, size, and agent\_datastate\_s::tasks.

#### 8.47.13.1 EXPORTMC int MC\_GetAgentStatus (MCAgent\_t agent)

Get an agent's current status.

##### Returns:

returns type 'enum MC\_AgentStatus\_e'

Definition at line 757 of file libmc.c.

References agent\_s::agent\_status, agent\_s::lock, MUTEX\_LOCK, and MUTEX\_UNLOCK.

#### 8.47.13.2 EXPORTCH int MC\_GetAgentStatus\_chdl (void \*varg)

Definition at line 2415 of file libmc.c.

Referenced by agent\_RunChScriptThread().

#### 8.47.13.3 EXPORTMC enum MC\_AgentType\_e MC\_GetAgentType (MCAgent\_t agent)

Get an agent's type.

##### Returns:

returns type 'enum MC\_AgentType\_e'

Definition at line 880 of file libmc.c.

References agent\_s::agent\_type.

**8.47.13.4 EXPORTMC char\* MC\_GetAgentXMLString (MCAgent\_t agent)**

Get an agent's xml string.

**Returns:**

a malloc'd character string containing the agent's xml code

Definition at line 767 of file libmc.c.

References agent\_s::datastate, mxmlSaveAllocString(), and agent\_datastate\_s::xml\_agent\_root.

**8.47.13.5 EXPORTCH char\* MC\_GetAgentXMLString\_chdl (void \* varg)**

Definition at line 2431 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.47.13.6 int MC\_GetAllAgents (MCAgency\_t attr, MCAgent\_t \*\* agents, int \* num\_agents)**

Definition at line 890 of file libmc.c.

References mc\_platform\_s::agent\_queue, mc\_platform\_s::giant, mc\_platform\_s::giant\_lock, MC\_HaltAgency(), agency\_s::mc\_platform, MC\_ResumeAgency(), MUTEX\_LOCK, and MUTEX\_UNLOCK.

**8.47.13.7 EXPORTCH int MC\_GetTimeOfDay\_chdl (void \* varg)**

Definition at line 2447 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.47.13.8 EXPORTMC int MC\_HaltAgency (MCAgency\_t agency)**

Halt an agency: Do not process new entries in queues.

**Parameters:**

*agency* A handle to a running MobileC agency.

**Returns:**

0 on success, non-zero on failure.

Definition at line 926 of file libmc.c.

References mc\_platform\_s::giant, mc\_platform\_s::giant\_lock, agency\_s::mc\_platform, MUTEX\_LOCK, and MUTEX\_UNLOCK.

Referenced by MC\_GetAllAgents(), and MC\_HaltAgency\_chdl().

**8.47.13.9 EXPORTCH int MC\_HaltAgency\_chdl (void \* varg)**

Definition at line 2462 of file libmc.c.

References CHECK\_NULL, MC\_HaltAgency(), and agency\_s::mc\_platform.

Referenced by agent\_RunChScriptThread().

#### 8.47.13.10 EXPORTMC MCAgency\_t MC\_Initialize (int *port*, MCAgencyOptions\_t \* *options*)

Initialize and start a MobileC agency.

##### Parameters:

*port* the TCP port the agency should bind to

*options* initialized MobileC options or NULL for default options

##### Returns:

a handle to a running MobileC agency or NULL on failure

#### 8.47.14 Example

Definition at line 935 of file libmc.c.

References CHECK\_NULL, agency\_s::client, MCAgencyOptions\_s::default\_agent\_status, agency\_s::default\_agentstatus, MCAgencyOptions\_s::enable\_security, agency\_s::enable\_security, HOST\_NAME\_MAX, agency\_s::hostName, MC\_InitializeAgencyOptions(), agency\_s::mc\_platform, mc\_platform\_Initialize(), MC\_THREAD\_ALL, agency\_s::portno, agency\_s::server, MCAgencyOptions\_s::stack\_size, agency\_s::stack\_size, MCAgencyOptions\_s::threads, and agency\_s::threads.

#### 8.47.14.1 EXPORTMC int MC\_InitializeAgencyOptions (struct MCAgencyOptions\_s \* *options*)

Initialize MobileC options.

##### Parameters:

*options* options to initialize.

##### Returns:

0 on success, error\_code\_t on failure

##### Note:

MobileC options should be initialized with this function before any of its members are modified.

#### 8.47.15 Example

Definition at line 990 of file libmc.c.

References MCAgencyOptions\_s::default\_agent\_status, MCAgencyOptions\_s::enable\_security, MC\_THREAD\_ALL, MC\_WAIT\_CH, MCAgencyOptions\_s::modified, MCAgencyOptions\_s::stack\_size, and MCAgencyOptions\_s::threads.

Referenced by MC\_Initialize().

**8.47.15.1 EXPORTMC int MC\_LoadAgentFromFile (MCAgency\_t *attr*, const char \**filename*)**

Load an agent from a file into an agency.

**Parameters:**

*agency* A valid and running Mobile-C agency  
*filename* Filename containing the agent to load

**Returns:**

0 on success, non-zero on failure.

Definition at line 1007 of file libmc.c.

References `agency_s::mc_platform`, `message_Destroy()`, `message_InitializeFromString()`, `message_New()`, `mc_platform_s::message_queue`, `MXML_DESCEND`, `mxmlFindElement()`, `mxmlLoadString()`, `message_s::to_address`, `message_s::xml_payload`, and `message_s::xml_root`.

**8.47.15.2 int MC\_MainLoop (MCAgency\_t *attr*)**

Wait indefinitely.

**Note:**

This function is intended to block the calling thread forever.

Definition at line 1632 of file libmc.c.

References `mc_platform_s::ams`, and `agency_s::mc_platform`.

**8.47.15.3 EXPORTMC int MC\_MutexLock (MCAgency\_t *attr*, int *id*)**

Locks a MobileC synchronization variable as a mutex.

**Parameters:**

*attr* a MobileC agency handle  
*id* the synchronization variable id to lock

**Returns:**

0 on success, `error_code_t` type on failure

**8.47.16 Example**

Consider the following agents, which use the agent-space version of this api function. Note that the 'sleep' agent is sent first, followed by the 'wake' agent.

Definition at line 1077 of file libmc.c.

References `syncListNode_s::lock`, `agency_s::mc_platform`, `MUTEX_LOCK`, `mc_platform_s::syncList`, and `syncListFind()`.

**8.47.16.1 EXPORTCH int MC\_MutexLock\_chdl (void \* *varg*)**

Definition at line 2480 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.47.16.2 EXPORTMC int MC\_MutexUnlock (MCAgency\_t *attr*, int *id*)**

Definition at line 1089 of file libmc.c.

References syncListNode\_s::lock, agency\_s::mc\_platform, MUTEX\_UNLOCK, mc\_platform\_s::syncList, and syncListFind().

**8.47.16.3 EXPORTCH int MC\_MutexUnlock\_chdl (void \* *varg*)**

Definition at line 2504 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.47.16.4 EXPORTMC int MC\_PrintAgentCode (MCAgent\_t *agent*)**

Prints an agents code to stdout.

**Returns:**

0 on success, error\_code\_t on failure

Definition at line 1101 of file libmc.c.

References agent\_datastate\_s::agent\_code, agent\_s::datastate, agent\_s::lock, MUTEX\_LOCK, MUTEX\_UNLOCK, agent\_datastate\_s::number\_of\_tasks, and agent\_datastate\_s::task\_progress.

**8.47.16.5 EXPORTCH int MC\_PrintAgentCode\_chdl (void \* *varg*)**

Definition at line 2528 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.47.16.6 EXPORTMC int MC\_RegisterService (MCAgency\_t *agency*, MCAgent\_t *agent*, int *agentID*, const char \* *agentName*, char \*\* *serviceNames*, int *numServices*)**

Register a new service with the Directory Facilitator.

**Parameters:**

*agency* a MobileC agency handle

*agent* (OPTIONAL: See note) a MobileC agent

*agentID* (OPTIONAL: See note) a MobileC agent id

*agentName* (OPTIONAL: See note) a MobileC agent name

*serviceNames* an array of character strings of service names

*numServices* the number of services described in 'serviceNames'

**Returns:**

0 on success, error\_code\_t type on failure

**Note:**

Three of the input arguments are optional. The function expects as input the arguments 'agent XOR (agentID AND agentName)'.

### 8.47.17 Example

Definition at line 1117 of file libmc.c.

References CHECK\_NULL, mc\_platform\_s::df, df\_AddRequest(), df\_request\_list\_node\_New(), agent\_s::id, MC\_ERR\_INVALID\_ARGS, MC\_ERR\_MEMORY, agency\_s::mc\_platform, MUTEX\_INIT, MUTEX\_T, and agent\_s::name.

#### 8.47.17.1 EXPORTCH int MC\_RegisterService\_chdl (void \* *varg*)

Definition at line 2544 of file libmc.c.

Referenced by agent\_RunChScriptThread().

#### 8.47.17.2 EXPORTMC int MC\_ResetSignal (MCAgency\_t *attr*)

Reset a MobileC signal.

**Returns:**

0 on success, error\_code\_t on failure

**See also:**

[MC\\_WaitSignal\(\)](#)

Definition at line 1251 of file libmc.c.

References COND\_SIGNAL, mc\_platform\_s::giant, mc\_platform\_s::giant\_cond, mc\_platform\_s::giant\_lock, MC\_NO\_SIGNAL, agency\_s::mc\_platform, mc\_platform\_s::MC\_signal, MUTEX\_LOCK, and MUTEX\_UNLOCK.

#### 8.47.17.3 EXPORTMC int MC\_ResumeAgency (MCAgency\_t *agency*)

Resumes a halted agency.

**Parameters:**

*agency* An agency previously halted with the [MC\\_HaltAgency\(\)](#) function.

**Returns:**

0 on success, non-zero on failure.



Definition at line 1196 of file libmc.c.

References mc\_platform\_s::giant, mc\_platform\_s::giant\_lock, agency\_s::mc\_platform, MUTEX\_LOCK, and MUTEX\_UNLOCK.

Referenced by MC\_GetAllAgents().

#### 8.47.17.4 EXPORTCH int MC\_ResumeAgency\_chdl (void \* *varg*)

Definition at line 2581 of file libmc.c.

Referenced by agent\_RunChScriptThread().

#### 8.47.17.5 EXPORTMC MCAgent\_t MC\_RetrieveAgent (MCAgency\_t *attr*)

Retrieves the oldest agent from an agency.

##### Returns:

a valid agent or NULL on failure

Definition at line 1205 of file libmc.c.

References mc\_platform\_s::agent\_queue, agent\_s::agent\_status, ListSearch(), MC\_AGENT\_NEUTRAL, agency\_s::mc\_platform, MUTEX\_LOCK, and MUTEX\_UNLOCK.

#### 8.47.17.6 EXPORTCH MCAgent\_t MC\_RetrieveAgent\_chdl (void \* *varg*)

Definition at line 2599 of file libmc.c.

Referenced by agent\_RunChScriptThread().

#### 8.47.17.7 EXPORTMC char\* MC\_RetrieveAgentCode (MCAgent\_t *agent*)

Retrieves an agent's Ch code.

##### Returns:

a malloc'd character string on success, NULL on failure

Definition at line 1235 of file libmc.c.

References agent\_datastate\_s::agent\_code, agent\_s::datastate, agent\_s::lock, MUTEX\_LOCK, MUTEX\_UNLOCK, and agent\_datastate\_s::task\_progress.

#### 8.47.17.8 EXPORTCH char\* MC\_RetrieveAgentCode\_chdl (void \* *varg*)

Definition at line 2616 of file libmc.c.

Referenced by agent\_RunChScriptThread().

### 8.47.17.9 EXPORTMC int MC\_SearchForService (MCAgency\_t *attr*, const char \* *searchString*, char \*\*\* *agentNames*, char \*\*\* *serviceNames*, int \*\* *agentIDs*, int \* *numResults*)

Search the directory facilitator for a service.

#### Returns:

0 on success, error\_code\_t on failure

#### Parameters:

*attr* (input) a MobileC agency handle  
*searchString* (input) substring to search services for  
*agentNames* (return) array of agent names with matching services  
*serviceNames* (return) array of matching service names  
*agentIDs* (return) array of matching agent IDs  
*numResults* (return) number of matching results

### 8.47.18 Example

Definition at line 1262 of file libmc.c.

References CHECK\_NULL, COND\_SLEEP\_ACTION, mc\_platform\_s::df, df\_AddRequest(), df\_request\_list\_node\_Destroy(), df\_request\_list\_node\_New(), df\_request\_search\_Destroy(), df\_request\_search\_New(), MC\_ERR\_MEMORY, agency\_s::mc\_platform, MC\_SUCCESS, and search.

#### 8.47.18.1 EXPORTCH int MC\_SearchForService\_chdl (void \* *varg*)

Definition at line 2632 of file libmc.c.

Referenced by agent\_RunChScriptThread().

#### 8.47.18.2 EXPORTMC int MC\_SemaphorePost (MCAgency\_t *attr*, int *id*)

Post to a MobileC synchronization variable semaphore.

#### Parameters:

*attr* a MobileC agency handle  
*id* the synchronization variable id to post to

#### Returns:

0 on success, error\_code\_t type on failure

### 8.47.19 Example

Definition at line 1312 of file libmc.c.

References agency\_s::mc\_platform, syncListNode\_s::sem, SEMAPHORE\_POST, mc\_platform\_s::syncList, and syncListFind().

**8.47.19.1 EXPORTCH int MC\_SemaphorePost\_chdl (void \* *varg*)**

Definition at line 2673 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.47.19.2 EXPORTMC int MC\_SemaphoreWait (MCAgency\_t *attr*, int *id*)**

Decreases a MobileC synchronization variable semaphore count by one.

**Parameters:**

*attr* a MobileC agency handle

*id* synchronization variable id to wait on

**Returns:**

0 on MC\_SUCCESS, error\_code\_t type of failure

**Note:**

If the semaphore count is already zero, this function will block until another thread posts to the semaphore.

**8.47.20 Example**

Definition at line 1324 of file libmc.c.

References agency\_s::mc\_platform, syncListNode\_s::sem, SEMAPHORE\_WAIT, mc\_platform\_s::syncList, and syncListFind().

**8.47.20.1 EXPORTCH int MC\_SemaphoreWait\_chdl (void \* *varg*)**

Definition at line 2697 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.47.20.2 EXPORTMC int MC\_SendAgentMigrationMessage (MCAgency\_t *attr*, const char \* *message*, const char \* *hostname*, int *port*)**

Sends an agent migration message.

**Parameters:**

*attr* a MobileC agency handle

*message* a valid MobileC xml agent migration message

*hostname* host to send the message to

*port* port to send the message to

Definition at line 1346 of file libmc.c.

References MC\_ERR, agency\_s::mc\_platform, message\_Destroy(), message\_InitializeFromString(), message\_New(), and mc\_platform\_s::message\_queue.

**8.47.20.3 EXPORTCH int MC\_SendAgentMigrationMessage\_chdl (void \* *varg*)**

Definition at line 2721 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.47.20.4 EXPORTMC int MC\_SendAgentMigrationMessageFile (MCAgency\_t *attr*, const char \* *filename*, const char \* *hostname*, int *port*)**

Sends an agent migration message.

**Parameters:**

*attr* a MobileC agency handle

*filename* file containing a valid MobileC xml agent migration message

*hostname* hostname to send the agent to

*port* port to send the agent to

Definition at line 1377 of file libmc.c.

References agency\_s::mc\_platform, message\_Destroy(), message\_InitializeFromString(), message\_New(), and mc\_platform\_s::message\_queue.

**8.47.20.5 EXPORTCH int MC\_SendAgentMigrationMessageFile\_chdl (void \* *varg*)**

Definition at line 2746 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.47.20.6 int MC\_SendCh (MCAgency\_t *attr*, const char \* *filename*, const char \* *remotehost*, int *port*)**

Definition at line 1336 of file libmc.c.

**8.47.20.7 EXPORTMC int MC\_SendSteerCommand (MCAgency\_t *attr*, enum MC\_SteerCommand\_e *cmd*)**

Definition at line 1453 of file libmc.c.

References COND\_BROADCAST, agency\_s::mc\_platform, mc\_platform\_s::MC\_steer\_command, mc\_platform\_s::MC\_steer\_cond, mc\_platform\_s::MC\_steer\_lock, MUTEX\_LOCK, and MUTEX\_UNLOCK.

**8.47.20.8 EXPORTCH int MC\_SendSteerCommand\_chdl (void \* *varg*)**

Definition at line 2766 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.47.20.9 int MC\_SetAgentStatus (MCAgent\_t *agent*, int *status*)**

Set an agent's status.

**Parameters:**

*agent* a MobileC agent

*status* agent status of type 'enum MC\_AgentStatus\_e'

**Returns:**

0 on success, or error\_code\_t on failure

Definition at line 1463 of file libmc.c.

References agent\_s::agent\_status, mc\_platform\_s::ams, COND\_SIGNAL, agent\_s::lock, agent\_s::mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, and agent\_s::orphan.

Referenced by MC\_DeleteAgent().

**8.47.20.10 EXPORTCH int MC\_SetAgentStatus\_chdl (void \* *varg*)**

Definition at line 2789 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.47.20.11 int MC\_SetDefaultAgentStatus (MCAgency\_t *agency*, enum MC\_AgentStatus\_e *status*)**

Sets default incoming agent status.

**Parameters:**

*agency* a MobileC agency handle

*status* the status to set all incoming agents

**Returns:**

0 on success, error\_type\_t on failure

**Note:**

using this function will override any status the incoming agent attempts to set for itself.

Definition at line 1478 of file libmc.c.

References mc\_platform\_s::default\_agentstatus, and agency\_s::mc\_platform.

**8.47.20.12 EXPORTCH int MC\_SetDefaultAgentStatus\_chdl (void \* *varg*)**

Definition at line 2807 of file libmc.c.

Referenced by agent\_RunChScriptThread().

**8.47.20.13 EXPORTMC int MC\_SetThreadOff (MCAgencyOptions\_t \* *options*, enum MC\_ThreadIndex\_e *index*)**

Sets a MobileC thread to "off" status.

**Parameters:**

*options* MobileC options previously initialized with [MC\\_InitializeAgencyOptions\(\)](#)

*index* the thread to set

**Returns:**

0 on success, error\_code\_t on failure

**Note:**

This function must be called before [MC\\_Initialize\(\)](#). Once an agency is started with MC\_Initialize, the MC\_SetThread functions will have no effect.

Definition at line 1505 of file libmc.c.

References SET\_THREAD\_OFF, and MCAgencyOptions\_s::threads.

**8.47.20.14 EXPORTMC int MC\_SetThreadOn (MCAgencyOptions\_t \* *options*, enum MC\_ThreadIndex\_e *index*)**

Sets a MobileC thread to "on" status.

**Parameters:**

*options* MobileC options previously initialized with [MC\\_InitializeAgencyOptions\(\)](#)

*index* the thread to set

**Returns:**

0 on success, error\_code\_t on failure

**Note:**

This function must be called before [MC\\_Initialize\(\)](#). Once an agency is started with MC\_Initialize, the MC\_SetThread functions will have no effect.

Definition at line 1488 of file libmc.c.

References SET\_THREAD\_ON, and MCAgencyOptions\_s::threads.

**8.47.20.15 EXPORTMC int MC\_SetThreadsAllOff (MCAgencyOptions\_t \* *options*)**

Set all MobileC threads to 'off' status.

**Parameters:**

*options* a MobileC options structure initialized with with the [MC\\_InitializeAgencyOptions\(\)](#) function.

**Returns:**

0 on success, error code on failure.

Definition at line 1512 of file libmc.c.

References MC\_THREAD\_ALL, SET\_THREAD\_OFF, and MCAgencyOptions\_s::threads.

**8.47.20.16 EXPORTMC int MC\_SetThreadsAllOn (MCAgencyOptions\_t \* *options*)**

Set all Mobile-C threads on.

**Parameters:**

*options* MobileC options structure, initialized with [MC\\_InitializeAgencyOptions\(\)](#)

**Returns:**

0 on success, error code on failure.

Definition at line 1495 of file libmc.c.

References `MC_THREAD_ALL`, `SET_THREAD_ON`, and `MCAgencyOptions_s::threads`.

**8.47.20.17 EXPORTMC int MC\_Steer (MCAgency\_t *attr*, int(\*) (void \**data*) *funcptr*, void \* *arg*)**

Set up a steerable algorithm.

**Parameters:**

*attr* a MobileC agency handle

*funcptr* a function pointer to the algorithm

*arg* an argument for the algorithm function

**Returns:**

0 on success, `error_code_t` on failure

**Note:**

The algorithm function must contain a call to `MC_SteerControl` in order for the algorithm to be steerable.

**8.47.21 Example**

Definition at line 1522 of file libmc.c.

References `agency_s::mc_platform`, `MC_RESTART`, `MC_RUN`, `mc_platform_s::MC_steer_command`, `mc_platform_s::MC_steer_lock`, `MUTEX_LOCK`, and `MUTEX_UNLOCK`.

**8.47.21.1 EXPORTMC enum MC\_SteerCommand\_e MC\_SteerControl (void)**

The MobileC user-algorithm steering function.

**Returns:**

The current steering command

**Note:**

This function belongs inside a user's steerable algorithm.

See also:

[MC\\_Steer\(\)](#)

Definition at line 1541 of file libmc.c.

References COND\_WAIT, mc\_platform\_s::MC\_steer\_command, mc\_platform\_s::MC\_steer\_cond, mc\_platform\_s::MC\_steer\_lock, MC\_SUSPEND, MUTEX\_LOCK, and MUTEX\_UNLOCK.

#### 8.47.21.2 EXPORTMC int MC\_SyncDelete (MCAgency\_t attr, int id)

Deletes a previously initialized synchronization variable.

**Parameters:**

*attr* a MobileC agency handle

*id* the sync variable id to delete

**Returns:**

0 on success, or error\_code\_t on failure

Definition at line 1557 of file libmc.c.

References syncList\_s::giant\_lock, syncListNode\_s::lock, MC\_ERR\_NOT\_FOUND, agency\_s::mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, mc\_platform\_s::syncList, syncListFind(), syncListNodeDestroy(), and syncListRemove().

#### 8.47.21.3 EXPORTCH int MC\_SyncDelete\_chdl (void \* varg)

Definition at line 2830 of file libmc.c.

Referenced by agent\_RunChScriptThread().

#### 8.47.21.4 EXPORTMC int MC\_SyncInit (MCAgency\_t attr, int id)

Initializes a new MobileC synchronization variable.

**Parameters:**

*attr* a MobileC agency handle

*id* the requested sync variable id

**Returns:**

new sync variable's id. May be different than the requested id.

**Note:**

Each synchronization variable may be used as a mutex, condition variable, or semaphore. However, it should only be used as one type of synchronization variable per instance, or undefined behaviour may result.

See also:

[MC\\_MutexLock\(\)](#), [MC\\_MutexUnlock\(\)](#), [MC\\_CondWait\(\)](#), [MC\\_CondSignal\(\)](#), [MC\\_CondBroadcast](#), [MC\\_SemaphorePost\(\)](#), [MC\\_SemaphoreWait\(\)](#)



Definition at line 1587 of file libmc.c.

References syncList\_s::giant\_lock, syncListNode\_s::id, agency\_s::mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, node, mc\_platform\_s::syncList, syncListAddNode(), syncListFind(), and syncListNodeNew().

#### 8.47.21.5 EXPORTCH int MC\_SyncInit\_chdl (void \* *varg*)

Definition at line 2853 of file libmc.c.

Referenced by agent\_RunChScriptThread().

#### 8.47.21.6 EXPORTMC int MC\_TerminateAgent (MCAgent\_t *agent*)

Halt a running agent.

##### Returns:

0 on success, error\_code\_t on failure

Definition at line 1612 of file libmc.c.

References agent\_s::agent\_interp.

Referenced by MC\_DeleteAgent().

#### 8.47.21.7 EXPORTCH int MC\_TerminateAgent\_chdl (void \* *varg*)

Definition at line 2877 of file libmc.c.

Referenced by agent\_RunChScriptThread().

#### 8.47.21.8 EXPORTMC int MC\_WaitAgent (MCAgency\_t *attr*)

Wait indefinitely.

##### Note:

This function is intended to block the calling thread forever. Wait for an agent arrival event  
This function blocks until an agent arrival signal is triggered, at which point it unblocks.

Definition at line 1639 of file libmc.c.

References mc\_platform\_s::agent\_queue, COND\_WAIT, mc\_platform, agency\_s::mc\_platform, MUTEX\_LOCK, MUTEX\_UNLOCK, and size.

#### 8.47.21.9 EXPORTMC MCAgent\_t MC\_WaitRetrieveAgent (MCAgency\_t *attr*)

Wait and retrieve an agent.

##### Returns:

a valid MobileC agent on success, or NULL on failure

**Note:**

This function blocks until the arrival of an agent. The agent is retrieved after it is initialized, but before it is executed.

Definition at line 1659 of file libmc.c.

References `mc_platform_s::agent_queue`, `ListSearch()`, `agency_s::mc_platform`, `MC_RECV_AGENT`, `MC_WaitSignal()`, `MUTEX_LOCK`, and `MUTEX_UNLOCK`.

**8.47.21.10 EXPORTMC int MC\_WaitSignal (MCAgency\_t *attr*, int *signals*)**

Wait for a MobileC signal.

**Parameters:**

*attr* a MobileC agency handle

*signals* a flag of signals to wait for, of type 'enum MC\_Signal\_e'

**Returns:**

0 on success, `error_code_t` on failure

**Note:**

the parameter 'signals' may be something like 'MC\_RECV\_MESSAGE | MC\_RECV\_AGENT', etc.

**8.47.22 Example**

Definition at line 1677 of file libmc.c.

References `COND_WAIT`, `mc_platform_s::giant`, `mc_platform_s::giant_lock`, `agency_s::mc_platform`, `mc_platform_s::MC_signal`, `mc_platform_s::MC_signal_cond`, `mc_platform_s::MC_signal_lock`, `MUTEX_LOCK`, and `MUTEX_UNLOCK`.

Referenced by `MC_WaitRetrieveAgent()`.

**8.47.23 Variable Documentation****8.47.23.1 mc\_platform\_p g\_mc\_platform**

Definition at line 65 of file libmc.c.

Referenced by `fipa_envelope_Compose__from()`.

## 8.48 /home/dko/projects/mobilec/trunk/src/mc\_list/list.c File Reference

```
#include "list.h"
#include <stdio.h>
#include <stdlib.h>
```

### Functions

- [list\\_p ListInitialize](#) (void)
- void [ListTerminate](#) ([list\\_p](#) list)
- [int list\\_pGetSize](#) ([list\\_p](#) list)
- DATA [ListGetHead](#) ([list\\_p](#) list)
- DATA [ListPop](#) ([list\\_p](#) list)
- DATA [ListSearch](#) ([list\\_p](#) list, const [int](#) index)
- [int ListAdd](#) ([list\\_p](#) list, DATA data)
- [int ListInsert](#) ([list\\_p](#) list, DATA data, const [int](#) index)
- DATA [ListDelete](#) ([list\\_p](#) list, const [int](#) index)

### 8.48.1 Function Documentation

#### 8.48.1.1 [int list\\_pGetSize](#) ([list\\_p](#) *list*)

Definition at line 73 of file list.c.

References [list\\_s::size](#).

#### 8.48.1.2 [int ListAdd](#) ([list\\_p](#) *list*, DATA *data*)

Definition at line 161 of file list.c.

References [list\\_s::listhead](#), [listNode\\_s::next](#), [listNode\\_s::node\\_data](#), and [list\\_s::size](#).

Referenced by [barrier\\_queue\\_Add\(\)](#), [df\\_Add\(\)](#), [df\\_AddRequest\(\)](#), and [syncListAddNode\(\)](#).

#### 8.48.1.3 [DATA ListDelete](#) ([list\\_p](#) *list*, const [int](#) *index*)

Definition at line 216 of file list.c.

References [DATA](#), [list\\_s::listhead](#), [listNode\\_s::next](#), [listNode\\_s::node\\_data](#), and [list\\_s::size](#).

Referenced by [barrier\\_queue\\_Delete\(\)](#), [syncListDelete\(\)](#), and [syncListRemove\(\)](#).

#### 8.48.1.4 [DATA ListGetHead](#) ([list\\_p](#) *list*)

Definition at line 79 of file list.c.

References [list\\_s::listhead](#), and [listNode\\_s::node\\_data](#).

#### 8.48.1.5 **list\_p ListInitialize (void)**

Definition at line 33 of file list.c.

References list\_s::listhead, and list\_s::size.

Referenced by barrier\_queue\_New(), df\_Initialize(), df\_request\_list\_New(), and syncListInit().

#### 8.48.1.6 **int ListInsert (list\_p list, DATA data, const int index)**

Definition at line 203 of file list.c.

#### 8.48.1.7 **DATA ListPop (list\_p list)**

Definition at line 89 of file list.c.

References DATA, list\_s::listhead, listNode\_s::next, listNode\_s::node\_data, and list\_s::size.

Referenced by barrier\_queue\_Pop(), df\_Destroy(), df\_request\_list\_Destroy(), and df\_request\_list\_Pop().

#### 8.48.1.8 **DATA ListSearch (list\_p list, const int index)**

Definition at line 124 of file list.c.

References list\_s::listhead, listNode\_s::next, listNode\_s::node\_data, and list\_s::size.

Referenced by ams\_ManageAgentList(), ams\_Print(), AP\_QUEUE\_SEARCH\_TEMPLATE(), AP\_QUEUE\_STD\_DEFN\_TEMPLATE(), barrier\_queue\_Delete(), MC\_RetrieveAgent(), MC\_WaitRetrieveAgent(), message\_queue\_SendOutgoing(), syncListDelete(), and syncListRemove().

#### 8.48.1.9 **void ListTerminate (list\_p list)**

Definition at line 49 of file list.c.

References list\_s::listhead, and list\_s::size.

Referenced by barrier\_queue\_Destroy(), df\_Destroy(), and df\_request\_list\_Destroy().

## 8.49 /home/dko/projects/mobilec/trunk/src/mc\_list/list.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
```

### Data Structures

- struct [listNode\\_s](#)
- struct [list\\_s](#)

### Defines

- #define [DATA](#) void\*
- #define [QUEUE\\_TEMPLATE](#)(name, node\_type, search\_type, search\_var\_name)

### Typedefs

- typedef struct [listNode\\_s](#) [listNode\\_t](#)
- typedef [listNode\\_t](#) \* [listNode\\_p](#)
- typedef struct [list\\_s](#) [list\\_t](#)
- typedef [list\\_t](#) \* [list\\_p](#)

### Functions

- [list\\_p](#) [ListInitialize](#) (void)
- void [ListTerminate](#) ([list\\_p](#) list)
- int [ListGetSize](#) ([list\\_p](#) list)
- int [ListAdd](#) ([list\\_p](#) list, [DATA](#) data)
- int [ListInsert](#) ([list\\_p](#) list, [DATA](#) data, const int index)
- [DATA](#) [ListGetHead](#) ([list\\_p](#) list)
- [DATA](#) [ListPop](#) ([list\\_p](#) list)
- [DATA](#) [ListSearch](#) ([list\\_p](#) list, const int index)
- [DATA](#) [ListDelete](#) ([list\\_p](#) list, const int index)

#### 8.49.1 Define Documentation

##### 8.49.1.1 #define [DATA](#) void\*

Definition at line 30 of file list.h.

Referenced by [barrier\\_queue\\_Add\(\)](#), [ListDelete\(\)](#), [ListPop\(\)](#), and [syncListAddNode\(\)](#).

##### 8.49.1.2 #define [QUEUE\\_TEMPLATE](#)(name, node\_type, search\_type, search\_var\_name)

Definition at line 68 of file list.h.

## 8.49.2 Typedef Documentation

### 8.49.2.1 typedef list\_t\* list\_p

Definition at line 49 of file list.h.

### 8.49.2.2 typedef struct list\_s list\_t

### 8.49.2.3 typedef listNode\_t\* listNode\_p

Definition at line 39 of file list.h.

### 8.49.2.4 typedef struct listNode\_s listNode\_t

## 8.49.3 Function Documentation

### 8.49.3.1 int ListAdd (list\_p *list*, DATA *data*)

Definition at line 161 of file list.c.

References list\_s::listhead, listNode\_s::next, listNode\_s::node\_data, and list\_s::size.

Referenced by barrier\_queue\_Add(), df\_Add(), df\_AddRequest(), and syncListAddNode().

### 8.49.3.2 DATA ListDelete (list\_p *list*, const int *index*)

Definition at line 216 of file list.c.

References DATA, list\_s::listhead, listNode\_s::next, listNode\_s::node\_data, and list\_s::size.

Referenced by barrier\_queue\_Delete(), syncListDelete(), and syncListRemove().

### 8.49.3.3 DATA ListGetHead (list\_p *list*)

Definition at line 79 of file list.c.

References list\_s::listhead, and listNode\_s::node\_data.

### 8.49.3.4 int ListGetSize (list\_p *list*)

### 8.49.3.5 list\_p ListInitialize (void)

Definition at line 33 of file list.c.

References list\_s::listhead, and list\_s::size.

Referenced by barrier\_queue\_New(), df\_Initialize(), df\_request\_list\_New(), and syncListInit().

### 8.49.3.6 int ListInsert (list\_p *list*, DATA *data*, const int *index*)

Definition at line 203 of file list.c.

#### 8.49.3.7 DATA ListPop (*list\_p list*)

Definition at line 89 of file list.c.

References DATA, list\_s::listhead, listNode\_s::next, listNode\_s::node\_data, and list\_s::size.

Referenced by barrier\_queue\_Pop(), df\_Destroy(), df\_request\_list\_Destroy(), and df\_request\_list\_Pop().

#### 8.49.3.8 DATA ListSearch (*list\_p list*, *const int index*)

Definition at line 124 of file list.c.

References list\_s::listhead, listNode\_s::next, listNode\_s::node\_data, and list\_s::size.

Referenced by ams\_ManageAgentList(), ams\_Print(), AP\_QUEUE\_SEARCH\_TEMPLATE(), AP\_QUEUE\_STD\_DEFN\_TEMPLATE(), barrier\_queue\_Delete(), MC\_RetrieveAgent(), MC\_WaitRetrieveAgent(), message\_queue\_SendOutgoing(), syncListDelete(), and syncListRemove().

#### 8.49.3.9 void ListTerminate (*list\_p list*)

Definition at line 49 of file list.c.

References list\_s::listhead, and list\_s::size.

Referenced by barrier\_queue\_Destroy(), df\_Destroy(), and df\_request\_list\_Destroy().

## 8.50 /home/dko/projects/mobilec/trunk/src/mc\_platform.c File Reference

```
#include <unistd.h>
#include <sys/socket.h>
#include <netdb.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include "include/acc.h"
#include "include/mc_platform.h"
```

### Defines

- #define [DEFAULT\\_HOSTNAME\\_LENGTH](#) 200

### Functions

- [mc\\_platform\\_p mc\\_platform\\_Initialize](#) ([MCAgency\\_t](#) agency)
- [int mc\\_platform\\_Destroy](#) ([mc\\_platform\\_p](#) platform)

#### 8.50.1 Define Documentation

##### 8.50.1.1 #define DEFAULT\_HOSTNAME\_LENGTH 200

Definition at line 45 of file mc\_platform.c.

Referenced by [mc\\_platform\\_Initialize\(\)](#).

#### 8.50.2 Function Documentation

##### 8.50.2.1 int mc\_platform\_Destroy (mc\_platform\_p platform)

Definition at line 235 of file mc\_platform.c.

References [mc\\_platform\\_s::acc](#), [acc\\_Destroy\(\)](#), [mc\\_platform\\_s::agent\\_queue](#), [mc\\_platform\\_s::ams](#), [ams\\_Destroy\(\)](#), [mc\\_platform\\_s::asm\\_message\\_queue](#), [mc\\_platform\\_s::barrier\\_queue](#), [barrier\\_queue\\_Destroy\(\)](#), [mc\\_platform\\_s::cmd\\_prompt](#), [cmd\\_prompt\\_Destroy\(\)](#), [COND\\_DESTROY](#), [mc\\_platform\\_s::connection\\_queue](#), [mc\\_platform\\_s::df](#), [df\\_Destroy\(\)](#), [mc\\_platform\\_s::giant\\_cond](#), [mc\\_platform\\_s::giant\\_lock](#), [mc\\_platform\\_s::interp\\_options](#), [mc\\_platform\\_s::MC\\_signal\\_cond](#), [mc\\_platform\\_s::MC\\_signal\\_lock](#), [mc\\_platform\\_s::MC\\_steer\\_cond](#), [mc\\_platform\\_s::MC\\_steer\\_lock](#), [MC\\_SUCCESS](#), [mc\\_platform\\_s::MC\\_sync\\_cond](#), [mc\\_platform\\_s::MC\\_sync\\_lock](#), [mc\\_platform\\_s::message\\_queue](#), [MUTEX\\_DESTROY](#), and [mc\\_platform\\_s::quit\\_lock](#).

Referenced by [handler\\_QUIT\(\)](#), and [MC\\_End\(\)](#).

##### 8.50.2.2 mc\_platform\_p mc\_platform\_Initialize (MCAgency\_t agency)

Definition at line 47 of file mc\_platform.c.



References `mc_platform_s::acc`, `acc_Initialize()`, `acc_Start()`, `mc_platform_s::agent_queue`, `mc_platform_s::ams`, `ams_Initialize()`, `ams_Start()`, `mc_platform_s::asm_message_queue`, `mc_platform_s::barrier_queue`, `barrier_queue_New()`, `CHECK_NULL`, `mc_platform_s::cmd_prompt`, `cmd_prompt_Initialize()`, `cmd_prompt_Start()`, `COND_INIT`, `COND_T`, `COND_WAIT`, `mc_platform_s::connection_queue`, `agency_s::default_agentstatus`, `mc_platform_s::default_agentstatus`, `DEFAULT_HOSTNAME_LENGTH`, `mc_platform_s::df`, `df_Initialize()`, `df_Start()`, `agency_s::enable_security`, `mc_platform_s::err`, `GET_THREAD_MODE`, `mc_platform_s::giant`, `mc_platform_s::giant_cond`, `mc_platform_s::giant_lock`, `mc_platform_s::hostname`, `mc_platform_s::interp_options`, `agency_s::last_error`, `MC_ERR_MEMORY`, `MC_NO_SIGNAL`, `mc_platform`, `mc_platform_s::MC_signal`, `mc_platform_s::MC_signal_cond`, `mc_platform_s::MC_signal_lock`, `mc_platform_s::MC_steer_cond`, `mc_platform_s::MC_steer_lock`, `mc_platform_s::MC_sync_cond`, `mc_platform_s::MC_sync_lock`, `MC_THREAD_ACC`, `MC_THREAD_ALL`, `MC_THREAD_AMS`, `MC_THREAD_CP`, `MC_THREAD_DF`, `mc_platform_s::message_queue`, `MUTEX_INIT`, `MUTEX_LOCK`, `MUTEX_T`, `MUTEX_UNLOCK`, `mc_platform_s::port`, `agency_s::portno`, `mc_platform_s::quit`, `mc_platform_s::quit_lock`, `agency_s::stack_size`, `mc_platform_s::stack_size`, `mc_platform_s::syncList`, `syncListInit()`, and `agency_s::threads`.

Referenced by `MC_Initialize()`.

## 8.51 /home/dko/projects/mobilec/trunk/src/mc\_rwlock.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "include/macros.h"
#include "include/mc_error.h"
#include "include/mc_rwlock.h"
```

### Functions

- [int mc\\_rwlock\\_init \(mc\\_rwlock\\_p rwlock\)](#)
- [int mc\\_rwlock\\_destroy \(mc\\_rwlock\\_p rwlock\)](#)
- [int mc\\_rwlock\\_rdlock \(mc\\_rwlock\\_p rwlock\)](#)
- [int mc\\_rwlock\\_rdunlock \(mc\\_rwlock\\_p rwlock\)](#)
- [int mc\\_rwlock\\_wrlock \(mc\\_rwlock\\_p rwlock\)](#)
- [int mc\\_rwlock\\_wrunlock \(mc\\_rwlock\\_p rwlock\)](#)

### 8.51.1 Function Documentation

#### 8.51.1.1 int mc\_rwlock\_destroy (mc\_rwlock\_p *rwlock*)

Definition at line 61 of file mc\_rwlock.c.

References mc\_rwlock\_s::cond, COND\_DESTROY, mc\_rwlock\_s::lock, and MUTEX\_DESTROY.

#### 8.51.1.2 int mc\_rwlock\_init (mc\_rwlock\_p *rwlock*)

Definition at line 41 of file mc\_rwlock.c.

References CHECK\_NULL, mc\_rwlock\_s::cond, COND\_INIT, COND\_T, mc\_rwlock\_s::lock, MC\_ERR\_MEMORY, MUTEX\_INIT, MUTEX\_T, mc\_rwlock\_s::num\_readers, mc\_rwlock\_s::write\_flag, and mc\_rwlock\_s::write\_request.

#### 8.51.1.3 int mc\_rwlock\_rdlock (mc\_rwlock\_p *rwlock*)

Definition at line 74 of file mc\_rwlock.c.

References mc\_rwlock\_s::cond, COND\_WAIT, mc\_rwlock\_s::lock, MUTEX\_LOCK, MUTEX\_UNLOCK, mc\_rwlock\_s::num\_readers, mc\_rwlock\_s::write\_flag, and mc\_rwlock\_s::write\_request.

#### 8.51.1.4 int mc\_rwlock\_rdunlock (mc\_rwlock\_p *rwlock*)

Definition at line 89 of file mc\_rwlock.c.

References mc\_rwlock\_s::cond, COND\_SIGNAL, mc\_rwlock\_s::lock, MUTEX\_LOCK, MUTEX\_UNLOCK, and mc\_rwlock\_s::num\_readers.

**8.51.1.5 int mc\_rwlock\_wrlock (mc\_rwlock\_p *rwlock*)**

Definition at line 102 of file mc\_rwlock.c.

References mc\_rwlock\_s::cond, COND\_WAIT, mc\_rwlock\_s::lock, MUTEX\_LOCK, MUTEX\_UNLOCK, mc\_rwlock\_s::num\_readers, mc\_rwlock\_s::write\_flag, and mc\_rwlock\_s::write\_request.

**8.51.1.6 int mc\_rwlock\_wrunlock (mc\_rwlock\_p *rwlock*)**

Definition at line 121 of file mc\_rwlock.c.

References mc\_rwlock\_s::cond, COND\_SIGNAL, mc\_rwlock\_s::lock, MUTEX\_LOCK, MUTEX\_UNLOCK, and mc\_rwlock\_s::write\_flag.

## 8.52 /home/dko/projects/mobilec/trunk/src/mc\_sync/sync\_list.c File Reference

```
#include <pthread.h>
#include "sync_list.h"
#include "../mc_list/list.h"
#include "../include/mc_error.h"
```

### Functions

- [int syncListNodeInit](#) (struct [syncListNode\\_s](#) \**node*)
- [struct syncListNode\\_s \\* syncListNodeNew](#) (void)
- [int syncListNodeDestroy](#) (struct [syncListNode\\_s](#) \**node*)
- [syncListNode\\_t \\* syncListFind](#) (int *id*, struct [syncList\\_s](#) \**list*)
- [struct syncList\\_s \\* syncListInit](#) (void)
- [int syncListAddNode](#) (struct [syncListNode\\_s](#) \**node*, struct [syncList\\_s](#) \**list*)
- [int syncListNew](#) (int *id*, struct [syncList\\_s](#) \**list*)
- [int syncListDelete](#) (int *id*, struct [syncList\\_s](#) \**list*)
- [syncListNode\\_t \\* syncListRemove](#) (int *id*, struct [syncList\\_s](#) \**list*)

### 8.52.1 Function Documentation

#### 8.52.1.1 int syncListAddNode (struct syncListNode\_s \* *node*, struct syncList\_s \* *list*)

Definition at line 86 of file `sync_list.c`.

References `DATA`, `syncListNode_s::id`, `syncList_s::list`, `ListAdd()`, `list_s::listhead`, `syncList_s::lock`, `listNode_s::next`, `listNode_s::node_data`, `RWLOCK_WRLock`, and `RWLOCK_WRunlock`.

Referenced by `MC_SyncInit()`, and `syncListNew()`.

#### 8.52.1.2 int syncListDelete (int *id*, struct syncList\_s \* *list*)

Definition at line 115 of file `sync_list.c`.

References `syncListNode_s::id`, `syncList_s::list`, `ListDelete()`, `ListSearch()`, `syncList_s::lock`, `MC_ERR_NOT_FOUND`, `RWLOCK_WRLock`, `RWLOCK_WRunlock`, `list_s::size`, and `syncListNodeDestroy()`.

#### 8.52.1.3 syncListNode\_t\* syncListFind (int *id*, struct syncList\_s \* *list*)

Definition at line 56 of file `sync_list.c`.

References `syncList_s::list`, `list_s::listhead`, `syncList_s::lock`, `listNode_s::next`, `listNode_s::node_data`, `RWLOCK_RDlock`, and `RWLOCK_RDunlock`.

Referenced by `MC_CondBroadcast()`, `MC_CondReset()`, `MC_CondSignal()`, `MC_CondWait()`, `MC_MutexLock()`, `MC_MutexUnlock()`, `MC_SemaphorePost()`, `MC_SemaphoreWait()`, `MC_SyncDelete()`, and `MC_SyncInit()`.

**8.52.1.4 struct syncList\_s\* syncListInit (void) [read]**

Definition at line 72 of file sync\_list.c.

References ListInitialize(), syncList\_s::lock, MUTEX\_INIT, MUTEX\_T, RWLOCK\_INIT, and RWLOCK\_T.

Referenced by mc\_platform\_Initialize().

**8.52.1.5 int syncListNew (int id, struct syncList\_s \* list)**

Definition at line 105 of file sync\_list.c.

References node, syncListAddNode(), and syncListNodeInit().

**8.52.1.6 int syncListNodeDestroy (struct syncListNode\_s \* node)**

Definition at line 44 of file sync\_list.c.

References syncListNode\_s::cond, COND\_DESTROY, syncListNode\_s::lock, MUTEX\_DESTROY, syncListNode\_s::sem, and SEMAPHORE\_DESTROY.

Referenced by MC\_SyncDelete(), and syncListDelete().

**8.52.1.7 int syncListNodeInit (struct syncListNode\_s \* node)**

Definition at line 11 of file sync\_list.c.

References CHECK\_NULL, syncListNode\_s::cond, COND\_INIT, COND\_T, syncListNode\_s::lock, MUTEX\_INIT, MUTEX\_T, syncListNode\_s::sem, SEMAPHORE\_INIT, and SEMAPHORE\_T.

Referenced by syncListNew().

**8.52.1.8 struct syncListNode\_s\* syncListNodeNew (void) [read]**

Definition at line 26 of file sync\_list.c.

References CHECK\_NULL, syncListNode\_s::cond, COND\_INIT, COND\_T, syncListNode\_s::lock, MUTEX\_INIT, MUTEX\_T, syncListNode\_s::sem, SEMAPHORE\_INIT, SEMAPHORE\_T, and syncListNode\_s::signalled.

Referenced by MC\_SyncInit().

**8.52.1.9 syncListNode\_t\* syncListRemove (int id, struct syncList\_s \* list)**

Definition at line 132 of file sync\_list.c.

References syncListNode\_s::id, syncList\_s::list, ListDelete(), ListSearch(), syncList\_s::lock, RWLOCK\_WRLock, RWLOCK\_WRunLock, and list\_s::size.

Referenced by MC\_SyncDelete().

## 8.53 /home/dko/projects/mobilec/trunk/src/mc\_sync/sync\_list.h File Reference

```
#include "../include/macros.h"
#include "../mc_list/list.h"
#include "../include/mc_rwlock.h"
```

### Data Structures

- struct [syncListNode\\_s](#)
- struct [syncList\\_s](#)

### Typedefs

- typedef struct [syncListNode\\_s](#) [syncListNode\\_t](#)
- typedef [syncListNode\\_t](#) \* [syncListNode\\_p](#)
- typedef struct [syncList\\_s](#) [syncList\\_t](#)
- typedef [syncList\\_t](#) \* [syncList\\_p](#)

### Functions

- [int syncListNodeInit](#) (struct [syncListNode\\_s](#) \*[node](#))
- [int syncListNodeDestroy](#) (struct [syncListNode\\_s](#) \*[node](#))
- [syncListNode\\_t](#) \* [syncListFind](#) (int [id](#), struct [syncList\\_s](#) \*[list](#))
- struct [syncListNode\\_s](#) \* [syncListNodeNew](#) (void)
- [int syncListDelete](#) (int [id](#), struct [syncList\\_s](#) \*[list](#))
- [syncListNode\\_t](#) \* [syncListRemove](#) (int [id](#), struct [syncList\\_s](#) \*[list](#))
- struct [syncList\\_s](#) \* [syncListInit](#) (void)
- [int syncListAddNode](#) (struct [syncListNode\\_s](#) \*[node](#), struct [syncList\\_s](#) \*[list](#))
- [int syncListNew](#) (int [id](#), struct [syncList\\_s](#) \*[list](#))
- [syncListNode\\_t](#) \* [syncListGet](#) (int [id](#), struct [syncList\\_s](#) \*[list](#))

#### 8.53.1 Typedef Documentation

##### 8.53.1.1 typedef [syncList\\_t](#)\* [syncList\\_p](#)

Definition at line 33 of file [sync\\_list.h](#).

##### 8.53.1.2 typedef struct [syncList\\_s](#) [syncList\\_t](#)

##### 8.53.1.3 typedef [syncListNode\\_t](#)\* [syncListNode\\_p](#)

Definition at line 23 of file [sync\\_list.h](#).

#### 8.53.1.4 typedef struct syncListNode\_s syncListNode\_t

### 8.53.2 Function Documentation

#### 8.53.2.1 int syncListAddNode (struct syncListNode\_s \* *node*, struct syncList\_s \* *list*)

Definition at line 86 of file sync\_list.c.

References DATA, syncListNode\_s::id, syncList\_s::list, ListAdd(), list\_s::listhead, syncList\_s::lock, listNode\_s::next, listNode\_s::node\_data, RWLOCK\_WRLock, and RWLOCK\_WRunLock.

Referenced by MC\_SyncInit(), and syncListNew().

#### 8.53.2.2 int syncListDelete (int *id*, struct syncList\_s \* *list*)

Definition at line 115 of file sync\_list.c.

References syncListNode\_s::id, syncList\_s::list, ListDelete(), ListSearch(), syncList\_s::lock, MC\_ERR\_NOT\_FOUND, RWLOCK\_WRLock, RWLOCK\_WRunLock, list\_s::size, and syncListNodeDestroy().

#### 8.53.2.3 syncListNode\_t\* syncListFind (int *id*, struct syncList\_s \* *list*)

Definition at line 56 of file sync\_list.c.

References syncList\_s::list, list\_s::listhead, syncList\_s::lock, listNode\_s::next, listNode\_s::node\_data, RWLOCK\_RDLOCK, and RWLOCK\_RDUnLock.

Referenced by MC\_CondBroadcast(), MC\_CondReset(), MC\_CondSignal(), MC\_CondWait(), MC\_MutexLock(), MC\_MutexUnlock(), MC\_SemaphorePost(), MC\_SemaphoreWait(), MC\_SyncDelete(), and MC\_SyncInit().

#### 8.53.2.4 syncListNode\_t\* syncListGet (int *id*, struct syncList\_s \* *list*)

#### 8.53.2.5 struct syncList\_s\* syncListInit (void) [read]

Definition at line 72 of file sync\_list.c.

References ListInitialize(), syncList\_s::lock, MUTEX\_INIT, MUTEX\_T, RWLOCK\_INIT, and RWLOCK\_T.

Referenced by mc\_platform\_Initialize().

#### 8.53.2.6 int syncListNew (int *id*, struct syncList\_s \* *list*)

Definition at line 105 of file sync\_list.c.

References node, syncListAddNode(), and syncListNodeInit().

#### 8.53.2.7 int syncListNodeDestroy (struct syncListNode\_s \* *node*)

Definition at line 44 of file sync\_list.c.

References syncListNode\_s::cond, COND\_DESTROY, syncListNode\_s::lock, MUTEX\_DESTROY, syncListNode\_s::sem, and SEMAPHORE\_DESTROY.

Referenced by MC\_SyncDelete(), and syncListDelete().

#### **8.53.2.8 int syncListNodeInit (struct syncListNode\_s \* node)**

Definition at line 11 of file sync\_list.c.

References CHECK\_NULL, syncListNode\_s::cond, COND\_INIT, COND\_T, syncListNode\_s::lock, MUTEX\_INIT, MUTEX\_T, syncListNode\_s::sem, SEMAPHORE\_INIT, and SEMAPHORE\_T.

Referenced by syncListNew().

#### **8.53.2.9 struct syncListNode\_s\* syncListNodeNew (void) [read]**

Definition at line 26 of file sync\_list.c.

References CHECK\_NULL, syncListNode\_s::cond, COND\_INIT, COND\_T, syncListNode\_s::lock, MUTEX\_INIT, MUTEX\_T, syncListNode\_s::sem, SEMAPHORE\_INIT, SEMAPHORE\_T, and syncListNode\_s::signalled.

Referenced by MC\_SyncInit().

#### **8.53.2.10 syncListNode\_t\* syncListRemove (int id, struct syncList\_s \* list)**

Definition at line 132 of file sync\_list.c.

References syncListNode\_s::id, syncList\_s::list, ListDelete(), ListSearch(), syncList\_s::lock, RWLOCK\_WRLOCK, RWLOCK\_WRUNLOCK, and list\_s::size.

Referenced by MC\_SyncDelete().



## 8.54 /home/dko/projects/mobilec/trunk/src/message.c File Reference

```
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <netdb.h>
#include <mxml.h>
#include "include/libmc.h"
#include "include/agent.h"
#include "include/mc_platform.h"
#include "include/message.h"
#include "include/mtp_http.h"
#include "include/xml_compose.h"
#include "include/xml_helper.h"
#include "include/xml_parser.h"
#include "security/asm_node.h"
```

### Defines

- `#define` [SOCKET\\_INPUT\\_SIZE](#) 4096

### Functions

- [message\\_p](#) [message\\_New](#) (void)
- [message\\_p](#) [message\\_Copy](#) ([message\\_p](#) src)
- [int](#) [message\\_InitializeFromAgent](#) ([mc\\_platform\\_p](#) mc\_platform, [message\\_p](#) message, [agent\\_p](#) agent)
- [int](#) [message\\_InitializeFromConnection](#) ([mc\\_platform\\_p](#) mc\_platform, [message\\_p](#) message, [connection\\_p](#) connection)
- [int](#) [http\\_to\\_hostport](#) (const char \*http\_str, char \*\*host, [int](#) \*port, char \*\*target)
- [int](#) [message\\_InitializeFromString](#) ([mc\\_platform\\_p](#) mc\_platform, [message\\_p](#) message, const char \*string, const char \*destination\_host, [int](#) destination\_port, const char \*target)
- [int](#) [message\\_Destroy](#) ([message\\_p](#) message)
- [int](#) [message\\_Send](#) ([message\\_p](#) message)

## 8.54.1 Define Documentation

### 8.54.1.1 `#define SOCKET_INPUT_SIZE 4096`

Definition at line 58 of file message.c.

## 8.54.2 Function Documentation

### 8.54.2.1 `int http_to_hostport (const char * http_str, char ** host, int * port, char ** target)`

Definition at line 505 of file message.c.

References MC\_ERR\_PARSE.

Referenced by MC\_AclSend().

### 8.54.2.2 `message_p message_Copy (message_p src)`

Definition at line 310 of file message.c.

### 8.54.2.3 `int message_Destroy (message_p message)`

Definition at line 604 of file message.c.

References message\_s::addr, message\_s::agent\_xml\_flag, message\_s::from\_address, MC\_SUCCESS, message\_s::message\_body, mxmlDelete(), message\_s::target, message\_s::to\_address, message\_s::update\_name, and message\_s::xml\_root.

Referenced by acc\_MessageHandlerThread(), acc\_Thread(), ams\_ManageAgentList(), MC\_AclSend(), MC\_LoadAgentFromFile(), MC\_SendAgentMigrationMessage(), MC\_SendAgentMigrationMessageFile(), message\_InitializeFromConnection(), and message\_queue\_SendOutgoing().

### 8.54.2.4 `int message_InitializeFromAgent (mc_platform_p mc_platform, message_p message, agent_p agent)`

Definition at line 318 of file message.c.

References message\_s::addr, agent\_xml\_compose(), message\_s::agent\_xml\_flag, CHECK\_NULL, agent\_s::datastate, mc\_platform\_s::err, message\_s::from\_address, agent\_s::home, mc\_platform\_s::hostname, MC\_ERR\_MEMORY, MC\_SUCCESS, message\_s::message\_body, message\_s::message\_id, message\_s::message\_type, MOBILE\_AGENT, MXML\_NO\_CALLBACK, mxmlSaveAllocString(), agent\_datastate\_s::number\_of\_tasks, mc\_platform\_s::port, agent\_task\_s::server\_name, strtok\_r, message\_s::target, agent\_datastate\_s::task\_progress, agent\_datastate\_s::tasks, message\_s::to\_address, message\_s::update\_name, WARN, and message\_s::xml\_root.

### 8.54.2.5 `int message_InitializeFromConnection (mc_platform_p mc_platform, message_p message, connection_p connection)`

Definition at line 417 of file message.c.

References connection\_s::addr, message\_s::addr, CHECK\_NULL, connection\_s::clientfd, connection\_s::connect\_id, message\_s::connect\_id, message\_s::from\_address, MC\_ERR\_CONNECT, MC\_ERR\_

PARSE, MC\_SUCCESS, message\_s::message\_body, message\_Destroy(), message\_s::message\_id, message\_xml\_parse(), MXML\_NO\_CALLBACK, mxmlLoadString(), SOCKET\_INPUT\_SIZE, message\_s::target, message\_s::to\_address, and message\_s::xml\_root.

#### 8.54.2.6 int message\_InitializeFromString (mc\_platform\_p mc\_platform, message\_p message, const char \* string, const char \* destination\_host, int destination\_port, const char \* target)

Definition at line 540 of file message.c.

References message\_s::addr, CHECK\_NULL, message\_s::connect\_id, mc\_platform\_s::err, message\_s::from\_address, mc\_platform\_s::hostname, MC\_ERR\_MEMORY, MC\_SUCCESS, message\_s::message\_body, message\_s::message\_id, message\_s::message\_type, MOBILE\_AGENT, mc\_platform\_s::port, message\_s::target, message\_s::to\_address, message\_s::update\_name, and message\_s::xml\_root.

#### 8.54.2.7 message\_p message\_New (void)

Definition at line 286 of file message.c.

References message\_s::addr, message\_s::agent\_xml\_flag, CHECK\_NULL, message\_s::connect\_id, message\_s::from\_address, message\_s::http\_type, message\_s::isHTTP, message\_s::message\_body, message\_s::message\_id, message\_s::message\_type, message\_s::target, message\_s::to\_address, message\_s::update\_name, message\_s::update\_num, message\_s::xml\_payload, and message\_s::xml\_root.

Referenced by acc\_Thread(), ams\_ManageAgentList(), MC\_LoadAgentFromFile(), MC\_SendAgentMigrationMessage(), MC\_SendAgentMigrationMessageFile(), and mtp\_http\_CreateMessage().

#### 8.54.2.8 int message\_Send (message\_p message)

Definition at line 642 of file message.c.

References CHECK\_NULL, MC\_ERR, MC\_ERR\_CONNECT, MC\_ERR\_SEND, message\_s::message\_body, mtp\_http\_ComposeMessage(), mtp\_http\_New(), mtp\_http\_Parse(), port, send, SOCKET\_INPUT\_SIZE, strtok\_r, and message\_s::to\_address.

Referenced by acc\_MessageHandlerThread(), MC\_AclSend(), and message\_queue\_SendOutgoing().

## 8.55 /home/dko/projects/mobilec/trunk/src/message\_queue.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include "include/data_structures.h"
#include "include/mc_platform.h"
```

### Functions

- void [message\\_queue\\_SendOutgoing](#) (struct [mc\\_platform\\_s](#) \*[mc\\_platform](#), [message\\_queue\\_p](#) [mqueue](#))

#### 8.55.1 Function Documentation

##### 8.55.1.1 void [message\\_queue\\_SendOutgoing](#) (struct [mc\\_platform\\_s](#) \* [mc\\_platform](#), [message\\_queue\\_p](#) [mqueue](#))

Definition at line 52 of file [message\\_queue.c](#).

References [CHECK\\_NULL](#), [mc\\_platform\\_s::hostname](#), [ListSearch\(\)](#), [message\\_Destroy\(\)](#), [message\\_Send\(\)](#), [MUTEX\\_LOCK](#), [MUTEX\\_UNLOCK](#), [mc\\_platform\\_s::port](#), and [message\\_s::to\\_address](#).

## 8.56 /home/dko/projects/mobilec/trunk/src/mtp\_http.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <unistd.h>
#include "config.h"
#include "include/connection.h"
#include "include/mtp_http.h"
#include "include/macros.h"
#include "include/mc_error.h"
#include "include/message.h"
#include "include/dynstring.h"
```

### Defines

- #define [SAFE\\_FREE](#)(elem)
- #define [HTTP\\_PARSE\\_EXPR](#)(parse\_name, struct\_name)
- #define [SAFE\\_FREE](#)(object)

### Functions

- [int mtp\\_http\\_Destroy](#) ([mtp\\_http\\_p](#) http)
- [mtp\\_http\\_p mtp\\_http\\_New](#) (void)
- [int mtp\\_http\\_InitializeFromConnection](#) ([mtp\\_http\\_p](#) http, [connection\\_p](#) connection)
- const char \* [http\\_GetExpression](#) (const char \*string, char \*\*expr)  
*Parse an html expression.*
- [int http\\_ParseExpression](#) (const char \*expression\_string, char \*\*name, char \*\*value)  
*Parse an expression into its name and value.*
- [int mtp\\_http\\_Parse](#) (struct [mtp\\_http\\_s](#) \*http, const char \*string)
- const char \* [http\\_ParseHeader](#) ([mtp\\_http\\_p](#) http, const char \*string)
- const char \* [http\\_GetToken](#) (const char \*string, char \*\*token)
- [int mtp\\_http\\_ParseResponse](#) (struct [mtp\\_http\\_s](#) \*http, const char \*string)
- [int mtp\\_http\\_ComposeMessage](#) ([message\\_p](#) message)
- struct [message\\_s](#) \* [mtp\\_http\\_CreateMessage](#) ([mtp\\_http\\_t](#) \*mtp\_http, char \*hostname, [int](#) port)

### 8.56.1 Define Documentation

#### 8.56.1.1 #define [HTTP\\_PARSE\\_EXPR](#)(parse\_name, struct\_name)

**Value:**

```
if ( !strcmp(name, parse_name) ) { \
    http->struct_name = (char*)malloc \
    ( \
        sizeof(char) * \
        (strlen(value)+1) \
    ); \
    strcpy(http->struct_name, value); \
} else
```

Referenced by mtp\_http\_Parse().

#### 8.56.1.2 #define SAFE\_FREE(object)

**Value:**

```
if(object) free(object); \
    object = NULL
```

#### 8.56.1.3 #define SAFE\_FREE(elem)

**Value:**

```
if(elem) \
    free(elem)
```

Referenced by mtp\_http\_Destroy(), and mtp\_http\_Parse().

### 8.56.2 Function Documentation

#### 8.56.2.1 const char\* http\_GetExpression (const char \* *string*, char \*\* *expr*)

Parse an html expression.

**Parameters:**

*string* (input) The html block of text: Will parse the first expression pointed to by 'string'.

*expr* (output) The allocated expression

**Returns:**

A pointer to the next expression segment of the string block, or NULL.

Definition at line 210 of file mtp\_http.c.

Referenced by mtp\_http\_Parse().

#### 8.56.2.2 const char\* http\_GetToken (const char \* *string*, char \*\* *token*)

Definition at line 633 of file mtp\_http.c.

Referenced by http\_ParseHeader().

**8.56.2.3 int http\_ParseExpression (const char \* *expression\_string*, char \*\* *name*, char \*\* *value*)**

Parse an expression into its name and value.

**Parameters:**

*expression\_string* (input) The expression  
*name* (output) An allocated name string or NULL  
*value* (output) An allocated value string or NULL

**Returns:**

error\_code\_t type

**Note:**

an http expression is something like 'Date: Mon, 23 May 2005 22:38:34 GMT'  
 ' where 'Date' is the name and the remainder of the string is the value

Definition at line 269 of file mtp\_http.c.

References CHECK\_NULL, MC\_ERR\_PARSE, and MC\_SUCCESS.

Referenced by mtp\_http\_Parse().

**8.56.2.4 const char\* http\_ParseHeader (mtp\_http\_p *http*, const char \* *string*)**

Definition at line 544 of file mtp\_http.c.

References HTTP\_CONNECT, HTTP\_DELETE, HTTP\_GET, http\_GetToken(), HTTP\_HEAD, HTTP\_OPTIONS, mtp\_http\_s::http\_performative, HTTP\_PERFORMATIVE\_UNDEF, HTTP\_POST, HTTP\_PUT, HTTP\_RESPONSE, HTTP\_TRACE, mtp\_http\_s::response\_code, mtp\_http\_s::response\_string, and mtp\_http\_s::target.

Referenced by mtp\_http\_Parse().

**8.56.2.5 int mtp\_http\_ComposeMessage (message\_p *message*)**

Definition at line 672 of file mtp\_http.c.

References message\_s::isHTTP, MC\_SUCCESS, message\_s::message\_body, PACKAGE\_VERSION, message\_s::target, and message\_s::to\_address.

**8.56.2.6 struct message\_s\* mtp\_http\_CreateMessage (mtp\_http\_t \* *mtp\_http*, char \* *hostname*, int *port*) [read]**

Definition at line 718 of file mtp\_http.c.

References mtp\_http\_s::content, mtp\_http\_content\_s::content\_type, mtp\_http\_content\_s::data, dynstring\_Append(), dynstring\_Destroy(), dynstring\_New(), mtp\_http\_s::host, message\_s::isHTTP, dynstring\_s::len, dynstring\_s::message, message\_s::message\_body, message\_New(), mtp\_http\_s::message\_parts, PACKAGE\_VERSION, mtp\_http\_s::target, and message\_s::to\_address.

Referenced by MC\_AclSend().

### 8.56.2.7 `int mtp_http_Destroy (mtp_http_p http)`

Definition at line 53 of file `mtp_http.c`.

References `mtp_http_s::accept_ranges`, `mtp_http_s::boundary`, `mtp_http_s::connection`, `mtp_http_s::content`, `mtp_http_s::content_length`, `mtp_http_content_s::content_type`, `mtp_http_s::content_type`, `mtp_http_content_s::data`, `mtp_http_s::date`, `mtp_http_s::host`, `mtp_http_s::http_version`, `mtp_http_s::message_parts`, `mtp_http_s::return_code`, `SAFE_FREE`, `mtp_http_s::server`, `mtp_http_s::target`, and `mtp_http_s::user_agent`.

Referenced by `acc_Thread()`, and `MC_AclSend()`.

### 8.56.2.8 `int mtp_http_InitializeFromConnection (mtp_http_p http, connection_p connection)`

Definition at line 97 of file `mtp_http.c`.

References `CHECK_NULL`, `connection_s::clientfd`, `ERR`, `MC_ERR_CONNECT`, `mtp_http_Parse()`, `send`, and `SOCKET_INPUT_SIZE`.

### 8.56.2.9 `mtp_http_p mtp_http_New (void)`

Definition at line 85 of file `mtp_http.c`.

References `CHECK_NULL`, and `mtp_http_s::content`.

Referenced by `acc_Thread()`, `MC_AclSend()`, and `message_Send()`.

### 8.56.2.10 `int mtp_http_Parse (struct mtp_http_s * http, const char * string)`

Definition at line 329 of file `mtp_http.c`.

References `mtp_http_s::boundary`, `mtp_http_s::content`, `mtp_http_content_s::content_type`, `mtp_http_s::content_type`, `mtp_http_content_s::data`, `http_GetExpression()`, `HTTP_PARSE_EXPR`, `http_ParseExpression()`, `http_ParseHeader()`, `mtp_http_s::http_performative`, `HTTP_POST`, `HTTP_PUT`, `HTTP_RESPONSE`, `MC_SUCCESS`, `mtp_http_s::message_parts`, and `SAFE_FREE`.

Referenced by `message_Send()`, and `mtp_http_InitializeFromConnection()`.

### 8.56.2.11 `int mtp_http_ParseResponse (struct mtp_http_s * http, const char * string)`

Definition at line 666 of file `mtp_http.c`.



## 8.57 /home/dko/projects/mobilec/trunk/src/mxml-2.2.2/config.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdarg.h>
#include <ctype.h>
```

### Defines

- `#define` [MXML\\_VERSION](#) "Mini-XML v2.2.2"
- `#define` [HAVE\\_VSNPRINTF](#) 1
- `#define` [HAVE\\_STRDUP](#) 1

### Functions

- `char *` [mxml\\_strdup](#) (`const char *`, `va_list`)

#### 8.57.1 Define Documentation

##### 8.57.1.1 `#define HAVE_STRDUP 1`

Definition at line 49 of file config.h.

##### 8.57.1.2 `#define HAVE_VSNPRINTF 1`

Definition at line 42 of file config.h.

##### 8.57.1.3 `#define MXML_VERSION "Mini-XML v2.2.2"`

Definition at line 35 of file config.h.

Referenced by `write_documentation()`.

#### 8.57.2 Function Documentation

##### 8.57.2.1 `char* mxml_strdup (const char *, va_list)`

Definition at line 62 of file mxml-string.c.

Referenced by `mxml_error()`, `mxmlNewTextf()`, and `mxmlSetTextf()`.

## 8.58 /home/dko/projects/mobilec/trunk/src/mxml-2.2.2/vcnet/config.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdarg.h>
#include <ctype.h>
```

### Defines

- #define [MXML\\_VERSION](#) "Mini-XML v2.0"
- #define [HAVE\\_STRDUP](#) 1
- #define [vsnprintf](#) mxml\_vsnprintf

### Functions

- char \* [mxml\\_strdup](#) (const char \*, va\_list)
- int [mxml\\_vsnprintf](#) (char \*, size\_t, const char \*, va\_list)

#### 8.58.1 Define Documentation

##### 8.58.1.1 #define HAVE\_STRDUP 1

Definition at line 51 of file config.h.

##### 8.58.1.2 #define MXML\_VERSION "Mini-XML v2.0"

Definition at line 37 of file config.h.

##### 8.58.1.3 #define vsnprintf mxml\_vsnprintf

Definition at line 67 of file config.h.

#### 8.58.2 Function Documentation

##### 8.58.2.1 char\* mxml\_strdup (const char \*, va\_list)

Definition at line 62 of file mxml-string.c.

##### 8.58.2.2 int mxml\_vsnprintf (char \*, size\_t, const char \*, va\_list)

Definition at line 107 of file mxml-string.c.

## 8.59 /home/dko/projects/mobilec/trunk/src/mxml-2.2.2/vcnet2005/config.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdarg.h>
#include <ctype.h>
```

### Defines

- `#define MXML_VERSION` "Mini-XML v2.0"
- `#define HAVE_STRDUP` 1
- `#define vsnprintf` mxml\_vsnprintf

### Functions

- `char * mxml_strdup` (const char \*, va\_list)
- `int mxml_vsnprintf` (char \*, size\_t, const char \*, va\_list)

#### 8.59.1 Define Documentation

##### 8.59.1.1 `#define HAVE_STRDUP` 1

Definition at line 51 of file config.h.

##### 8.59.1.2 `#define MXML_VERSION` "Mini-XML v2.0"

Definition at line 37 of file config.h.

##### 8.59.1.3 `#define vsnprintf` mxml\_vsnprintf

Definition at line 67 of file config.h.

#### 8.59.2 Function Documentation

##### 8.59.2.1 `char* mxml_strdup` (const char \*, va\_list)

Definition at line 62 of file mxml-string.c.

References `mxml_vsnprintf()`.

##### 8.59.2.2 `int mxml_vsnprintf` (char \*, size\_t, const char \*, va\_list)

Definition at line 107 of file mxml-string.c.

References size.

Referenced by `mxml_strdup()`.

## 8.60 /home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-attr.c File Reference

```
#include "config.h"
#include "mxml.h"
```

### Functions

- const char \* [mxmlElementGetAttr](#) (mxml\_node\_t \*node, const char \*name)
- void [mxmlElementSetAttr](#) (mxml\_node\_t \*node, const char \*name, const char \*value)

### 8.60.1 Function Documentation

#### 8.60.1.1 const char\* mxmlElementGetAttr (mxml\_node\_t \* node, const char \* name)

Definition at line 43 of file mxml-attr.c.

References [mxml\\_value\\_s::attrs](#), [mxml\\_value\\_u::element](#), [MXML\\_ELEMENT](#), [mxml\\_attr\\_s::name](#), [mxml\\_value\\_s::num\\_attrs](#), [mxml\\_node\\_s::type](#), [mxml\\_attr\\_s::value](#), and [mxml\\_node\\_s::value](#).

Referenced by [agent\\_xml\\_parse\\_\\_agent\\_code\(\)](#), [agent\\_xml\\_parse\\_\\_data\(\)](#), [agent\\_xml\\_parse\\_\\_task\(\)](#), [agent\\_xml\\_parse\\_\\_tasks\(\)](#), [fipa\\_envelope\\_HandleReceived\(\)](#), [index\\_compare\(\)](#), [index\\_find\(\)](#), [mxml\\_parse\\_element\(\)](#), [mxmlFindElement\(\)](#), [mxmlIndexNew\(\)](#), [scan\\_file\(\)](#), [sort\\_node\(\)](#), [type\\_cb\(\)](#), and [write\\_documentation\(\)](#).

#### 8.60.1.2 void mxmlElementSetAttr (mxml\_node\_t \* node, const char \* name, const char \* value)

Definition at line 90 of file mxml-attr.c.

References [mxml\\_value\\_s::attrs](#), [mxml\\_value\\_u::element](#), [MXML\\_ELEMENT](#), [mxml\\_error\(\)](#), [mxml\\_value\\_s::name](#), [mxml\\_attr\\_s::name](#), [mxml\\_value\\_s::num\\_attrs](#), [mxml\\_node\\_s::type](#), [mxml\\_attr\\_s::value](#), and [mxml\\_node\\_s::value](#).

Referenced by [add\\_variable\(\)](#), [agent\\_xml\\_compose\\_\\_agent\\_code\(\)](#), [agent\\_xml\\_compose\\_\\_create\\_row\\_nodes\(\)](#), [agent\\_xml\\_compose\\_\\_data\(\)](#), [agent\\_xml\\_compose\\_\\_message\(\)](#), [agent\\_xml\\_compose\\_\\_task\(\)](#), [agent\\_xml\\_compose\\_\\_tasks\(\)](#), [fipa\\_envelope\\_Compose\\_\\_params\(\)](#), [main\(\)](#), [mxml\\_parse\\_element\(\)](#), [scan\\_file\(\)](#), [sort\\_node\(\)](#), and [update\\_comment\(\)](#).

## 8.61 /home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-entity.c File Reference

```
#include "config.h"
#include "mxml.h"
```

### Functions

- static [int default\\_callback](#) (const char \*name)
- [int mxmlEntityAddCallback](#) (int(\*cb)(const char \*name))
- const char \* [mxmlEntityGetName](#) (int val)
- [int mxmlEntityGetValue](#) (const char \*name)
- void [mxmlEntityRemoveCallback](#) (int(\*cb)(const char \*name))

### Variables

- static [int num\\_callbacks](#) = 1
- static [int\(\\* callbacks](#) [100])(const char \*name)

#### 8.61.1 Function Documentation

##### 8.61.1.1 static int default\_callback (const char \* *name*) [static]

Definition at line 169 of file mxml-entity.c.

References [int](#).

##### 8.61.1.2 int mxmlEntityAddCallback (int(\*) (const char \*name) *cb*)

Definition at line 65 of file mxml-entity.c.

References [callbacks](#), [mxml\\_error\(\)](#), and [num\\_callbacks](#).

##### 8.61.1.3 const char\* mxmlEntityGetName (int *val*)

Definition at line 91 of file mxml-entity.c.

Referenced by [mxml\\_write\\_name\(\)](#), and [mxml\\_write\\_string\(\)](#).

##### 8.61.1.4 int mxmlEntityGetValue (const char \* *name*)

Definition at line 121 of file mxml-entity.c.

References [callbacks](#), and [num\\_callbacks](#).

Referenced by [mxml\\_get\\_entity\(\)](#).

#### 8.61.1.5 void mxmlEntityRemoveCallback (int(\*) (const char \*name) cb)

Definition at line 140 of file mxml-entity.c.

References callbacks, and num\_callbacks.

### 8.61.2 Variable Documentation

#### 8.61.2.1 int(\* callbacks[100])(const char \*name) [static]

**Initial value:**

```
{
    default_callback
}
```

Referenced by mxmlEntityAddCallback(), mxmlEntityGetValue(), and mxmlEntityRemoveCallback().

#### 8.61.2.2 int num\_callbacks = 1 [static]

Definition at line 53 of file mxml-entity.c.

Referenced by mxmlEntityAddCallback(), mxmlEntityGetValue(), and mxmlEntityRemoveCallback().

## 8.62 /home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-file.c File Reference

```
#include "config.h"
#include "mxml.h"
#include <unistd.h>
```

### Data Structures

- struct [mxml\\_fdbuf\\_s](#)

### Defines

- #define [ENCODE\\_UTF8](#) 0
- #define [ENCODE\\_UTF16BE](#) 1
- #define [ENCODE\\_UTF16LE](#) 2
- #define [mxml\\_bad\\_char](#)(ch) ((ch) < ' ' && (ch) != '\n' && (ch) != '\r' && (ch) != '\t')

### Typedefs

- typedef struct [mxml\\_fdbuf\\_s](#) [mxml\\_fdbuf\\_t](#)

### Functions

- static [int](#) [mxml\\_add\\_char](#) ([int](#) ch, [char](#) \*\*ptr, [char](#) \*\*buffer, [int](#) \*bufsize)
- static [int](#) [mxml\\_fd\\_getc](#) ([void](#) \*p, [int](#) \*encoding)
- static [int](#) [mxml\\_fd\\_putc](#) ([int](#) ch, [void](#) \*p)
- static [int](#) [mxml\\_fd\\_read](#) ([mxml\\_fdbuf\\_t](#) \*buf)
- static [int](#) [mxml\\_fd\\_write](#) ([mxml\\_fdbuf\\_t](#) \*buf)
- static [int](#) [mxml\\_file\\_getc](#) ([void](#) \*p, [int](#) \*encoding)
- static [int](#) [mxml\\_file\\_putc](#) ([int](#) ch, [void](#) \*p)
- static [int](#) [mxml\\_get\\_entity](#) ([mxml\\_node\\_t](#) \*parent, [void](#) \*p, [int](#) \*encoding, [int](#)(\*getc\_cb)([void](#) \*, [int](#) \*))
- static [mxml\\_node\\_t](#) \* [mxml\\_load\\_data](#) ([mxml\\_node\\_t](#) \*top, [void](#) \*p, [mxml\\_type\\_t](#)(\*cb)([mxml\\_node\\_t](#) \*), [int](#)(\*getc\_cb)([void](#) \*, [int](#) \*))
- static [int](#) [mxml\\_parse\\_element](#) ([mxml\\_node\\_t](#) \*node, [void](#) \*p, [int](#) \*encoding, [int](#)(\*getc\_cb)([void](#) \*, [int](#) \*))
- static [int](#) [mxml\\_string\\_getc](#) ([void](#) \*p, [int](#) \*encoding)
- static [int](#) [mxml\\_string\\_putc](#) ([int](#) ch, [void](#) \*p)
- static [int](#) [mxml\\_write\\_name](#) ([const](#) [char](#) \*s, [void](#) \*p, [int](#)(\*putc\_cb)([int](#), [void](#) \*))
- static [int](#) [mxml\\_write\\_node](#) ([mxml\\_node\\_t](#) \*node, [void](#) \*p, [const](#) [char](#) \*(\*cb)([mxml\\_node\\_t](#) \*, [int](#)), [int](#) col, [int](#)(\*putc\_cb)([int](#), [void](#) \*))
- static [int](#) [mxml\\_write\\_string](#) ([const](#) [char](#) \*s, [void](#) \*p, [int](#)(\*putc\_cb)([int](#), [void](#) \*))
- static [int](#) [mxml\\_write\\_ws](#) ([mxml\\_node\\_t](#) \*node, [void](#) \*p, [const](#) [char](#) \*(\*cb)([mxml\\_node\\_t](#) \*, [int](#)), [int](#) ws, [int](#) col, [int](#)(\*putc\_cb)([int](#), [void](#) \*))
- [mxml\\_node\\_t](#) \* [mxmlLoadFd](#) ([mxml\\_node\\_t](#) \*top, [int](#) fd, [mxml\\_type\\_t](#)(\*cb)([mxml\\_node\\_t](#) \*node))
- [mxml\\_node\\_t](#) \* [mxmlLoadFile](#) ([mxml\\_node\\_t](#) \*top, [FILE](#) \*fp, [mxml\\_type\\_t](#)(\*cb)([mxml\\_node\\_t](#) \*node))



- `mxml_node_t * mxmlLoadString (mxml_node_t *top, const char *s, mxml_type_t(*cb)(mxml_node_t *node))`
- `char * mxmlSaveAllocString (mxml_node_t *node, const char *(*cb)(mxml_node_t *node, int ws))`
- `int mxmlSaveFd (mxml_node_t *node, int fd, const char *(*cb)(mxml_node_t *node, int ws))`
- `int mxmlSaveFile (mxml_node_t *node, FILE *fp, const char *(*cb)(mxml_node_t *node, int ws))`
- `int mxmlSaveString (mxml_node_t *node, char *buffer, int bufsize, const char *(*cb)(mxml_node_t *node, int ws))`
- `void mxmlSetCustomHandlers (mxml_custom_load_cb_t load, mxml_custom_save_cb_t save)`
- `void mxmlSetErrorCallback (void(*cb)(const char *))`

## Variables

- `void(* mxml_error_cb )(const char *)`
- `static mxml_custom_load_cb_t mxml_custom_load_cb = NULL`
- `static mxml_custom_save_cb_t mxml_custom_save_cb = NULL`

### 8.62.1 Define Documentation

#### 8.62.1.1 `#define ENCODE_UTF16BE 1`

Definition at line 68 of file mxml-file.c.

Referenced by `mxml_fd_getc()`, `mxml_file_getc()`, and `mxml_string_getc()`.

#### 8.62.1.2 `#define ENCODE_UTF16LE 2`

Definition at line 69 of file mxml-file.c.

Referenced by `mxml_fd_getc()`, `mxml_file_getc()`, and `mxml_string_getc()`.

#### 8.62.1.3 `#define ENCODE_UTF8 0`

Definition at line 67 of file mxml-file.c.

Referenced by `mxml_fd_getc()`, `mxml_file_getc()`, `mxml_load_data()`, and `mxml_string_getc()`.

#### 8.62.1.4 `#define mxml_bad_char(ch) ((ch) < ' ' && (ch) != '\n' && (ch) != '\r' && (ch) != '\t')`

Definition at line 76 of file mxml-file.c.

Referenced by `mxml_fd_getc()`, `mxml_file_getc()`, `mxml_get_entity()`, and `mxml_string_getc()`.

### 8.62.2 Typedef Documentation

#### 8.62.2.1 `typedef struct mxml_fdbuf_s mxml_fdbuf_t`

### 8.62.3 Function Documentation

#### 8.62.3.1 `static int mxml_add_char (int ch, char **ptr, char **buffer, int *bufsize) [static]`

Definition at line 483 of file mxml-file.c.

References `mxml_error()`.

Referenced by `mxml_load_data()`, and `mxml_parse_element()`.

#### **8.62.3.2** `static int mxml_fd_getc (void *p, int *encoding)` `[static]`

Definition at line 563 of file `mxml-file.c`.

References `mxml_fdbuf_s::current`, `ENCODE_UTF16BE`, `ENCODE_UTF16LE`, `ENCODE_UTF8`, `mxml_fdbuf_s::end`, `mxml_bad_char`, `mxml_error()`, and `mxml_fd_read()`.

Referenced by `mxmlLoadFd()`.

#### **8.62.3.3** `static int mxml_fd_putc (int ch, void *p)` `[static]`

Definition at line 850 of file `mxml-file.c`.

References `mxml_fdbuf_s::current`, `mxml_fdbuf_s::end`, and `mxml_fd_write()`.

Referenced by `mxmlSaveFd()`.

#### **8.62.3.4** `static int mxml_fd_read (mxml_fdbuf_t *buf)` `[static]`

Definition at line 919 of file `mxml-file.c`.

References `mxml_fdbuf_s::buffer`, `mxml_fdbuf_s::current`, `mxml_fdbuf_s::end`, and `mxml_fdbuf_s::fd`.

Referenced by `mxml_fd_getc()`.

#### **8.62.3.5** `static int mxml_fd_write (mxml_fdbuf_t *buf)` `[static]`

Definition at line 958 of file `mxml-file.c`.

References `mxml_fdbuf_s::buffer`, `mxml_fdbuf_s::current`, and `mxml_fdbuf_s::fd`.

Referenced by `mxml_fd_putc()`, and `mxmlSaveFd()`.

#### **8.62.3.6** `static int mxml_file_getc (void *p, int *encoding)` `[static]`

Definition at line 1001 of file `mxml-file.c`.

References `ENCODE_UTF16BE`, `ENCODE_UTF16LE`, `ENCODE_UTF8`, `mxml_bad_char`, and `mxml_error()`.

Referenced by `mxmlLoadFile()`.

#### **8.62.3.7** `static int mxml_file_putc (int ch, void *p)` `[static]`

Definition at line 1200 of file `mxml-file.c`.

Referenced by `mxmlSaveFile()`.

**8.62.3.8** `static int mxml_get_entity (mxml_node_t * parent, void * p, int * encoding, int(*) (void *, int *) getc_cb)` [static]

Definition at line 1255 of file mxml-file.c.

References mxml\_value\_u::element, mxml\_bad\_char, mxml\_error(), mxmlEntityGetValue(), mxml\_value\_s::name, and mxml\_node\_s::value.

Referenced by mxml\_load\_data(), and mxml\_parse\_element().

**8.62.3.9** `static mxml_node_t * mxml_load_data (mxml_node_t * top, void * p, mxml_type_t(*) (mxml_node_t *) cb, int(*) (void *, int *) getc_cb)` [static]

Definition at line 1316 of file mxml-file.c.

References mxml\_value\_u::element, ENCODE\_UTF8, mxml\_add\_char(), MXML\_CUSTOM, mxml\_custom\_load\_cb, mxml\_error(), mxml\_get\_entity(), MXML\_INTEGER, MXML\_OPAQUE, mxml\_parse\_element(), MXML\_REAL, MXML\_TEXT, mxmlDelete(), mxmlNewCustom(), mxmlNewElement(), mxmlNewInteger(), mxmlNewOpaque(), mxmlNewReal(), mxmlNewText(), mxml\_value\_s::name, mxml\_node\_s::parent, and mxml\_node\_s::value.

Referenced by mxmlLoadFd(), mxmlLoadFile(), and mxmlLoadString().

**8.62.3.10** `static int mxml_parse_element (mxml_node_t * node, void * p, int * encoding, int(*) (void *, int *) getc_cb)` [static]

Definition at line 1840 of file mxml-file.c.

References mxml\_value\_u::element, mxml\_add\_char(), mxml\_error(), mxml\_get\_entity(), mxmlElementGetAttr(), mxmlElementSetAttr(), mxml\_value\_s::name, and mxml\_node\_s::value.

Referenced by mxml\_load\_data().

**8.62.3.11** `static int mxml_string_getc (void * p, int * encoding)` [static]

Definition at line 2102 of file mxml-file.c.

References ENCODE\_UTF16BE, ENCODE\_UTF16LE, ENCODE\_UTF8, mxml\_bad\_char, and mxml\_error().

Referenced by mxmlLoadString().

**8.62.3.12** `static int mxml_string_putc (int ch, void * p)` [static]

Definition at line 2338 of file mxml-file.c.

Referenced by mxmlSaveString().

**8.62.3.13** `static int mxml_write_name (const char * s, void * p, int(*) (int, void *) putc_cb)` [static]

Definition at line 2412 of file mxml-file.c.

References mxmlEntityGetName().

Referenced by mxml\_write\_node().

**8.62.3.14** `static int mxml_write_node (mxml_node_t * node, void * p, const char *(*)(mxml_node_t *, int) cb, int col, int(*)(int, void *) putc_cb) [static]`

Definition at line 2487 of file mxml-file.c.

References mxml\_value\_s::attrs, mxml\_node\_s::child, mxml\_value\_u::element, mxml\_value\_u::integer, MXML\_CUSTOM, mxml\_custom\_save\_cb, MXML\_ELEMENT, MXML\_INTEGER, MXML\_OPAQUE, MXML\_REAL, MXML\_TEXT, MXML\_WRAP, mxml\_write\_name(), mxml\_write\_string(), mxml\_write\_ws(), MXML\_WS\_AFTER\_CLOSE, MXML\_WS\_AFTER\_OPEN, MXML\_WS\_BEFORE\_CLOSE, MXML\_WS\_BEFORE\_OPEN, mxml\_attr\_s::name, mxml\_value\_s::name, mxml\_node\_s::next, mxml\_value\_s::num\_attrs, mxml\_value\_u::opaque, mxml\_node\_s::prev, mxml\_value\_u::real, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, mxml\_attr\_s::value, mxml\_node\_s::value, and mxml\_text\_s::whitespace.

Referenced by mxmlSaveFd(), mxmlSaveFile(), and mxmlSaveString().

**8.62.3.15** `static int mxml_write_string (const char * s, void * p, int(*)(int, void *) putc_cb) [static]`

Definition at line 2768 of file mxml-file.c.

References mxmlEntityGetName().

Referenced by mxml\_write\_node().

**8.62.3.16** `static int mxml_write_ws (mxml_node_t * node, void * p, const char *(*)(mxml_node_t *, int) cb, int ws, int col, int(*)(int, void *) putc_cb) [static]`

Definition at line 2808 of file mxml-file.c.

References MXML\_TAB.

Referenced by mxml\_write\_node().

**8.62.3.17** `mxml_node_t* mxmlLoadFd (mxml_node_t * top, int fd, mxml_type_t(*)(mxml_node_t * node) cb)`

Definition at line 159 of file mxml-file.c.

References mxml\_fdbuf\_s::buffer, mxml\_fdbuf\_s::current, mxml\_fdbuf\_s::end, mxml\_fdbuf\_s::fd, mxml\_fd\_getc(), and mxml\_load\_data().

Referenced by main().

**8.62.3.18** `mxml_node_t* mxmlLoadFile (mxml_node_t * top, FILE * fp, mxml_type_t(*)(mxml_node_t * node) cb)`

Definition at line 199 of file mxml-file.c.

References mxml\_file\_getc(), and mxml\_load\_data().

Referenced by main().

**8.62.3.19 mxml\_node\_t\* mxmlLoadString (mxml\_node\_t \* *top*, const char \* *s*, mxml\_type\_t(\*) (mxml\_node\_t \**node*) *cb*)**

Definition at line 228 of file mxml-file.c.

References mxml\_load\_data(), and mxml\_string\_getc().

Referenced by acc\_Thread(), agent\_xml\_compose(), fipa\_envelope\_Compose(), fipa\_envelope\_Parse(), main(), MC\_LoadAgentFromFile(), and message\_InitializeFromConnection().

**8.62.3.20 char\* mxmlSaveAllocString (mxml\_node\_t \* *node*, const char \*(\*) (mxml\_node\_t \**node*, int *ws*) *cb*)**

Definition at line 258 of file mxml-file.c.

References mxmlSaveString().

Referenced by fipa\_envelope\_Compose(), MC\_GetAgentXMLString(), and message\_InitializeFromAgent().

**8.62.3.21 int mxmlSaveFd (mxml\_node\_t \* *node*, int *fd*, const char \*(\*) (mxml\_node\_t \**node*, int *ws*) *cb*)**

Definition at line 315 of file mxml-file.c.

References mxml\_fdbuf\_s::buffer, mxml\_fdbuf\_s::current, mxml\_fdbuf\_s::end, mxml\_fdbuf\_s::fd, mxml\_fd\_putc(), mxml\_fd\_write(), and mxml\_write\_node().

Referenced by main().

**8.62.3.22 int mxmlSaveFile (mxml\_node\_t \* *node*, FILE \* *fp*, const char \*(\*) (mxml\_node\_t \**node*, int *ws*) *cb*)**

Definition at line 362 of file mxml-file.c.

References mxml\_file\_putc(), and mxml\_write\_node().

Referenced by main().

**8.62.3.23 int mxmlSaveString (mxml\_node\_t \* *node*, char \* *buffer*, int *bufsize*, const char \*(\*) (mxml\_node\_t \**node*, int *ws*) *cb*)**

Definition at line 404 of file mxml-file.c.

References mxml\_string\_putc(), and mxml\_write\_node().

Referenced by main(), and mxmlSaveAllocString().

**8.62.3.24 void mxmlSetCustomHandlers (mxml\_custom\_load\_cb\_t *load*, mxml\_custom\_save\_cb\_t *save*)**

Definition at line 456 of file mxml-file.c.

References mxml\_custom\_load\_cb, and mxml\_custom\_save\_cb.

**8.62.3.25 void mxmlSetErrorCallback (void(\*) (const char \*) cb)**

Definition at line 471 of file mxml-file.c.

References mxml\_error\_cb.

**8.62.4 Variable Documentation****8.62.4.1 mxml\_custom\_load\_cb\_t mxml\_custom\_load\_cb = NULL [static]**

Definition at line 103 of file mxml-file.c.

Referenced by mxml\_load\_data(), and mxmlSetCustomHandlers().

**8.62.4.2 mxml\_custom\_save\_cb\_t mxml\_custom\_save\_cb = NULL [static]**

Definition at line 104 of file mxml-file.c.

Referenced by mxml\_write\_node(), and mxmlSetCustomHandlers().

**8.62.4.3 void(\* mxml\_error\_cb)(const char \*)**

Referenced by mxml\_error(), and mxmlSetErrorCallback().

## 8.63 /home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-index.c File Reference

```
#include "config.h"
#include "mxml.h"
```

### Functions

- static [int index\\_compare](#) ([mxml\\_index\\_t](#) \*ind, [mxml\\_node\\_t](#) \*first, [mxml\\_node\\_t](#) \*second)
- static [int index\\_find](#) ([mxml\\_index\\_t](#) \*ind, const char \*element, const char \*value, [mxml\\_node\\_t](#) \*node)
- static void [index\\_sort](#) ([mxml\\_index\\_t](#) \*ind, int left, int right)
- void [mxmlIndexDelete](#) ([mxml\\_index\\_t](#) \*ind)
- [mxml\\_node\\_t](#) \* [mxmlIndexEnum](#) ([mxml\\_index\\_t](#) \*ind)
- [mxml\\_node\\_t](#) \* [mxmlIndexFind](#) ([mxml\\_index\\_t](#) \*ind, const char \*element, const char \*value)
- [mxml\\_index\\_t](#) \* [mxmlIndexNew](#) ([mxml\\_node\\_t](#) \*node, const char \*element, const char \*attr)
- [mxml\\_node\\_t](#) \* [mxmlIndexReset](#) ([mxml\\_index\\_t](#) \*ind)

### 8.63.1 Function Documentation

**8.63.1.1** static int [index\\_compare](#) ([mxml\\_index\\_t](#) \* *ind*, [mxml\\_node\\_t](#) \* *first*, [mxml\\_node\\_t](#) \* *second*) [static]

Definition at line 494 of file [mxml-index.c](#).

References [mxml\\_index\\_s::attr](#), [mxml\\_value\\_u::element](#), [mxmlElementGetAttr\(\)](#), [mxml\\_value\\_s::name](#), and [mxml\\_node\\_s::value](#).

Referenced by [index\\_sort\(\)](#).

**8.63.1.2** static int [index\\_find](#) ([mxml\\_index\\_t](#) \* *ind*, const char \* *element*, const char \* *value*, [mxml\\_node\\_t](#) \* *node*) [static]

Definition at line 533 of file [mxml-index.c](#).

References [mxml\\_index\\_s::attr](#), [mxml\\_value\\_u::element](#), [mxmlElementGetAttr\(\)](#), [mxml\\_value\\_s::name](#), and [mxml\\_node\\_s::value](#).

Referenced by [mxmlIndexFind\(\)](#).

**8.63.1.3** static void [index\\_sort](#) ([mxml\\_index\\_t](#) \* *ind*, int *left*, int *right*) [static]

Definition at line 576 of file [mxml-index.c](#).

References [index\\_compare\(\)](#), and [mxml\\_index\\_s::nodes](#).

Referenced by [mxmlIndexNew\(\)](#).

**8.63.1.4** void [mxmlIndexDelete](#) ([mxml\\_index\\_t](#) \* *ind*)

Definition at line 58 of file [mxml-index.c](#).

References `mxml_index_s::alloc_nodes`, `mxml_index_s::attr`, and `mxml_index_s::nodes`.

Referenced by `main()`, and `mxmlIndexNew()`.

#### 8.63.1.5 `mxml_node_t* mxmlIndexEnum (mxml_index_t * ind)`

Definition at line 88 of file `mxml-index.c`.

References `mxml_index_s::cur_node`, `mxml_index_s::nodes`, and `mxml_index_s::num_nodes`.

Referenced by `main()`, and `mxmlIndexFind()`.

#### 8.63.1.6 `mxml_node_t* mxmlIndexFind (mxml_index_t * ind, const char * element, const char * value)`

Definition at line 118 of file `mxml-index.c`.

References `mxml_index_s::attr`, `mxml_index_s::cur_node`, `index_find()`, `mxmlIndexEnum()`, `mxml_index_s::nodes`, and `mxml_index_s::num_nodes`.

Referenced by `main()`.

#### 8.63.1.7 `mxml_index_t* mxmlIndexNew (mxml_node_t * node, const char * element, const char * attr)`

Definition at line 301 of file `mxml-index.c`.

References `mxml_index_s::alloc_nodes`, `mxml_index_s::attr`, `mxml_value_u::element`, `index_sort()`, `MXML_DESCEND`, `mxml_error()`, `mxmlElementGetAttr()`, `mxmlFindElement()`, `mxmlIndexDelete()`, `mxml_value_s::name`, `mxml_index_s::nodes`, `mxml_index_s::num_nodes`, and `mxml_node_s::value`.

Referenced by `main()`.

#### 8.63.1.8 `mxml_node_t* mxmlIndexReset (mxml_index_t * ind)`

Definition at line 459 of file `mxml-index.c`.

References `mxml_index_s::cur_node`, `mxml_index_s::nodes`, and `mxml_index_s::num_nodes`.

Referenced by `main()`.



## 8.64 /home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-node.c File Reference

```
#include "config.h"
#include "mxml.h"
```

### Functions

- static `mxml_node_t * mxml_new` (`mxml_node_t *parent`, `mxml_type_t type`)
- void `mxmlAdd` (`mxml_node_t *parent`, `int where`, `mxml_node_t *child`, `mxml_node_t *node`)
- void `mxmlDelete` (`mxml_node_t *node`)
- `mxml_node_t * mxmlNewCustom` (`mxml_node_t *parent`, `void *data`, `void(*destroy)(void *)`)
- `mxml_node_t * mxmlNewElement` (`mxml_node_t *parent`, `const char *name`)
- `mxml_node_t * mxmlNewInteger` (`mxml_node_t *parent`, `int integer`)
- `mxml_node_t * mxmlNewOpaque` (`mxml_node_t *parent`, `const char *opaque`)
- `mxml_node_t * mxmlNewReal` (`mxml_node_t *parent`, `double real`)
- `mxml_node_t * mxmlNewText` (`mxml_node_t *parent`, `int whitespace`, `const char *string`)
- `mxml_node_t * mxmlNewTextf` (`mxml_node_t *parent`, `int whitespace`, `const char *format,...`)
- void `mxmlRemove` (`mxml_node_t *node`)

### 8.64.1 Function Documentation

#### 8.64.1.1 static `mxml_node_t * mxml_new` (`mxml_node_t *parent`, `mxml_type_t type`) [static]

Definition at line 617 of file `mxml-node.c`.

References `MXML_ADD_AFTER`, `MXML_ADD_TO_PARENT`, `mxmlAdd()`, `node`, and `mxml_node_s::type`.

Referenced by `mxmlNewCustom()`, `mxmlNewElement()`, `mxmlNewInteger()`, `mxmlNewOpaque()`, `mxmlNewReal()`, `mxmlNewText()`, and `mxmlNewTextf()`.

#### 8.64.1.2 void `mxmlAdd` (`mxml_node_t *parent`, `int where`, `mxml_node_t *child`, `mxml_node_t *node`)

Definition at line 62 of file `mxml-node.c`.

References `mxml_node_s::child`, `mxml_node_s::last_child`, `MXML_ADD_AFTER`, `MXML_ADD_BEFORE`, `mxmlRemove()`, `mxml_node_s::next`, `mxml_node_s::parent`, and `mxml_node_s::prev`.

Referenced by `add_variable()`, `agent_xml_compose()`, `agent_xml_compose__agent_data()`, `agent_xml_compose__create_row_nodes()`, `agent_xml_compose__data()`, `agent_xml_compose__gaf_message()`, `agent_xml_compose__message()`, `agent_xml_compose__mobile_agent()`, `agent_xml_compose__task()`, `agent_xml_compose__tasks()`, `fipa_envelope_Compose()`, `fipa_envelope_Compose__envelope()`, `fipa_envelope_Compose__params()`, `mxml_new()`, `scan_file()`, and `sort_node()`.

#### 8.64.1.3 void `mxmlDelete` (`mxml_node_t *node`)

Definition at line 196 of file `mxml-node.c`.

References `mxml_value_s::attrs`, `mxml_node_s::child`, `mxml_value_u::custom`, `mxml_custom_s::data`, `mxml_custom_s::destroy`, `mxml_value_u::element`, `MXML_CUSTOM`, `MXML_ELEMENT`, `MXML_INTEGER`, `MXML_OPAQUE`, `MXML_REAL`, `MXML_TEXT`, `mxmlDelete()`, `mxmlRemove()`, `mxml_attr_s::name`, `mxml_value_s::name`, `mxml_value_s::num_attrs`, `mxml_value_u::opaque`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::type`, `mxml_attr_s::value`, and `mxml_node_s::value`.

Referenced by `add_variable()`, `agent_datastate_Destroy()`, `fipa_envelope_Compose()`, `fipa_envelope_Parse()`, `main()`, `message_Destroy()`, `mxml_load_data()`, `mxmlDelete()`, `scan_file()`, and `sort_node()`.

#### 8.64.1.4 `mxml_node_t* mxmlNewCustom (mxml_node_t * parent, void * data, void(*) (void *) destroy)`

Definition at line 287 of file `mxml-node.c`.

References `mxml_value_u::custom`, `mxml_custom_s::data`, `mxml_custom_s::destroy`, `MXML_CUSTOM`, `mxml_new()`, `node`, and `mxml_node_s::value`.

Referenced by `mxml_load_data()`.

#### 8.64.1.5 `mxml_node_t* mxmlNewElement (mxml_node_t * parent, const char * name)`

Definition at line 323 of file `mxml-node.c`.

References `mxml_value_u::element`, `MXML_ELEMENT`, `mxml_new()`, `mxml_value_s::name`, `node`, and `mxml_node_s::value`.

Referenced by `add_variable()`, `agent_xml_compose__agent_code()`, `agent_xml_compose__agent_data()`, `agent_xml_compose__create_row_nodes()`, `agent_xml_compose__data()`, `agent_xml_compose__gaf_message()`, `agent_xml_compose__home()`, `agent_xml_compose__message()`, `agent_xml_compose__mobile_agent()`, `agent_xml_compose__name()`, `agent_xml_compose__owner()`, `agent_xml_compose__task()`, `agent_xml_compose__tasks()`, `fipa_envelope_Compose__acl_representation()`, `fipa_envelope_Compose__date()`, `fipa_envelope_Compose__envelope()`, `fipa_envelope_Compose__from()`, `fipa_envelope_Compose__intended_receiver()`, `fipa_envelope_Compose__params()`, `fipa_envelope_Compose__payload_encoding()`, `fipa_envelope_Compose__to()`, `main()`, `mxml_load_data()`, `scan_file()`, and `xml_new_cdata()`.

#### 8.64.1.6 `mxml_node_t* mxmlNewInteger (mxml_node_t * parent, int integer)`

Definition at line 361 of file `mxml-node.c`.

References `mxml_value_u::integer`, `MXML_INTEGER`, `mxml_new()`, `node`, and `mxml_node_s::value`.

Referenced by `main()`, and `mxml_load_data()`.

#### 8.64.1.7 `mxml_node_t* mxmlNewOpaque (mxml_node_t * parent, const char * opaque)`

Definition at line 392 of file `mxml-node.c`.

References `mxml_new()`, `MXML_OPAQUE`, `node`, `mxml_value_u::opaque`, and `mxml_node_s::value`.

Referenced by `main()`, and `mxml_load_data()`.

#### 8.64.1.8 `mxml_node_t* mxmlNewReal (mxml_node_t * parent, double real)`

Definition at line 430 of file `mxml-node.c`.

References `mxml_new()`, `MXML_REAL`, `node`, `mxml_value_u::real`, and `mxml_node_s::value`.

Referenced by `main()`, and `mxml_load_data()`.

#### 8.64.1.9 `mxml_node_t* mxmlNewText (mxml_node_t * parent, int whitespace, const char * string)`

Definition at line 462 of file `mxml-node.c`.

References `mxml_new()`, `MXML_TEXT`, `node`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::value`, and `mxml_text_s::whitespace`.

Referenced by `agent_xml_compose__create_row_nodes()`, `agent_xml_compose__home()`, `agent_xml_compose__name()`, `agent_xml_compose__owner()`, `fipa_envelope_Compose__acl_representation()`, `fipa_envelope_Compose__date()`, `fipa_envelope_Compose__from()`, `fipa_envelope_Compose__intended_receiver()`, `fipa_envelope_Compose__payload_encoding()`, `fipa_envelope_Compose__to()`, `main()`, `mxml_load_data()`, and `scan_file()`.

#### 8.64.1.10 `mxml_node_t* mxmlNewTextf (mxml_node_t * parent, int whitespace, const char * format, ...)`

Definition at line 506 of file `mxml-node.c`.

References `mxml_new()`, `mxml_strdupf()`, `MXML_TEXT`, `node`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::value`, and `mxml_text_s::whitespace`.

#### 8.64.1.11 `void mxmlRemove (mxml_node_t * node)`

Definition at line 553 of file `mxml-node.c`.

References `mxml_node_s::child`, `mxml_node_s::last_child`, `mxml_node_s::next`, `mxml_node_s::parent`, and `mxml_node_s::prev`.

Referenced by `mxmlAdd()`, and `mxmlDelete()`.

## 8.65 /home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-private.c File Reference

```
#include "config.h"
#include "mxml.h"
```

### Functions

- void [mxml\\_error](#) (const char \*format,...)
- [mxml\\_type\\_t mxml\\_integer\\_cb](#) ([mxml\\_node\\_t](#) \*node)
- [mxml\\_type\\_t mxml\\_opaque\\_cb](#) ([mxml\\_node\\_t](#) \*node)
- [mxml\\_type\\_t mxml\\_real\\_cb](#) ([mxml\\_node\\_t](#) \*node)

### Variables

- void(\* [mxml\\_error\\_cb](#) )(const char \*) = NULL

### 8.65.1 Function Documentation

#### 8.65.1.1 void [mxml\\_error](#) (const char \**format*, ...)

Definition at line 49 of file `mxml-private.c`.

References [mxml\\_error\\_cb](#), and [mxml\\_strdup\(\)](#).

Referenced by [mxml\\_add\\_char\(\)](#), [mxml\\_fd\\_getc\(\)](#), [mxml\\_file\\_getc\(\)](#), [mxml\\_get\\_entity\(\)](#), [mxml\\_load\\_data\(\)](#), [mxml\\_parse\\_element\(\)](#), [mxml\\_string\\_getc\(\)](#), [mxmlElementSetAttr\(\)](#), [mxmlEntityAddCallback\(\)](#), and [mxmlIndexNew\(\)](#).

#### 8.65.1.2 [mxml\\_type\\_t mxml\\_integer\\_cb](#) ([mxml\\_node\\_t](#) \**node*)

Definition at line 95 of file `mxml-private.c`.

References `MXML_INTEGER`.

#### 8.65.1.3 [mxml\\_type\\_t mxml\\_opaque\\_cb](#) ([mxml\\_node\\_t](#) \**node*)

Definition at line 108 of file `mxml-private.c`.

References `MXML_OPAQUE`.

#### 8.65.1.4 [mxml\\_type\\_t mxml\\_real\\_cb](#) ([mxml\\_node\\_t](#) \**node*)

Definition at line 121 of file `mxml-private.c`.

References `MXML_REAL`.

## 8.65.2 Variable Documentation

### 8.65.2.1 `void(* mxml_error_cb)(const char *) = NULL`

Referenced by `mxml_error()`, and `mxmlSetErrorCallback()`.

## 8.66 /home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-search.c File Reference

```
#include "config.h"
#include "mxml.h"
```

### Functions

- [mxml\\_node\\_t \\* mxmlFindElement](#) ([mxml\\_node\\_t \\* node](#), [mxml\\_node\\_t \\* top](#), [const char \\* name](#), [const char \\* attr](#), [const char \\* value](#), [int descend](#))
- [mxml\\_node\\_t \\* mxmlWalkNext](#) ([mxml\\_node\\_t \\* node](#), [mxml\\_node\\_t \\* top](#), [int descend](#))
- [mxml\\_node\\_t \\* mxmlWalkPrev](#) ([mxml\\_node\\_t \\* node](#), [mxml\\_node\\_t \\* top](#), [int descend](#))

### 8.66.1 Function Documentation

#### 8.66.1.1 [mxml\\_node\\_t \\* mxmlFindElement](#) ([mxml\\_node\\_t \\* node](#), [mxml\\_node\\_t \\* top](#), [const char \\* name](#), [const char \\* attr](#), [const char \\* value](#), [int descend](#))

Definition at line 51 of file `mxml-search.c`.

References `mxml_value_u::element`, `MXML_DESCEND`, `MXML_ELEMENT`, `mxmlElementGetAttr()`, `mxmlWalkNext()`, `mxml_value_s::name`, `mxml_node_s::next`, `mxml_node_s::type`, and `mxml_node_s::value`.

Referenced by `agent_xml_parse__fill_row_data()`, `agent_xml_parse__task()`, `agent_xml_parse__tasks()`, `fipa_envelope_HandleAclRepresentation()`, `fipa_envelope_HandleComments()`, `fipa_envelope_HandleDate()`, `fipa_envelope_HandleEnvelope()`, `fipa_envelope_HandleFrom()`, `fipa_envelope_HandleIntendedReceiver()`, `fipa_envelope_HandleParams()`, `fipa_envelope_HandlePayloadEncoding()`, `fipa_envelope_HandlePayloadLength()`, `fipa_envelope_HandleReceived()`, `fipa_envelope_HandleTo()`, `fipa_envelope_ParseAddresses()`, `fipa_envelope_ParseAgentIdentifier()`, `fipa_envelope_ParseResolvers()`, `main()`, `MC_LoadAgentFromFile()`, `mxmlIndexNew()`, `scan_file()`, `sort_node()`, `write_documentation()`, `write_element()`, `xml_find_sibling()`, and `xml_get_child()`.

#### 8.66.1.2 [mxml\\_node\\_t \\* mxmlWalkNext](#) ([mxml\\_node\\_t \\* node](#), [mxml\\_node\\_t \\* top](#), [int descend](#))

Definition at line 133 of file `mxml-search.c`.

References `mxml_node_s::child`, `mxml_node_s::next`, and `mxml_node_s::parent`.

Referenced by `mxmlFindElement()`, and `write_element()`.

#### 8.66.1.3 [mxml\\_node\\_t \\* mxmlWalkPrev](#) ([mxml\\_node\\_t \\* node](#), [mxml\\_node\\_t \\* top](#), [int descend](#))

Definition at line 169 of file `mxml-search.c`.

References `mxml_node_s::last_child`, `mxml_node_s::parent`, and `mxml_node_s::prev`.

## 8.67 /home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-set.c File Reference

```
#include "config.h"
#include "mxml.h"
```

### Functions

- `int mxmlSetCustom (mxml_node_t *node, void *data, void(*destroy)(void *))`
- `int mxmlSetElement (mxml_node_t *node, const char *name)`
- `int mxmlSetInteger (mxml_node_t *node, int integer)`
- `int mxmlSetOpaque (mxml_node_t *node, const char *opaque)`
- `int mxmlSetReal (mxml_node_t *node, double real)`
- `int mxmlSetText (mxml_node_t *node, int whitespace, const char *string)`
- `int mxmlSetTextf (mxml_node_t *node, int whitespace, const char *format,...)`

### 8.67.1 Function Documentation

#### 8.67.1.1 `int mxmlSetCustom (mxml_node_t * node, void * data, void(*) (void *) destroy)`

Definition at line 46 of file mxml-set.c.

References `mxml_value_u::custom`, `mxml_custom_s::data`, `mxml_custom_s::destroy`, `MXML_CUSTOM`, `mxml_node_s::type`, and `mxml_node_s::value`.

#### 8.67.1.2 `int mxmlSetElement (mxml_node_t * node, const char * name)`

Definition at line 79 of file mxml-set.c.

References `mxml_value_u::element`, `MXML_ELEMENT`, `mxml_value_s::name`, `mxml_node_s::type`, and `mxml_node_s::value`.

#### 8.67.1.3 `int mxmlSetInteger (mxml_node_t * node, int integer)`

Definition at line 109 of file mxml-set.c.

References `mxml_value_u::integer`, `MXML_INTEGER`, `mxml_node_s::type`, and `mxml_node_s::value`.

#### 8.67.1.4 `int mxmlSetOpaque (mxml_node_t * node, const char * opaque)`

Definition at line 136 of file mxml-set.c.

References `MXML_OPAQUE`, `mxml_value_u::opaque`, `mxml_node_s::type`, and `mxml_node_s::value`.

#### 8.67.1.5 `int mxmlSetReal (mxml_node_t * node, double real)`

Definition at line 166 of file mxml-set.c.

References `MXML_REAL`, `mxml_value_u::real`, `mxml_node_s::type`, and `mxml_node_s::value`.

**8.67.1.6 int mxmlSetText (mxml\_node\_t \* *node*, int *whitespace*, const char \* *string*)**

Definition at line 193 of file mxml-set.c.

References MXML\_TEXT, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, mxml\_node\_s::value, and mxml\_text\_s::whitespace.

**8.67.1.7 int mxmlSetTextf (mxml\_node\_t \* *node*, int *whitespace*, const char \* *format*, ...)**

Definition at line 225 of file mxml-set.c.

References mxml\_strdupf(), MXML\_TEXT, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, mxml\_node\_s::value, and mxml\_text\_s::whitespace.



## 8.68 /home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml-string.c File Reference

```
#include "config.h"
```

### Functions

- `char * mxml_strdup` (`const char *format, va_list ap`)
- `int mxml_vsnprintf` (`char *buffer, size_t bufsz, const char *format, va_list ap`)

### 8.68.1 Function Documentation

#### 8.68.1.1 `char* mxml_strdup (const char *format, va_list ap)`

Definition at line 62 of file mxml-string.c.

References `mxml_vsnprintf()`.

Referenced by `mxml_error()`, `mxmlNewTextf()`, and `mxmlSetTextf()`.

#### 8.68.1.2 `int mxml_vsnprintf (char *buffer, size_t bufsize, const char *format, va_list ap)`

Definition at line 107 of file mxml-string.c.

References `size`.

Referenced by `mxml_strdup()`.

## 8.69 /home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxml.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <errno.h>
```

### Data Structures

- struct [mxml\\_attr\\_s](#)
- struct [mxml\\_value\\_s](#)
- struct [mxml\\_text\\_s](#)
- struct [mxml\\_custom\\_s](#)
- union [mxml\\_value\\_u](#)
- struct [mxml\\_node\\_s](#)
- struct [mxml\\_index\\_s](#)

### Defines

- #define [MXML\\_WRAP](#) 72
- #define [MXML\\_TAB](#) 8
- #define [MXML\\_NO\\_CALLBACK](#) 0
- #define [MXML\\_INTEGER\\_CALLBACK](#) [mxml\\_integer\\_cb](#)
- #define [MXML\\_OPAQUE\\_CALLBACK](#) [mxml\\_opaque\\_cb](#)
- #define [MXML\\_REAL\\_CALLBACK](#) [mxml\\_real\\_cb](#)
- #define [MXML\\_TEXT\\_CALLBACK](#) 0
- #define [MXML\\_NO\\_PARENT](#) 0
- #define [MXML\\_DESCEND](#) 1
- #define [MXML\\_NO\\_DESCEND](#) 0
- #define [MXML\\_DESCEND\\_FIRST](#) -1
- #define [MXML\\_WS\\_BEFORE\\_OPEN](#) 0
- #define [MXML\\_WS\\_AFTER\\_OPEN](#) 1
- #define [MXML\\_WS\\_BEFORE\\_CLOSE](#) 2
- #define [MXML\\_WS\\_AFTER\\_CLOSE](#) 3
- #define [MXML\\_ADD\\_BEFORE](#) 0
- #define [MXML\\_ADD\\_AFTER](#) 1
- #define [MXML\\_ADD\\_TO\\_PARENT](#) NULL

### Typedefs

- typedef enum [mxml\\_type\\_e](#) [mxml\\_type\\_t](#)
- typedef struct [mxml\\_attr\\_s](#) [mxml\\_attr\\_t](#)
- typedef struct [mxml\\_value\\_s](#) [mxml\\_element\\_t](#)
- typedef struct [mxml\\_text\\_s](#) [mxml\\_text\\_t](#)
- typedef struct [mxml\\_custom\\_s](#) [mxml\\_custom\\_t](#)

- typedef union [mxml\\_value\\_u](#) [mxml\\_value\\_t](#)
- typedef struct [mxml\\_node\\_s](#) [mxml\\_node\\_t](#)
- typedef struct [mxml\\_index\\_s](#) [mxml\\_index\\_t](#)
- typedef [int](#)(\* [mxml\\_custom\\_load\\_cb\\_t](#))([mxml\\_node\\_t](#) \*, const char \*)
- typedef char \*(\* [mxml\\_custom\\_save\\_cb\\_t](#))([mxml\\_node\\_t](#) \*)

## Enumerations

- enum [mxml\\_type\\_e](#) {  
[MXML\\_ELEMENT](#), [MXML\\_INTEGER](#), [MXML\\_OPAQUE](#), [MXML\\_REAL](#),  
[MXML\\_TEXT](#), [MXML\\_CUSTOM](#) }

## Functions

- void [mxmlAdd](#) ([mxml\\_node\\_t](#) \*parent, [int](#) where, [mxml\\_node\\_t](#) \*child, [mxml\\_node\\_t](#) \*node)
- void [mxmlDelete](#) ([mxml\\_node\\_t](#) \*node)
- const char \* [mxmlElementGetAttr](#) ([mxml\\_node\\_t](#) \*node, const char \*name)
- void [mxmlElementSetAttr](#) ([mxml\\_node\\_t](#) \*node, const char \*name, const char \*value)
- [int](#) [mxmlEntityAddCallback](#) ([int](#)(\*cb)(const char \*name))
- const char \* [mxmlEntityGetName](#) ([int](#) val)
- [int](#) [mxmlEntityGetValue](#) (const char \*name)
- void [mxmlEntityRemoveCallback](#) ([int](#)(\*cb)(const char \*name))
- [mxml\\_node\\_t](#) \* [mxmlFindElement](#) ([mxml\\_node\\_t](#) \*node, [mxml\\_node\\_t](#) \*top, const char \*name, const char \*attr, const char \*value, [int](#) descend)
- void [mxmlIndexDelete](#) ([mxml\\_index\\_t](#) \*ind)
- [mxml\\_node\\_t](#) \* [mxmlIndexEnum](#) ([mxml\\_index\\_t](#) \*ind)
- [mxml\\_node\\_t](#) \* [mxmlIndexFind](#) ([mxml\\_index\\_t](#) \*ind, const char \*element, const char \*value)
- [mxml\\_index\\_t](#) \* [mxmlIndexNew](#) ([mxml\\_node\\_t](#) \*node, const char \*element, const char \*attr)
- [mxml\\_node\\_t](#) \* [mxmlIndexReset](#) ([mxml\\_index\\_t](#) \*ind)
- [mxml\\_node\\_t](#) \* [mxmlLoadFd](#) ([mxml\\_node\\_t](#) \*top, [int](#) fd, [mxml\\_type\\_t](#)(\*cb)([mxml\\_node\\_t](#) \*))
- [mxml\\_node\\_t](#) \* [mxmlLoadFile](#) ([mxml\\_node\\_t](#) \*top, FILE \*fp, [mxml\\_type\\_t](#)(\*cb)([mxml\\_node\\_t](#) \*))
- [mxml\\_node\\_t](#) \* [mxmlLoadString](#) ([mxml\\_node\\_t](#) \*top, const char \*s, [mxml\\_type\\_t](#)(\*cb)([mxml\\_node\\_t](#) \*))
- [mxml\\_node\\_t](#) \* [mxmlNewCustom](#) ([mxml\\_node\\_t](#) \*parent, void \*data, void(\*destroy)(void \*))
- [mxml\\_node\\_t](#) \* [mxmlNewElement](#) ([mxml\\_node\\_t](#) \*parent, const char \*name)
- [mxml\\_node\\_t](#) \* [mxmlNewInteger](#) ([mxml\\_node\\_t](#) \*parent, [int](#) integer)
- [mxml\\_node\\_t](#) \* [mxmlNewOpaque](#) ([mxml\\_node\\_t](#) \*parent, const char \*opaque)
- [mxml\\_node\\_t](#) \* [mxmlNewReal](#) ([mxml\\_node\\_t](#) \*parent, double real)
- [mxml\\_node\\_t](#) \* [mxmlNewText](#) ([mxml\\_node\\_t](#) \*parent, [int](#) whitespace, const char \*string)
- [mxml\\_node\\_t](#) \* [mxmlNewTextf](#) ([mxml\\_node\\_t](#) \*parent, [int](#) whitespace, const char \*format,...)
- void [mxmlRemove](#) ([mxml\\_node\\_t](#) \*node)
- char \* [mxmlSaveAllocString](#) ([mxml\\_node\\_t](#) \*node, const char \*(\*cb)([mxml\\_node\\_t](#) \*, [int](#)))
- [int](#) [mxmlSaveFd](#) ([mxml\\_node\\_t](#) \*node, [int](#) fd, const char \*(\*cb)([mxml\\_node\\_t](#) \*, [int](#)))
- [int](#) [mxmlSaveFile](#) ([mxml\\_node\\_t](#) \*node, FILE \*fp, const char \*(\*cb)([mxml\\_node\\_t](#) \*, [int](#)))
- [int](#) [mxmlSaveString](#) ([mxml\\_node\\_t](#) \*node, char \*buffer, [int](#) bufsize, const char \*(\*cb)([mxml\\_node\\_t](#) \*, [int](#)))
- [int](#) [mxmlSetCustom](#) ([mxml\\_node\\_t](#) \*node, void \*data, void(\*destroy)(void \*))
- void [mxmlSetCustomHandlers](#) ([mxml\\_custom\\_load\\_cb\\_t](#) load, [mxml\\_custom\\_save\\_cb\\_t](#) save)
- [int](#) [mxmlSetElement](#) ([mxml\\_node\\_t](#) \*node, const char \*name)

- void `mxmlSetErrorCallback` (void(\*cb)(const char \*))
- int `mxmlSetInteger` (mxml\_node\_t \*node, int integer)
- int `mxmlSetOpaque` (mxml\_node\_t \*node, const char \*opaque)
- int `mxmlSetReal` (mxml\_node\_t \*node, double real)
- int `mxmlSetText` (mxml\_node\_t \*node, int whitespace, const char \*string)
- int `mxmlSetTextf` (mxml\_node\_t \*node, int whitespace, const char \*format,...)
- mxml\_node\_t \* `mxmlWalkNext` (mxml\_node\_t \*node, mxml\_node\_t \*top, int descend)
- mxml\_node\_t \* `mxmlWalkPrev` (mxml\_node\_t \*node, mxml\_node\_t \*top, int descend)
- void `mxml_error` (const char \*format,...)
- mxml\_type\_t `mxml_integer_cb` (mxml\_node\_t \*node)
- mxml\_type\_t `mxml_opaque_cb` (mxml\_node\_t \*node)
- mxml\_type\_t `mxml_real_cb` (mxml\_node\_t \*node)

## 8.69.1 Define Documentation

### 8.69.1.1 #define MXML\_ADD\_AFTER 1

Definition at line 68 of file `mxml.h`.

Referenced by `add_variable()`, `agent_xml_compose()`, `agent_xml_compose__agent_data()`, `agent_xml_compose__create_row_nodes()`, `agent_xml_compose__data()`, `agent_xml_compose__gaf_message()`, `agent_xml_compose__message()`, `agent_xml_compose__mobile_agent()`, `agent_xml_compose__task()`, `agent_xml_compose__tasks()`, `fipa_envelope_Compose()`, `fipa_envelope_Compose__envelope()`, `fipa_envelope_Compose__params()`, `mxml_new()`, `mxmlAdd()`, `scan_file()`, and `sort_node()`.

### 8.69.1.2 #define MXML\_ADD\_BEFORE 0

Definition at line 67 of file `mxml.h`.

Referenced by `mxmlAdd()`, `scan_file()`, and `sort_node()`.

### 8.69.1.3 #define MXML\_ADD\_TO\_PARENT NULL

Definition at line 69 of file `mxml.h`.

Referenced by `add_variable()`, `agent_xml_compose()`, `agent_xml_compose__create_row_nodes()`, `fipa_envelope_Compose()`, `fipa_envelope_Compose__envelope()`, `fipa_envelope_Compose__params()`, `mxml_new()`, `scan_file()`, and `sort_node()`.

### 8.69.1.4 #define MXML\_DESCEND 1

Definition at line 58 of file `mxml.h`.

Referenced by `agent_xml_parse__tasks()`, `main()`, `MC_LoadAgentFromFile()`, `mxmlFindElement()`, `mxmlIndexNew()`, and `write_element()`.

### 8.69.1.5 #define MXML\_DESCEND\_FIRST -1

Definition at line 60 of file `mxml.h`.

Referenced by `agent_xml_parse__fill_row_data()`, `agent_xml_parse__task()`, `agent_xml_parse__tasks()`, `fipa_envelope_HandleAclRepresentation()`, `fipa_envelope_HandleComments()`, `fipa_envelope_HandleDate()`, `fipa_envelope_HandleEnvelope()`, `fipa_envelope_HandleFrom()`, `fipa_envelope_HandleIntendedReceiver()`, `fipa_envelope_HandleParams()`, `fipa_envelope_HandlePayloadEncoding()`, `fipa_envelope_HandlePayloadLength()`, `fipa_envelope_HandleReceived()`, `fipa_envelope_HandleTo()`, `fipa_envelope_ParseAddresses()`, `fipa_envelope_ParseAgentIdentifier()`, `fipa_envelope_ParseResolvers()`, `scan_file()`, `sort_node()`, and `write_documentation()`.

#### 8.69.1.6 `#define MXML_INTEGER_CALLBACK mxml_integer_cb`

Definition at line 48 of file `mxml.h`.

Referenced by `main()`.

#### 8.69.1.7 `#define MXML_NO_CALLBACK 0`

Definition at line 47 of file `mxml.h`.

Referenced by `acc_Thread()`, `agent_xml_compose()`, `fipa_envelope_Compose()`, `fipa_envelope_Parse()`, `main()`, `message_InitializeFromAgent()`, and `message_InitializeFromConnection()`.

#### 8.69.1.8 `#define MXML_NO_DESCEND 0`

Definition at line 59 of file `mxml.h`.

Referenced by `agent_xml_parse__task()`, `agent_xml_parse__tasks()`, `fipa_envelope_HandleIntendedReceiver()`, `fipa_envelope_HandleTo()`, `fipa_envelope_ParseAddresses()`, `fipa_envelope_ParseResolvers()`, `main()`, `write_documentation()`, `write_element()`, `xml_find_sibling()`, and `xml_get_deep_child()`.

#### 8.69.1.9 `#define MXML_NO_PARENT 0`

Definition at line 56 of file `mxml.h`.

Referenced by `agent_xml_compose__agent_code()`, `agent_xml_compose__create_row_nodes()`, `main()`, and `scan_file()`.

#### 8.69.1.10 `#define MXML_OPAQUE_CALLBACK mxml_opaque_cb`

Definition at line 50 of file `mxml.h`.

Referenced by `main()`.

#### 8.69.1.11 `#define MXML_REAL_CALLBACK mxml_real_cb`

Definition at line 52 of file `mxml.h`.

Referenced by `main()`.

#### 8.69.1.12 `#define MXML_TAB 8`

Definition at line 45 of file `mxml.h`.

Referenced by `mxml_write_ws()`.

#### **8.69.1.13 #define MXML\_TEXT\_CALLBACK 0**

Definition at line 54 of file `mxml.h`.

#### **8.69.1.14 #define MXML\_WRAP 72**

Definition at line 44 of file `mxml.h`.

Referenced by `mxml_write_node()`.

#### **8.69.1.15 #define MXML\_WS\_AFTER\_CLOSE 3**

Definition at line 65 of file `mxml.h`.

Referenced by `mxml_write_node()`, `whitespace_cb()`, and `ws_cb()`.

#### **8.69.1.16 #define MXML\_WS\_AFTER\_OPEN 1**

Definition at line 63 of file `mxml.h`.

Referenced by `mxml_write_node()`, `whitespace_cb()`, and `ws_cb()`.

#### **8.69.1.17 #define MXML\_WS\_BEFORE\_CLOSE 2**

Definition at line 64 of file `mxml.h`.

Referenced by `mxml_write_node()`, `whitespace_cb()`, and `ws_cb()`.

#### **8.69.1.18 #define MXML\_WS\_BEFORE\_OPEN 0**

Definition at line 62 of file `mxml.h`.

Referenced by `mxml_write_node()`, `whitespace_cb()`, and `ws_cb()`.

### **8.69.2 Typedef Documentation**

#### **8.69.2.1 typedef struct mxml\_attr\_s mxml\_attr\_t**

#### **8.69.2.2 typedef int(\* mxml\_custom\_load\_cb\_t)(mxml\_node\_t \*, const char \*)**

Definition at line 142 of file `mxml.h`.

#### **8.69.2.3 typedef char\*(\*) mxml\_custom\_save\_cb\_t)(mxml\_node\_t \*)**

Definition at line 145 of file `mxml.h`.

**8.69.2.4** typedef struct mxml\_custom\_s mxml\_custom\_t

**8.69.2.5** typedef struct mxml\_value\_s mxml\_element\_t

**8.69.2.6** typedef struct mxml\_index\_s mxml\_index\_t

**8.69.2.7** typedef struct mxml\_node\_s mxml\_node\_t

**8.69.2.8** typedef struct mxml\_text\_s mxml\_text\_t

**8.69.2.9** typedef enum mxml\_type\_e mxml\_type\_t

**8.69.2.10** typedef union mxml\_value\_u mxml\_value\_t

### 8.69.3 Enumeration Type Documentation

**8.69.3.1** enum mxml\_type\_e

Enumerator:

*MXML\_ELEMENT*

*MXML\_INTEGER*

*MXML\_OPAQUE*

*MXML\_REAL*

*MXML\_TEXT*

*MXML\_CUSTOM*

Definition at line 76 of file mxml.h.

### 8.69.4 Function Documentation

**8.69.4.1** void mxml\_error (const char \* *format*, ...)

Definition at line 49 of file mxml-private.c.

References mxml\_error\_cb, and mxml\_strdup().

Referenced by mxml\_add\_char(), mxml\_fd\_getc(), mxml\_file\_getc(), mxml\_get\_entity(), mxml\_load\_data(), mxml\_parse\_element(), mxml\_string\_getc(), mxmlElementSetAttr(), mxmlEntityAddCallback(), and mxmlIndexNew().

**8.69.4.2** mxml\_type\_t mxml\_integer\_cb (mxml\_node\_t \* *node*)

Definition at line 95 of file mxml-private.c.

References MXML\_INTEGER.

**8.69.4.3** mxml\_type\_t mxml\_opaque\_cb (mxml\_node\_t \* *node*)

Definition at line 108 of file mxml-private.c.

References MXML\_OPAQUE.

**8.69.4.4 mxml\_type\_t mxml\_real\_cb (mxml\_node\_t \* node)**

Definition at line 121 of file mxml-private.c.

References MXML\_REAL.

**8.69.4.5 void mxmlAdd (mxml\_node\_t \* parent, int where, mxml\_node\_t \* child, mxml\_node\_t \* node)**

Definition at line 62 of file mxml-node.c.

References mxml\_node\_s::child, mxml\_node\_s::last\_child, MXML\_ADD\_AFTER, MXML\_ADD\_BEFORE, mxmlRemove(), mxml\_node\_s::next, mxml\_node\_s::parent, and mxml\_node\_s::prev.

Referenced by add\_variable(), agent\_xml\_compose(), agent\_xml\_compose\_\_agent\_data(), agent\_xml\_compose\_\_create\_row\_nodes(), agent\_xml\_compose\_\_data(), agent\_xml\_compose\_\_gaf\_message(), agent\_xml\_compose\_\_message(), agent\_xml\_compose\_\_mobile\_agent(), agent\_xml\_compose\_\_task(), agent\_xml\_compose\_\_tasks(), fipa\_envelope\_Compose(), fipa\_envelope\_Compose\_\_envelope(), fipa\_envelope\_Compose\_\_params(), mxml\_new(), scan\_file(), and sort\_node().

**8.69.4.6 void mxmlDelete (mxml\_node\_t \* node)**

Definition at line 196 of file mxml-node.c.

References mxml\_value\_s::attrs, mxml\_node\_s::child, mxml\_value\_u::custom, mxml\_custom\_s::data, mxml\_custom\_s::destroy, mxml\_value\_u::element, MXML\_CUSTOM, MXML\_ELEMENT, MXML\_INTEGER, MXML\_OPAQUE, MXML\_REAL, MXML\_TEXT, mxmlDelete(), mxmlRemove(), mxml\_attr\_s::name, mxml\_value\_s::name, mxml\_value\_s::num\_attrs, mxml\_value\_u::opaque, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, mxml\_attr\_s::value, and mxml\_node\_s::value.

Referenced by add\_variable(), agent\_datastate\_Destroy(), fipa\_envelope\_Compose(), fipa\_envelope\_Parse(), main(), message\_Destroy(), mxml\_load\_data(), mxmlDelete(), scan\_file(), and sort\_node().

**8.69.4.7 const char\* mxmlElementGetAttr (mxml\_node\_t \* node, const char \* name)**

Definition at line 43 of file mxml-attr.c.

References mxml\_value\_s::attrs, mxml\_value\_u::element, MXML\_ELEMENT, mxml\_attr\_s::name, mxml\_value\_s::num\_attrs, mxml\_node\_s::type, mxml\_attr\_s::value, and mxml\_node\_s::value.

Referenced by agent\_xml\_parse\_\_agent\_code(), agent\_xml\_parse\_\_data(), agent\_xml\_parse\_\_task(), agent\_xml\_parse\_\_tasks(), fipa\_envelope\_HandleReceived(), index\_compare(), index\_find(), mxml\_parse\_element(), mxmlFindElement(), mxmlIndexNew(), scan\_file(), sort\_node(), type\_cb(), and write\_documentation().

**8.69.4.8 void mxmlElementSetAttr (mxml\_node\_t \* node, const char \* name, const char \* value)**

Definition at line 90 of file mxml-attr.c.

References mxml\_value\_s::attrs, mxml\_value\_u::element, MXML\_ELEMENT, mxml\_error(), mxml\_value\_s::name, mxml\_attr\_s::name, mxml\_value\_s::num\_attrs, mxml\_node\_s::type, mxml\_attr\_s::value, and mxml\_node\_s::value.

Referenced by add\_variable(), agent\_xml\_compose\_\_agent\_code(), agent\_xml\_compose\_\_create\_row\_nodes(), agent\_xml\_compose\_\_data(), agent\_xml\_compose\_\_message(), agent\_xml\_compose\_\_task(),



agent\_xml\_compose\_\_tasks(), fipa\_envelope\_Compose\_\_params(), main(), mxml\_parse\_element(), scan\_file(), sort\_node(), and update\_comment().

#### 8.69.4.9 int mxmlEntityAddCallback (int(\*) (const char \*name) cb)

Definition at line 65 of file mxml-entity.c.

References callbacks, mxml\_error(), and num\_callbacks.

#### 8.69.4.10 const char\* mxmlEntityGetName (int val)

Definition at line 91 of file mxml-entity.c.

Referenced by mxml\_write\_name(), and mxml\_write\_string().

#### 8.69.4.11 int mxmlEntityGetValue (const char \* name)

Definition at line 121 of file mxml-entity.c.

References callbacks, and num\_callbacks.

Referenced by mxml\_get\_entity().

#### 8.69.4.12 void mxmlEntityRemoveCallback (int(\*) (const char \*name) cb)

Definition at line 140 of file mxml-entity.c.

References callbacks, and num\_callbacks.

#### 8.69.4.13 mxml\_node\_t\* mxmlFindElement (mxml\_node\_t \* node, mxml\_node\_t \* top, const char \* name, const char \* attr, const char \* value, int descend)

Definition at line 51 of file mxml-search.c.

References mxml\_value\_u::element, MXML\_DESCEND, MXML\_ELEMENT, mxmlElementGetAttr(), mxmlWalkNext(), mxml\_value\_s::name, mxml\_node\_s::next, mxml\_node\_s::type, and mxml\_node\_s::value.

Referenced by agent\_xml\_parse\_\_fill\_row\_data(), agent\_xml\_parse\_\_task(), agent\_xml\_parse\_\_tasks(), fipa\_envelope\_HandleAclRepresentation(), fipa\_envelope\_HandleComments(), fipa\_envelope\_HandleDate(), fipa\_envelope\_HandleEnvelope(), fipa\_envelope\_HandleFrom(), fipa\_envelope\_HandleIntendedReceiver(), fipa\_envelope\_HandleParams(), fipa\_envelope\_HandlePayloadEncoding(), fipa\_envelope\_HandlePayloadLength(), fipa\_envelope\_HandleReceived(), fipa\_envelope\_HandleTo(), fipa\_envelope\_ParseAddresses(), fipa\_envelope\_ParseAgentIdentifier(), fipa\_envelope\_ParseResolvers(), main(), MC\_LoadAgentFromFile(), mxmlIndexNew(), scan\_file(), sort\_node(), write\_documentation(), write\_element(), xml\_find\_sibling(), and xml\_get\_child().

#### 8.69.4.14 void mxmlIndexDelete (mxml\_index\_t \* ind)

Definition at line 58 of file mxml-index.c.

References mxml\_index\_s::alloc\_nodes, mxml\_index\_s::attr, and mxml\_index\_s::nodes.

Referenced by main(), and mxmlIndexNew().

**8.69.4.15 mxml\_node\_t\* mxmlIndexEnum (mxml\_index\_t \* *ind*)**

Definition at line 88 of file mxml-index.c.

References mxml\_index\_s::cur\_node, mxml\_index\_s::nodes, and mxml\_index\_s::num\_nodes.

Referenced by main(), and mxmlIndexFind().

**8.69.4.16 mxml\_node\_t\* mxmlIndexFind (mxml\_index\_t \* *ind*, const char \* *element*, const char \* *value*)**

Definition at line 118 of file mxml-index.c.

References mxml\_index\_s::attr, mxml\_index\_s::cur\_node, index\_find(), mxmlIndexEnum(), mxml\_index\_s::nodes, and mxml\_index\_s::num\_nodes.

Referenced by main().

**8.69.4.17 mxml\_index\_t\* mxmlIndexNew (mxml\_node\_t \* *node*, const char \* *element*, const char \* *attr*)**

Definition at line 301 of file mxml-index.c.

References mxml\_index\_s::alloc\_nodes, mxml\_index\_s::attr, mxml\_value\_u::element, index\_sort(), MXML\_DESCEND, mxml\_error(), mxmlElementGetAttr(), mxmlFindElement(), mxmlIndexDelete(), mxml\_value\_s::name, mxml\_index\_s::nodes, mxml\_index\_s::num\_nodes, and mxml\_node\_s::value.

Referenced by main().

**8.69.4.18 mxml\_node\_t\* mxmlIndexReset (mxml\_index\_t \* *ind*)**

Definition at line 459 of file mxml-index.c.

References mxml\_index\_s::cur\_node, mxml\_index\_s::nodes, and mxml\_index\_s::num\_nodes.

Referenced by main().

**8.69.4.19 mxml\_node\_t\* mxmlLoadFd (mxml\_node\_t \* *top*, int *fd*, mxml\_type\_t(\*) (mxml\_node\_t \*) *cb*)****8.69.4.20 mxml\_node\_t\* mxmlLoadFile (mxml\_node\_t \* *top*, FILE \* *fp*, mxml\_type\_t(\*) (mxml\_node\_t \*) *cb*)****8.69.4.21 mxml\_node\_t\* mxmlLoadString (mxml\_node\_t \* *top*, const char \* *s*, mxml\_type\_t(\*) (mxml\_node\_t \*) *cb*)****8.69.4.22 mxml\_node\_t\* mxmlNewCustom (mxml\_node\_t \* *parent*, void \* *data*, void(\*) (void \*) *destroy*)**

Definition at line 287 of file mxml-node.c.

References mxml\_value\_u::custom, mxml\_custom\_s::data, mxml\_custom\_s::destroy, MXML\_CUSTOM, mxml\_new(), node, and mxml\_node\_s::value.

Referenced by mxml\_load\_data().

**8.69.4.23 mxml\_node\_t\* mxmlNewElement (mxml\_node\_t \* *parent*, const char \* *name*)**

Definition at line 323 of file mxml-node.c.

References mxml\_value\_u::element, MXML\_ELEMENT, mxml\_new(), mxml\_value\_s::name, node, and mxml\_node\_s::value.

Referenced by add\_variable(), agent\_xml\_compose\_\_agent\_code(), agent\_xml\_compose\_\_agent\_data(), agent\_xml\_compose\_\_create\_row\_nodes(), agent\_xml\_compose\_\_data(), agent\_xml\_compose\_\_gaf\_message(), agent\_xml\_compose\_\_home(), agent\_xml\_compose\_\_message(), agent\_xml\_compose\_\_mobile\_agent(), agent\_xml\_compose\_\_name(), agent\_xml\_compose\_\_owner(), agent\_xml\_compose\_\_task(), agent\_xml\_compose\_\_tasks(), fipa\_envelope\_Compose\_\_acl\_representation(), fipa\_envelope\_Compose\_\_date(), fipa\_envelope\_Compose\_\_envelope(), fipa\_envelope\_Compose\_\_from(), fipa\_envelope\_Compose\_\_intended\_receiver(), fipa\_envelope\_Compose\_\_params(), fipa\_envelope\_Compose\_\_payload\_encoding(), fipa\_envelope\_Compose\_\_to(), main(), mxml\_load\_data(), scan\_file(), and xml\_new\_cdata().

**8.69.4.24 mxml\_node\_t\* mxmlNewInteger (mxml\_node\_t \* *parent*, int *integer*)**

Definition at line 361 of file mxml-node.c.

References mxml\_value\_u::integer, MXML\_INTEGER, mxml\_new(), node, and mxml\_node\_s::value.

Referenced by main(), and mxml\_load\_data().

**8.69.4.25 mxml\_node\_t\* mxmlNewOpaque (mxml\_node\_t \* *parent*, const char \* *opaque*)**

Definition at line 392 of file mxml-node.c.

References mxml\_new(), MXML\_OPAQUE, node, mxml\_value\_u::opaque, and mxml\_node\_s::value.

Referenced by main(), and mxml\_load\_data().

**8.69.4.26 mxml\_node\_t\* mxmlNewReal (mxml\_node\_t \* *parent*, double *real*)**

Definition at line 430 of file mxml-node.c.

References mxml\_new(), MXML\_REAL, node, mxml\_value\_u::real, and mxml\_node\_s::value.

Referenced by main(), and mxml\_load\_data().

**8.69.4.27 mxml\_node\_t\* mxmlNewText (mxml\_node\_t \* *parent*, int *whitespace*, const char \* *string*)**

Definition at line 462 of file mxml-node.c.

References mxml\_new(), MXML\_TEXT, node, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::value, and mxml\_text\_s::whitespace.

Referenced by agent\_xml\_compose\_\_create\_row\_nodes(), agent\_xml\_compose\_\_home(), agent\_xml\_compose\_\_name(), agent\_xml\_compose\_\_owner(), fipa\_envelope\_Compose\_\_acl\_representation(), fipa\_envelope\_Compose\_\_date(), fipa\_envelope\_Compose\_\_from(), fipa\_envelope\_Compose\_\_intended\_receiver(), fipa\_envelope\_Compose\_\_payload\_encoding(), fipa\_envelope\_Compose\_\_to(), main(), mxml\_load\_data(), and scan\_file().

#### 8.69.4.28 **mxmxml\_node\_t\*** mxmxmlNewTextf (mxmxml\_node\_t \* *parent*, int *whitespace*, const char \* *format*, ...)

Definition at line 506 of file mxmxml-node.c.

References mxmxml\_new(), mxmxml\_strdupf(), MXMXML\_TEXT, node, mxmxml\_text\_s::string, mxmxml\_value\_u::text, mxmxml\_node\_s::value, and mxmxml\_text\_s::whitespace.

#### 8.69.4.29 **void** mxmxmlRemove (mxmxml\_node\_t \* *node*)

Definition at line 553 of file mxmxml-node.c.

References mxmxml\_node\_s::child, mxmxml\_node\_s::last\_child, mxmxml\_node\_s::next, mxmxml\_node\_s::parent, and mxmxml\_node\_s::prev.

Referenced by mxmxmlAdd(), and mxmxmlDelete().

#### 8.69.4.30 **char\*** mxmxmlSaveAllocString (mxmxml\_node\_t \* *node*, const char \*(\*)(mxmxml\_node\_t \*, int) *cb*)

#### 8.69.4.31 **int** mxmxmlSaveFd (mxmxml\_node\_t \* *node*, int *fd*, const char \*(\*)(mxmxml\_node\_t \*, int) *cb*)

#### 8.69.4.32 **int** mxmxmlSaveFile (mxmxml\_node\_t \* *node*, FILE \* *fp*, const char \*(\*)(mxmxml\_node\_t \*, int) *cb*)

#### 8.69.4.33 **int** mxmxmlSaveString (mxmxml\_node\_t \* *node*, char \* *buffer*, int *bufsize*, const char \*(\*)(mxmxml\_node\_t \*, int) *cb*)

#### 8.69.4.34 **int** mxmxmlSetCustom (mxmxml\_node\_t \* *node*, void \* *data*, void(\*)(void \*) *destroy*)

Definition at line 46 of file mxmxml-set.c.

References mxmxml\_value\_u::custom, mxmxml\_custom\_s::data, mxmxml\_custom\_s::destroy, MXMXML\_CUSTOM, mxmxml\_node\_s::type, and mxmxml\_node\_s::value.

#### 8.69.4.35 **void** mxmxmlSetCustomHandlers (mxmxml\_custom\_load\_cb\_t *load*, mxmxml\_custom\_save\_cb\_t *save*)

Definition at line 456 of file mxmxml-file.c.

References mxmxml\_custom\_load\_cb, and mxmxml\_custom\_save\_cb.

#### 8.69.4.36 **int** mxmxmlSetElement (mxmxml\_node\_t \* *node*, const char \* *name*)

Definition at line 79 of file mxmxml-set.c.

References mxmxml\_value\_u::element, MXMXML\_ELEMENT, mxmxml\_value\_s::name, mxmxml\_node\_s::type, and mxmxml\_node\_s::value.

#### 8.69.4.37 **void** mxmxmlSetErrorCallback (void(\*)(const char \*) *cb*)

Definition at line 471 of file mxmxml-file.c.

References mxmxml\_error\_cb.

**8.69.4.38 int mxmlSetInteger (mxml\_node\_t \* *node*, int *integer*)**

Definition at line 109 of file mxml-set.c.

References mxml\_value\_u::integer, MXML\_INTEGER, mxml\_node\_s::type, and mxml\_node\_s::value.

**8.69.4.39 int mxmlSetOpaque (mxml\_node\_t \* *node*, const char \* *opaque*)**

Definition at line 136 of file mxml-set.c.

References MXML\_OPAQUE, mxml\_value\_u::opaque, mxml\_node\_s::type, and mxml\_node\_s::value.

**8.69.4.40 int mxmlSetReal (mxml\_node\_t \* *node*, double *real*)**

Definition at line 166 of file mxml-set.c.

References MXML\_REAL, mxml\_value\_u::real, mxml\_node\_s::type, and mxml\_node\_s::value.

**8.69.4.41 int mxmlSetText (mxml\_node\_t \* *node*, int *whitespace*, const char \* *string*)**

Definition at line 193 of file mxml-set.c.

References MXML\_TEXT, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, mxml\_node\_s::value, and mxml\_text\_s::whitespace.

**8.69.4.42 int mxmlSetTextf (mxml\_node\_t \* *node*, int *whitespace*, const char \* *format*, ...)**

Definition at line 225 of file mxml-set.c.

References mxml\_strdupf(), MXML\_TEXT, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, mxml\_node\_s::value, and mxml\_text\_s::whitespace.

**8.69.4.43 mxml\_node\_t\* mxmlWalkNext (mxml\_node\_t \* *node*, mxml\_node\_t \* *top*, int *descend*)**

Definition at line 133 of file mxml-search.c.

References mxml\_node\_s::child, mxml\_node\_s::next, and mxml\_node\_s::parent.

Referenced by mxmlFindElement(), and write\_element().

**8.69.4.44 mxml\_node\_t\* mxmlWalkPrev (mxml\_node\_t \* *node*, mxml\_node\_t \* *top*, int *descend*)**

Definition at line 169 of file mxml-search.c.

References mxml\_node\_s::last\_child, mxml\_node\_s::parent, and mxml\_node\_s::prev.

## 8.70 /home/dko/projects/mobilec/trunk/src/mxml-2.2.2/mxmldoc.c File Reference

```
#include "config.h"
#include "mxml.h"
```

### Defines

- #define [STATE\\_NONE](#) 0
- #define [STATE\\_PREPROCESSOR](#) 1
- #define [STATE\\_C\\_COMMENT](#) 2
- #define [STATE\\_CXX\\_COMMENT](#) 3
- #define [STATE\\_STRING](#) 4
- #define [STATE\\_CHARACTER](#) 5
- #define [STATE\\_IDENTIFIER](#) 6

### Functions

- static [mxml\\_node\\_t](#) \* [add\\_variable](#) ([mxml\\_node\\_t](#) \*parent, const char \*name, [mxml\\_node\\_t](#) \*type)
- static void [safe\\_strcpy](#) (char \*dst, const char \*src)
- static [int](#) [scan\\_file](#) (const char \*filename, FILE \*fp, [mxml\\_node\\_t](#) \*doc)
- static void [sort\\_node](#) ([mxml\\_node\\_t](#) \*tree, [mxml\\_node\\_t](#) \*func)
- static void [update\\_comment](#) ([mxml\\_node\\_t](#) \*parent, [mxml\\_node\\_t](#) \*comment)
- static void [write\\_documentation](#) ([mxml\\_node\\_t](#) \*doc)
- static void [write\\_element](#) ([mxml\\_node\\_t](#) \*doc, [mxml\\_node\\_t](#) \*element)
- static void [write\\_string](#) (const char \*s)
- static const char \* [ws\\_cb](#) ([mxml\\_node\\_t](#) \*node, [int](#) where)
- [int](#) [main](#) ([int](#) argc, char \*argv[ ])

#### 8.70.1 Define Documentation

##### 8.70.1.1 #define STATE\_C\_COMMENT 2

Definition at line 125 of file mxmldoc.c.

Referenced by [scan\\_file\(\)](#).

##### 8.70.1.2 #define STATE\_CHARACTER 5

Definition at line 128 of file mxmldoc.c.

Referenced by [scan\\_file\(\)](#).

##### 8.70.1.3 #define STATE\_CXX\_COMMENT 3

Definition at line 126 of file mxmldoc.c.

Referenced by [scan\\_file\(\)](#).

#### 8.70.1.4 **#define STATE\_IDENTIFIER 6**

Definition at line 129 of file mxmldoc.c.

Referenced by scan\_file().

#### 8.70.1.5 **#define STATE\_NONE 0**

Definition at line 123 of file mxmldoc.c.

Referenced by scan\_file().

#### 8.70.1.6 **#define STATE\_PREPROCESSOR 1**

Definition at line 124 of file mxmldoc.c.

Referenced by scan\_file().

#### 8.70.1.7 **#define STATE\_STRING 4**

Definition at line 127 of file mxmldoc.c.

Referenced by scan\_file().

### 8.70.2 **Function Documentation**

#### 8.70.2.1 **static mxml\_node\_t \* add\_variable (mxml\_node\_t \* *parent*, const char \* *name*, mxml\_node\_t \* *type*)** [static]

Definition at line 309 of file mxmldoc.c.

References mxml\_node\_s::child, mxml\_node\_s::last\_child, MXML\_ADD\_AFTER, MXML\_ADD\_TO\_PARENT, mxmlAdd(), mxmlDelete(), mxmlElementSetAttr(), mxmlNewElement(), mxml\_node\_s::next, node, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::value, and mxml\_text\_s::whitespace.

Referenced by scan\_file().

#### 8.70.2.2 **int main (int *argc*, char \* *argv*[])**

Definition at line 155 of file mxmldoc.c.

References MXML\_DESCEND, MXML\_NO\_CALLBACK, mxmlDelete(), mxmlElementSetAttr(), mxmlFindElement(), mxmlLoadFile(), mxmlNewElement(), mxmlSaveFile(), scan\_file(), write\_documentation(), and ws\_cb().

#### 8.70.2.3 **static void safe\_strcpy (char \* *dst*, const char \* *src*)** [static]

Definition at line 423 of file mxmldoc.c.

Referenced by update\_comment().

#### 8.70.2.4 `static int scan_file (const char *filename, FILE *fp, mxml_node_t *doc)` [static]

Definition at line 438 of file mxmldoc.c.

References `add_variable()`, `mxml_node_s::child`, `mxml_value_u::element`, `mxml_node_s::last_child`, `MXML_ADD_AFTER`, `MXML_ADD_BEFORE`, `MXML_ADD_TO_PARENT`, `MXML_DESCEND_FIRST`, `MXML_NO_PARENT`, `mxmlAdd()`, `mxmlDelete()`, `mxmlElementGetAttr()`, `mxmlElementSetAttr()`, `mxmlFindElement()`, `mxmlNewElement()`, `mxmlNewText()`, `mxml_value_s::name`, `mxml_node_s::next`, `node`, `sort_node()`, `STATE_C_COMMENT`, `STATE_CHARACTER`, `STATE_CXX_COMMENT`, `STATE_IDENTIFIER`, `STATE_NONE`, `STATE_PREPROCESSOR`, `STATE_STRING`, `mxml_text_s::string`, `mxml_value_u::text`, `update_comment()`, `mxml_node_s::value`, and `mxml_text_s::whitespace`.

Referenced by `main()`.

#### 8.70.2.5 `static void sort_node (mxml_node_t *tree, mxml_node_t *func)` [static]

Definition at line 1613 of file mxmldoc.c.

References `mxml_node_s::child`, `mxml_value_u::element`, `MXML_ADD_AFTER`, `MXML_ADD_BEFORE`, `MXML_ADD_TO_PARENT`, `MXML_DESCEND_FIRST`, `mxmlAdd()`, `mxmlDelete()`, `mxmlElementGetAttr()`, `mxmlElementSetAttr()`, `mxmlFindElement()`, `mxml_value_s::name`, `mxml_node_s::next`, `mxml_node_s::parent`, and `mxml_node_s::value`.

Referenced by `scan_file()`.

#### 8.70.2.6 `static void update_comment (mxml_node_t *parent, mxml_node_t *comment)` [static]

Definition at line 1701 of file mxmldoc.c.

References `mxml_value_u::element`, `mxmlElementSetAttr()`, `mxml_value_s::name`, `safe_strcpy()`, `mxml_text_s::string`, `mxml_value_u::text`, and `mxml_node_s::value`.

Referenced by `scan_file()`.

#### 8.70.2.7 `static void write_documentation (mxml_node_t *doc)` [static]

Definition at line 1801 of file mxmldoc.c.

References `mxml_node_s::child`, `MXML_DESCEND_FIRST`, `MXML_NO_DESCEND`, `MXML_VERSION`, `mxmlElementGetAttr()`, `mxmlFindElement()`, and `write_element()`.

Referenced by `main()`.

#### 8.70.2.8 `static void write_element (mxml_node_t *doc, mxml_node_t *element)` [static]

Definition at line 2611 of file mxmldoc.c.

References `mxml_node_s::child`, `MXML_DESCEND`, `MXML_NO_DESCEND`, `MXML_TEXT`, `mxmlFindElement()`, `mxmlWalkNext()`, `node`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::type`, `mxml_node_s::value`, `mxml_text_s::whitespace`, and `write_string()`.

Referenced by `write_documentation()`.



**8.70.2.9 static void write\_string (const char \* *s*)** [static]

Definition at line 2656 of file mxmldoc.c.

Referenced by write\_element().

**8.70.2.10 static const char \* ws\_cb (mxml\_node\_t \* *node*, int *where*)** [static]

Definition at line 2714 of file mxmldoc.c.

References mxml\_value\_u::element, MXML\_WS\_AFTER\_CLOSE, MXML\_WS\_AFTER\_OPEN, MXML\_WS\_BEFORE\_CLOSE, MXML\_WS\_BEFORE\_OPEN, mxml\_value\_s::name, mxml\_node\_s::parent, and mxml\_node\_s::value.

Referenced by main().

## 8.71 /home/dko/projects/mobilec/trunk/src/mxml-2.2.2/test/class.cxx File Reference

### Data Structures

- class `foo_c`

### Variables

- `foo_c f`
- `foo_c foo`
- `bar = b`

### 8.71.1 Variable Documentation

#### 8.71.1.1 `bar = b`

Definition at line 77 of file class.cxx.

#### 8.71.1.2 `foo_c f`

Referenced by `foo_float_function()`, `main()`, `md2_file()`, `md4_file()`, `md5_check()`, `md5_file()`, `sha1_check()`, `sha1_file()`, `sha2_check()`, `sha2_file()`, `sha4_file()`, `x509_read_crtfile()`, and `x509_read_keyfile()`.

#### 8.71.1.3 `foo_c foo`

## 8.72 /home/dko/projects/mobilec/trunk/src/mxml-2.2.2/test/enum.cxx File Reference

### Typedefs

- typedef enum [foo\\_enum\\_e](#) [foo\\_enum\\_t](#)

### Enumerations

- enum [foo\\_enum\\_e](#) { [FOO\\_ONE](#), [FOO\\_TWO](#), [FOO\\_RED](#), [FOO\\_BLUE](#) }

#### 8.72.1 Typedef Documentation

8.72.1.1 typedef enum [foo\\_enum\\_e](#) [foo\\_enum\\_t](#)

#### 8.72.2 Enumeration Type Documentation

8.72.2.1 enum [foo\\_enum\\_e](#)

Enumerator:

*FOO\_ONE*  
*FOO\_TWO*  
*FOO\_RED*  
*FOO\_BLUE*

Definition at line 1 of file enum.cxx.

## 8.73 /home/dko/projects/mobilec/trunk/src/mxml-2.2.2/test/function.cxx File Reference

### Functions

- void [foo\\_void\\_function](#) ([int](#) one, float \*two, const char \*three)
- float [foo\\_float\\_function](#) ([int](#) one, const char \*two)
- [int](#) [foo\\_default\\_string](#) ([int](#) one, const char \*two="2")
- [int](#) [foo\\_default\\_int](#) ([int](#) one, [int](#) two=2)

### 8.73.1 Function Documentation

#### 8.73.1.1 [int](#) [foo\\_default\\_int](#) ([int](#) *one*, [int](#) *two* = 2)

Definition at line 65 of file function.cxx.

#### 8.73.1.2 [int](#) [foo\\_default\\_string](#) ([int](#) *one*, const char \* *two* = "2")

Definition at line 45 of file function.cxx.

#### 8.73.1.3 [float](#) [foo\\_float\\_function](#) ([int](#) *one*, const char \* *two*)

Definition at line 26 of file function.cxx.

References [f](#).

#### 8.73.1.4 [void](#) [foo\\_void\\_function](#) ([int](#) *one*, float \* *two*, const char \* *three*)

Definition at line 6 of file function.cxx.

## 8.74 /home/dko/projects/mobilec/trunk/src/mxml-2.2.2/test/struct.cxx File Reference

### Data Structures

- struct [foo\\_s](#)

### Typedefs

- typedef struct [foo\\_s](#) [foo\\_t](#)

#### 8.74.1 Typedef Documentation

##### 8.74.1.1 typedef struct [foo\\_s](#) [foo\\_t](#)

## 8.75 /home/dko/projects/mobilec/trunk/src/mxml-2.2.2/testmxml.c File Reference

```
#include "config.h"
#include "mxml.h"
#include <unistd.h>
#include <fcntl.h>
```

### Defines

- `#define O_BINARY 0`

### Functions

- `mxml_type_t type_cb (mxml_node_t *node)`
- `const char * whitespace_cb (mxml_node_t *node, int where)`
- `int main (int argc, char *argv[ ])`

#### 8.75.1 Define Documentation

##### 8.75.1.1 `#define O_BINARY 0`

Definition at line 42 of file testmxml.c.

Referenced by `main()`.

#### 8.75.2 Function Documentation

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

Definition at line 59 of file testmxml.c.

References `mxml_node_s::child`, `mxml_value_u::element`, `f`, `mxml_value_u::integer`, `mxml_node_s::last_child`, `MXML_DESCEND`, `MXML_ELEMENT`, `MXML_INTEGER`, `MXML_INTEGER_CALLBACK`, `MXML_NO_CALLBACK`, `MXML_NO_DESCEND`, `MXML_NO_PARENT`, `MXML_OPAQUE`, `MXML_OPAQUE_CALLBACK`, `MXML_REAL`, `MXML_REAL_CALLBACK`, `MXML_TEXT`, `mxmlDelete()`, `mxmlFindElement()`, `mxmlIndexDelete()`, `mxmlIndexEnum()`, `mxmlIndexFind()`, `mxmlIndexNew()`, `mxmlIndexReset()`, `mxmlLoadFd()`, `mxmlLoadFile()`, `mxmlLoadString()`, `mxmlNewElement()`, `mxmlNewInteger()`, `mxmlNewOpaque()`, `mxmlNewReal()`, `mxmlNewText()`, `mxmlSaveFd()`, `mxmlSaveFile()`, `mxmlSaveString()`, `mxml_value_s::name`, `mxml_node_s::next`, `node`, `mxml_index_s::num_nodes`, `O_BINARY`, `mxml_value_u::opaque`, `mxml_value_u::real`, `mxml_text_s::string`, `mxml_value_u::text`, `mxml_node_s::type`, `type_cb()`, `mxml_node_s::value`, `mxml_text_s::whitespace`, and `whitespace_cb()`.

##### 8.75.2.2 `mxml_type_t type_cb (mxml_node_t * node)`

Definition at line 542 of file testmxml.c.

References mxml\_value\_u::element, MXML\_INTEGER, MXML\_OPAQUE, MXML\_REAL, MXML\_TEXT, mxmlElementGetAttr(), mxml\_value\_s::name, and mxml\_node\_s::value.

Referenced by main().

### 8.75.2.3 `const char* whitespace_cb (mxml_node_t * node, int where)`

Definition at line 249 of file xml\_helper.c.

References MXML\_WS\_AFTER\_CLOSE, and MXML\_WS\_BEFORE\_OPEN.

## 8.76 `/home/dko/projects/mobilec/trunk/src/security/asm.c` File Reference

```
#include "../include/mc_platform.h"
#include "../include/message.h"
#include "asm.h"
#include "asm_message_composer.h"
#include "config.h"
#include "mc_dh.h"
```



## 8.77 /home/dko/projects/mobilec/trunk/src/security/asm.h File Reference

```
#include "config.h"
#include "../include/ap_queue_template.h"
#include "../include/data_structures.h"
#include "../mc_list/list.h"
#include "asm_node.h"
```

## 8.78 `/home/dko/projects/mobilec/trunk/src/security/asm_message_composer.c` File Reference

```
#include "asm_message_composer.h"  
#include "config.h"  
#include "../include/mc_platform.h"
```

## **8.79 /home/dko/projects/mobilec/trunk/src/security/asm\_message\_composer.h File Reference**

```
#include <mxml.h>
#include "asm.h"
#include "config.h"
```

## 8.80 `/home/dko/projects/mobilec/trunk/src/security/asm_message_parser.c` File Reference

```
#include "asm_message_parser.h"  
#include "../include/xml_helper.h"  
#include "config.h"
```

## 8.81 /home/dko/projects/mobilec/trunk/src/security/asm\_message\_parser.h File Reference

```
#include "../include/mc_error.h"
#include "../include/xml_parser.h"
#include "asm_node.h"
#include "config.h"
```

## 8.82 `/home/dko/projects/mobilec/trunk/src/security/asm_node.c` File Reference

```
#include <netdb.h>
#include "config.h"
#include "asm.h"
#include "asm_message_parser.h"
#include "asm_node.h"
#include "xyssl-0.7/include/xyssl/havege.h"
#include "xyssl-0.7/include/xyssl/bignum.h"
```

## 8.83 /home/dko/projects/mobilec/trunk/src/security/asm\_node.h File Reference

```
#include <mxml.h>
#include <netinet/in.h>
#include "../include/macros.h"
#include "../include/message.h"
#include "xyssl-0.7/include/xyssl/dhm.h"
#include "xyssl-0.7/include/xyssl/rsa.h"
#include "xyssl-0.7/include/xyssl/aes.h"
#include "config.h"
```

## 8.84 `/home/dko/projects/mobilec/trunk/src/security/mc_dh.c` File Reference

```
#include <stdio.h>
#include "../include/mc_error.h"
#include "xyssl-0.7/include/xyssl/dhm.h"
#include "xyssl-0.7/include/xyssl/havege.h"
#include "mc_dh.h"
#include "asm_node.h"
#include "config.h"
```



## 8.85 /home/dko/projects/mobilec/trunk/src/security/mc\_dh.h File Reference

```
#include "xyssl-0.7/include/xyssl/rsa.h"  
#include "asm_node.h"  
#include "config.h"
```

## 8.86 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/aes.h File Reference

### Data Structures

- struct [aes\\_context](#)  
*AES context structure.*

### Functions

- void [aes\\_set\\_key](#) ([aes\\_context](#) \*ctx, unsigned char \*key, int keysize)  
*AES key schedule.*
- void [aes\\_encrypt](#) ([aes\\_context](#) \*ctx, unsigned char input[16], unsigned char output[16])  
*AES block encryption (ECB mode).*
- void [aes\\_decrypt](#) ([aes\\_context](#) \*ctx, unsigned char input[16], unsigned char output[16])  
*AES block decryption (ECB mode).*
- void [aes\\_cbc\\_encrypt](#) ([aes\\_context](#) \*ctx, unsigned char iv[16], unsigned char \*input, unsigned char \*output, int len)  
*AES-CBC buffer encryption.*
- void [aes\\_cbc\\_decrypt](#) ([aes\\_context](#) \*ctx, unsigned char iv[16], unsigned char \*input, unsigned char \*output, int len)  
*AES-CBC buffer decryption.*
- int [aes\\_self\\_test](#) (int verbose)  
*Checkup routine.*

### 8.86.1 Detailed Description

Definition in file [aes.h](#).

### 8.86.2 Function Documentation

#### 8.86.2.1 void [aes\\_cbc\\_decrypt](#) ([aes\\_context](#) \* ctx, unsigned char iv[16], unsigned char \* input, unsigned char \* output, int len)

AES-CBC buffer decryption.

#### Parameters:

- ctx* AES context
- iv* initialization vector (modified after use)
- input* buffer holding the ciphertext

*output* buffer holding the plaintext  
*len* length of the data to be decrypted

Definition at line 825 of file aes.c.

Referenced by ssl\_decrypt\_buf().

#### 8.86.2.2 void aes\_cbc\_encrypt (aes\_context \* ctx, unsigned char iv[16], unsigned char \* input, unsigned char \* output, int len)

AES-CBC buffer encryption.

##### Parameters:

*ctx* AES context  
*iv* initialization vector (modified after use)  
*input* buffer holding the plaintext  
*output* buffer holding the ciphertext  
*len* length of the data to be encrypted

Definition at line 800 of file aes.c.

Referenced by main(), and ssl\_encrypt\_buf().

#### 8.86.2.3 void aes\_decrypt (aes\_context \* ctx, unsigned char input[16], unsigned char output[16])

AES block decryption (ECB mode).

##### Parameters:

*ctx* AES context  
*input* ciphertext block  
*output* plaintext block

Definition at line 709 of file aes.c.

Referenced by main().

#### 8.86.2.4 void aes\_encrypt (aes\_context \* ctx, unsigned char input[16], unsigned char output[16])

AES block encryption (ECB mode).

##### Parameters:

*ctx* AES context  
*input* plaintext block  
*output* ciphertext block

AES block encryption (ECB mode)

Definition at line 618 of file aes.c.

Referenced by main().

**8.86.2.5** `int aes_self_test (int verbose)`

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 923 of file aes.c.

Referenced by main().

**8.86.2.6** `void aes_set_key (aes_context * ctx, unsigned char * key, int keysize)`

AES key schedule.

**Parameters:**

*ctx* AES context to be initialized

*key* the secret key

*keysize* must be 128, 192 or 256

Referenced by main(), and ssl\_derive\_keys().

## 8.87 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/arc4.h File Reference

### Data Structures

- struct [arc4\\_context](#)  
*ARC4 context structure.*

### Functions

- void [arc4\\_setup](#) ([arc4\\_context](#) \*ctx, unsigned char \*key, [int](#) keylen)  
*ARC4 key schedule.*
- void [arc4\\_crypt](#) ([arc4\\_context](#) \*ctx, unsigned char \*buf, [int](#) buflen)  
*ARC4 cipher function.*
- [int](#) [arc4\\_self\\_test](#) ([int](#) verbose)

### 8.87.1 Detailed Description

Definition in file [arc4.h](#).

### 8.87.2 Function Documentation

#### 8.87.2.1 void [arc4\\_crypt](#) ([arc4\\_context](#) \* ctx, unsigned char \* buf, int buflen)

ARC4 cipher function.

#### Parameters:

*ctx* ARC4 context  
*buf* buffer to be processed  
*buflen* amount of data in buf

Definition at line 63 of file arc4.c.

References [arc4\\_context::m](#), [arc4\\_context::x](#), and [arc4\\_context::y](#).

Referenced by [main\(\)](#), [ssl\\_decrypt\\_buf\(\)](#), and [ssl\\_encrypt\\_buf\(\)](#).

#### 8.87.2.2 int [arc4\\_self\\_test](#) (int verbose)

Definition at line 150 of file arc4.c.

Referenced by [main\(\)](#).

**8.87.2.3 void arc4\_setup (arc4\_context \* *ctx*, unsigned char \* *key*, int *keylen*)**

ARC4 key schedule.

**Parameters:**

*ctx* ARC4 context to be initialized

*key* the secret key

*keylen* length of the key

Definition at line 38 of file arc4.c.

References arc4\_context::m, arc4\_context::x, and arc4\_context::y.

Referenced by main(), and ssl\_derive\_keys().

## 8.88 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/base64.h File Reference

### Defines

- `#define ERR_BASE64_BUFFER_TOO_SMALL 0x0010`
- `#define ERR_BASE64_INVALID_CHARACTER 0x0012`

### Functions

- `int base64_encode` (unsigned char \*dst, int \*dlen, unsigned char \*src, int slen)  
*Encode a buffer into base64 format.*
- `int base64_decode` (unsigned char \*dst, int \*dlen, unsigned char \*src, int slen)  
*Decode a base64-formatted buffer.*
- `int base64_self_test` (int verbose)  
*Checkup routine.*

### 8.88.1 Detailed Description

Definition in file [base64.h](#).

### 8.88.2 Define Documentation

#### 8.88.2.1 `#define ERR_BASE64_BUFFER_TOO_SMALL 0x0010`

Definition at line 14 of file [base64.h](#).

Referenced by [base64\\_decode\(\)](#), and [base64\\_encode\(\)](#).

#### 8.88.2.2 `#define ERR_BASE64_INVALID_CHARACTER 0x0012`

Definition at line 15 of file [base64.h](#).

Referenced by [base64\\_decode\(\)](#), [x509\\_add\\_certs\(\)](#), and [x509\\_parse\\_key\(\)](#).

### 8.88.3 Function Documentation

#### 8.88.3.1 `int base64_decode` (unsigned char \* dst, int \* dlen, unsigned char \* src, int slen)

Decode a base64-formatted buffer.

#### Parameters:

- dst* destination buffer
- dlen* size of the buffer (updated after call)
- src* source buffer

*slen* amount of data to be decoded

**Returns:**

0 if successful, ERR\_BASE64\_BUFFER\_TOO\_SMALL, or ERR\_BASE64\_INVALID\_DATA if an invalid char is found. \*dlen is always updated to reflect to amount of data that was written (or would have been written)

**Note:**

Call this function with \*dlen = 0 to obtain the required buffer size in \*dlen

Definition at line 121 of file base64.c.

References base64\_dec\_map, ERR\_BASE64\_BUFFER\_TOO\_SMALL, and ERR\_BASE64\_INVALID\_CHARACTER.

Referenced by x509\_add\_certs(), and x509\_parse\_key().

**8.88.3.2 int base64\_encode (unsigned char \*dst, int \*dlen, unsigned char \*src, int slen)**

Encode a buffer into base64 format.

**Parameters:**

*dst* destination buffer

*dlen* size of the buffer (updated after call)

*src* source buffer

*slen* amount of data to be encoded

**Returns:**

0 if successful, or ERR\_BASE64\_BUFFER\_TOO\_SMALL. \*dlen is always updated to reflect to amount of data that was written (or would have been written)

**Note:**

Call this function with \*dlen = 0 to obtain the required buffer size in \*dlen

Definition at line 61 of file base64.c.

References base64\_enc\_map, and ERR\_BASE64\_BUFFER\_TOO\_SMALL.

**8.88.3.3 int base64\_self\_test (int verbose)**

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 248 of file base64.c.

Referenced by main().



## 8.89 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/bignum.h File Reference

```
#include <stdio.h>
#include "ssl_conf.h"
```

### Data Structures

- struct [mpi](#)  
*MPI structure.*

### Defines

- #define [ERR\\_MPI\\_FILE\\_IO\\_ERROR](#) 0x0002
- #define [ERR\\_MPI\\_INVALID\\_CHARACTER](#) 0x0004
- #define [ERR\\_MPI\\_INVALID\\_PARAMETER](#) 0x0006
- #define [ERR\\_MPI\\_BUFFER\\_TOO\\_SMALL](#) 0x0008
- #define [ERR\\_MPI\\_NEGATIVE\\_VALUE](#) 0x000A
- #define [ERR\\_MPI\\_DIVISION\\_BY\\_ZERO](#) 0x000C
- #define [ERR\\_MPI\\_NOT\\_ACCEPTABLE](#) 0x000E
- #define [CHK](#)(fc) if( ( ret = fc ) != 0 ) goto cleanup
- #define [ciL](#) ([int](#)) sizeof([t\\_int](#))
- #define [biL](#) ([ciL](#) << 3)
- #define [biH](#) ([ciL](#) << 2)

### Typedefs

- typedef unsigned long [t\\_int](#)
- typedef unsigned long long [t\\_dbl](#)

### Functions

- void [mpi\\_init](#) ([mpi](#) \*X,...)  
*Initialize one or more [mpi](#).*
- void [mpi\\_free](#) ([mpi](#) \*X,...)  
*Unallocate one or more [mpi](#).*
- int [mpi\\_grow](#) ([mpi](#) \*X, int nblimbs)  
*Enlarge X to the specified number of limbs.*
- int [mpi\\_copy](#) ([mpi](#) \*X, [mpi](#) \*Y)  
*Copy the contents of Y into X.*
- void [mpi\\_swap](#) ([mpi](#) \*X, [mpi](#) \*Y)  
*Swap the contents of X and Y.*

- `int mpi_lset (mpi *X, int z)`  
*Set value from integer.*
- `int mpi_read_string (mpi *X, int radix, char *s)`  
*Import X from an ASCII string.*
- `int mpi_write_string (mpi *X, int radix, char *s, int *slen)`  
*Export X into an ASCII string.*
- `int mpi_read_file (mpi *X, int radix, FILE *fin)`  
*Read X from an opened file.*
- `int mpi_write_file (char *p, mpi *X, int radix, FILE *fout)`  
*Write X into an opened file, or stdout.*
- `int mpi_read_binary (mpi *X, unsigned char *buf, int buflen)`  
*Import X from unsigned binary data, big endian.*
- `int mpi_write_binary (mpi *X, unsigned char *buf, int *buflen)`  
*Export X into unsigned binary data, big endian.*
- `int mpi_msb (mpi *X)`  
*Return the total size in bits, without leading 0s.*
- `int mpi_lsb (mpi *X)`  
*Return the number of least significant bits.*
- `int mpi_shift_l (mpi *X, int count)`  
*Left-shift:  $X \ll = \text{count}$ .*
- `int mpi_shift_r (mpi *X, int count)`  
*Right-shift:  $X \gg = \text{count}$ .*
- `int mpi_cmp_abs (mpi *X, mpi *Y)`  
*Compare unsigned values.*
- `int mpi_cmp_mpi (mpi *X, mpi *Y)`  
*Compare signed values.*
- `int mpi_cmp_int (mpi *X, int z)`  
*Compare signed values.*
- `int mpi_add_abs (mpi *X, mpi *A, mpi *B)`  
*Unsigned addition:  $X = |A| + |B|$ .*
- `int mpi_sub_abs (mpi *X, mpi *A, mpi *B)`  
*Unsigned subtraction:  $X = |A| - |B|$ .*
- `int mpi_add_mpi (mpi *X, mpi *A, mpi *B)`

*Signed addition:  $X = A + B$ .*

- `int mpi_sub_mpi (mpi *X, mpi *A, mpi *B)`

*Signed subtraction:  $X = A - B$ .*

- `int mpi_add_int (mpi *X, mpi *A, int b)`

*Signed addition:  $X = A + b$ .*

- `int mpi_sub_int (mpi *X, mpi *A, int b)`

*Signed subtraction:  $X = A - b$ .*

- `int mpi_mul_mpi (mpi *X, mpi *A, mpi *B)`

*Baseline multiplication:  $X = A * B$ .*

- `int mpi_mul_int (mpi *X, mpi *A, t_int b)`

*Baseline multiplication:  $X = A * b$ .*

- `int mpi_div_mpi (mpi *Q, mpi *R, mpi *A, mpi *B)`

*Division by mpi:  $A = Q * B + R$ .*

- `int mpi_div_int (mpi *Q, mpi *R, mpi *A, int b)`

*Division by int:  $A = Q * b + R$ .*

- `int mpi_mod_mpi (mpi *R, mpi *A, mpi *B)`

*Modulo:  $R = A \bmod B$ .*

- `int mpi_mod_int (t_int *r, mpi *A, int b)`

*Modulo:  $r = A \bmod b$ .*

- `int mpi_exp_mod (mpi *X, mpi *A, mpi *E, mpi *N, mpi *_RR)`

*Sliding-window exponentiation:  $X = A^E \bmod N$ .*

- `int mpi_gcd (mpi *G, mpi *A, mpi *B)`

*Greatest common divisor:  $G = \gcd(A, B)$ .*

- `int mpi_inv_mod (mpi *X, mpi *A, mpi *N)`

*Modular inverse:  $X = A^{-1} \bmod N$ .*

- `int mpi_is_prime (mpi *X)`

*Miller-Rabin primality test.*

- `int mpi_gen_prime (mpi *X, int nbits, int dh_flag, int(*rng_f)(void *), void *rng_d)`

*Prime number generation.*

- `int mpi_self_test (int verbose)`

*Checkup routine.*

## 8.89.1 Detailed Description

Definition in file `bignum.h`.

## 8.89.2 Define Documentation

### 8.89.2.1 #define biH (ciL << 2)

Definition at line 51 of file bignum.h.

Referenced by mpi\_div\_mpi(), and mpi\_mod\_int().

### 8.89.2.2 #define biL (ciL << 3)

Definition at line 50 of file bignum.h.

Referenced by mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_lsb(), mpi\_montg\_init(), mpi\_msb(), mpi\_shift\_l(), and mpi\_shift\_r().

### 8.89.2.3 #define CHK(fc) if( ( ret = fc ) != 0 ) goto cleanup

Definition at line 26 of file bignum.h.

Referenced by dhm\_calc\_secret(), dhm\_make\_params(), dhm\_make\_public(), mpi\_add\_abs(), mpi\_add\_mpi(), mpi\_copy(), mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_gcd(), mpi\_gen\_prime(), mpi\_inv\_mod(), mpi\_is\_prime(), mpi\_lset(), mpi\_mod\_mpi(), mpi\_mul\_mpi(), mpi\_read\_binary(), mpi\_read\_string(), mpi\_shift\_l(), mpi\_sub\_abs(), mpi\_sub\_mpi(), mpi\_write\_file(), mpi\_write\_hlp(), mpi\_write\_string(), rsa\_check\_privkey(), rsa\_gen\_key(), rsa\_private(), rsa\_public(), rsa\_read\_private(), rsa\_read\_public(), rsa\_write\_private(), and rsa\_write\_public().

### 8.89.2.4 #define ciL (int) sizeof(t\_int)

Definition at line 49 of file bignum.h.

Referenced by mpi\_copy(), mpi\_free(), mpi\_grow(), mpi\_lset(), mpi\_montmul(), mpi\_read\_binary(), mpi\_read\_string(), mpi\_write\_binary(), and mpi\_write\_string().

### 8.89.2.5 #define ERR\_MPI\_BUFFER\_TOO\_SMALL 0x0008

Definition at line 21 of file bignum.h.

Referenced by mpi\_write\_binary(), and mpi\_write\_string().

### 8.89.2.6 #define ERR\_MPI\_DIVISION\_BY\_ZERO 0x000C

Definition at line 23 of file bignum.h.

Referenced by mpi\_div\_mpi(), and mpi\_mod\_int().

### 8.89.2.7 #define ERR\_MPI\_FILE\_IO\_ERROR 0x0002

Definition at line 18 of file bignum.h.

Referenced by mpi\_read\_file(), and mpi\_write\_file().

#### **8.89.2.8 #define ERR\_MPI\_INVALID\_CHARACTER 0x0004**

Definition at line 19 of file bignum.h.

Referenced by mpi\_get\_digit().

#### **8.89.2.9 #define ERR\_MPI\_INVALID\_PARAMETER 0x0006**

Definition at line 20 of file bignum.h.

Referenced by mpi\_exp\_mod(), mpi\_gen\_prime(), mpi\_inv\_mod(), mpi\_read\_string(), mpi\_write\_hlp(), and mpi\_write\_string().

#### **8.89.2.10 #define ERR\_MPI\_NEGATIVE\_VALUE 0x000A**

Definition at line 22 of file bignum.h.

Referenced by mpi\_sub\_abs().

#### **8.89.2.11 #define ERR\_MPI\_NOT\_ACCEPTABLE 0x000E**

Definition at line 24 of file bignum.h.

Referenced by mpi\_gen\_prime(), mpi\_inv\_mod(), and mpi\_is\_prime().

### **8.89.3 Typedef Documentation**

#### **8.89.3.1 typedef unsigned long long t\_dbl**

Definition at line 44 of file bignum.h.

#### **8.89.3.2 typedef unsigned long t\_int**

Definition at line 35 of file bignum.h.

### **8.89.4 Function Documentation**

#### **8.89.4.1 int mpi\_add\_abs (mpi \* X, mpi \* A, mpi \* B)**

Unsigned addition:  $X = |A| + |B|$ .

##### **Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 644 of file bignum.c.

References CHK, mpi\_copy(), mpi\_grow(), mpi::n, and mpi::p.

Referenced by mpi\_add\_mpi(), and mpi\_sub\_mpi().

**8.89.4.2 int mpi\_add\_int (mpi \* X, mpi \* A, int b)**

Signed addition:  $X = A + b$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 809 of file bignum.c.

References mpi\_add\_mpi(), mpi::n, mpi::p, and mpi::s.

Referenced by mpi\_gen\_prime(), and mpi\_read\_string().

**8.89.4.3 int mpi\_add\_mpi (mpi \* X, mpi \* A, mpi \* B)**

Signed addition:  $X = A + B$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 747 of file bignum.c.

References CHK, mpi\_add\_abs(), mpi\_cmp\_abs(), mpi\_sub\_abs(), and mpi::s.

Referenced by mpi\_add\_int(), mpi\_div\_mpi(), mpi\_inv\_mod(), mpi\_mod\_mpi(), and rsa\_private().

**8.89.4.4 int mpi\_cmp\_abs (mpi \* X, mpi \* Y)**

Compare unsigned values.

**Returns:**

1 if  $|X|$  is greater than  $|Y|$ , -1 if  $|X|$  is lesser than  $|Y|$  or 0 if  $|X|$  is equal to  $|Y|$

Definition at line 565 of file bignum.c.

References mpi::n, and mpi::p.

Referenced by mpi\_add\_mpi(), mpi\_div\_mpi(), mpi\_montmul(), mpi\_sub\_abs(), and mpi\_sub\_mpi().

**8.89.4.5 int mpi\_cmp\_int (mpi \* X, int z)**

Compare signed values.

**Returns:**

1 if X is greater than z, -1 if X is lesser than z or 0 if X is equal to z

Definition at line 628 of file bignum.c.

References mpi\_cmp\_mpi(), mpi::n, mpi::p, and mpi::s.

Referenced by mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_gcd(), mpi\_inv\_mod(), mpi\_is\_prime(), mpi\_mod\_mpi(), mpi\_write\_hlp(), rsa\_check\_privkey(), and rsa\_gen\_key().

#### 8.89.4.6 int mpi\_cmp\_mpi (mpi \* X, mpi \* Y)

Compare signed values.

##### Returns:

1 if X is greater than Y, -1 if X is lesser than Y or 0 if X is equal to Y

Definition at line 595 of file bignum.c.

References mpi::n, mpi::p, and mpi::s.

Referenced by dhm\_make\_params(), dhm\_make\_public(), mpi\_cmp\_int(), mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_gcd(), mpi\_inv\_mod(), mpi\_is\_prime(), mpi\_mod\_mpi(), rsa\_check\_privkey(), rsa\_gen\_key(), rsa\_private(), and rsa\_public().

#### 8.89.4.7 int mpi\_copy (mpi \* X, mpi \* Y)

Copy the contents of Y into X.

##### Returns:

0 if successful, 1 if memory allocation failed

Definition at line 118 of file bignum.c.

References CHK, ciL, mpi\_grow(), mpi::n, mpi::p, and mpi::s.

Referenced by mpi\_add\_abs(), mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_gcd(), mpi\_inv\_mod(), mpi\_is\_prime(), mpi\_mul\_mpi(), mpi\_sub\_abs(), and mpi\_write\_string().

#### 8.89.4.8 int mpi\_div\_int (mpi \* Q, mpi \* R, mpi \* A, int b)

Division by int:  $A = Q * b + R$ .

##### Returns:

0 if successful, 1 if memory allocation failed, ERR\_MPI\_DIVISION\_BY\_ZERO if  $b == 0$

##### Note:

Either Q or R can be NULL.

Definition at line 1120 of file bignum.c.

References mpi\_div\_mpi(), mpi::n, mpi::p, and mpi::s.

Referenced by main(), and mpi\_write\_hlp().

#### 8.89.4.9 int mpi\_div\_mpi (mpi \* Q, mpi \* R, mpi \* A, mpi \* B)

Division by mpi:  $A = Q * B + R$ .

##### Returns:

0 if successful, 1 if memory allocation failed, ERR\_MPI\_DIVISION\_BY\_ZERO if  $B == 0$

**Note:**

Either Q or R can be NULL.

Definition at line 956 of file bignum.c.

References biH, biL, CHK, ERR\_MPI\_DIVISION\_BY\_ZERO, mpi\_add\_mpi(), mpi\_cmp\_abs(), mpi\_cmp\_int(), mpi\_cmp\_mpi(), mpi\_copy(), mpi\_free(), mpi\_grow(), mpi\_init(), mpi\_lset(), mpi\_msb(), mpi\_mul\_int(), mpi\_shift\_l(), mpi\_shift\_r(), mpi\_sub\_mpi(), mpi::n, mpi::p, and mpi::s.

Referenced by mpi\_div\_int(), and mpi\_mod\_mpi().

**8.89.4.10 int mpi\_exp\_mod (mpi \* X, mpi \* A, mpi \* E, mpi \* N, mpi \* \_RR)**

Sliding-window exponentiation:  $X = A^E \bmod N$ .

**Returns:**

0 if successful, 1 if memory allocation failed, ERR\_MPI\_INVALID\_PARAMETER if N is negative or even

**Note:**

\_RR is used to avoid re-computing  $R \cdot R \bmod N$  across multiple calls, which speeds up things a bit. It can be set to NULL if the extra performance is unneeded.

Definition at line 1266 of file bignum.c.

References biL, CHK, ERR\_MPI\_INVALID\_PARAMETER, mpi\_cmp\_int(), mpi\_cmp\_mpi(), mpi\_copy(), mpi\_free(), mpi\_grow(), mpi\_init(), mpi\_lset(), mpi\_mod\_mpi(), mpi\_montg\_init(), mpi\_montmul(), mpi\_montred(), mpi\_msb(), mpi\_shift\_l(), mpi::n, and mpi::p.

Referenced by dhm\_calc\_secret(), dhm\_make\_params(), dhm\_make\_public(), main(), mpi\_is\_prime(), rsa\_private(), and rsa\_public().

**8.89.4.11 void mpi\_free (mpi \* X, ...)**

Unallocate one or more [mpi](#).

Definition at line 69 of file bignum.c.

References ciL, mpi::n, and mpi::p.

Referenced by dhm\_free(), main(), mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_gcd(), mpi\_gen\_prime(), mpi\_inv\_mod(), mpi\_is\_prime(), mpi\_mul\_mpi(), mpi\_read\_string(), mpi\_sub\_abs(), mpi\_write\_string(), rsa\_check\_privkey(), rsa\_free(), rsa\_gen\_key(), rsa\_private(), and rsa\_public().

**8.89.4.12 int mpi\_gcd (mpi \* G, mpi \* A, mpi \* B)**

Greatest common divisor:  $G = \gcd(A, B)$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 1443 of file bignum.c.



References CHK, mpi\_cmp\_int(), mpi\_cmp\_mpi(), mpi\_copy(), mpi\_free(), mpi\_init(), mpi\_lsb(), mpi\_lset(), mpi\_mul\_mpi(), mpi\_shift\_l(), mpi\_shift\_r(), mpi\_sub\_abs(), mpi::p, and mpi::s.

Referenced by mpi\_inv\_mod(), rsa\_check\_privkey(), and rsa\_gen\_key().

#### 8.89.4.13 int mpi\_gen\_prime (mpi \* X, int nbits, int dh\_flag, int(\*) (void \*) rng\_f, void \* rng\_d)

Prime number generation.

##### Parameters:

*X* destination [mpi](#)  
*nbits* required size of X in bits  
*dh\_flag* if 1, then (X-1)/2 will be prime too  
*rng\_f* points to the RNG function  
*rng\_d* points to the RNG data

##### Returns:

0 if successful (probably prime), 1 if memory allocation failed, ERR\_MPI\_INVALID\_PARAMETER if nbits is < 3

Definition at line 1713 of file bignum.c.

References BITS\_TO\_LIMBS, CHK, ERR\_MPI\_INVALID\_PARAMETER, ERR\_MPI\_NOT\_ACCEPTABLE, mpi\_add\_int(), mpi\_free(), mpi\_grow(), mpi\_init(), mpi\_is\_prime(), mpi\_lset(), mpi\_msb(), mpi\_shift\_l(), mpi\_shift\_r(), mpi\_sub\_int(), and mpi::p.

Referenced by main(), and rsa\_gen\_key().

#### 8.89.4.14 int mpi\_grow (mpi \* X, int nlimbs)

Enlarge X to the specified number of limbs.

##### Returns:

0 if successful, 1 if memory allocation failed

Definition at line 94 of file bignum.c.

References ciL, mpi::n, mpi::p, and mpi::s.

Referenced by dhm\_make\_params(), dhm\_make\_public(), mpi\_add\_abs(), mpi\_copy(), mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_gen\_prime(), mpi\_is\_prime(), mpi\_lset(), mpi\_mul\_mpi(), mpi\_read\_binary(), mpi\_read\_string(), and mpi\_shift\_l().

#### 8.89.4.15 void mpi\_init (mpi \* X, ...)

Initialize one or more [mpi](#).

Definition at line 51 of file bignum.c.

Referenced by main(), mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_gcd(), mpi\_gen\_prime(), mpi\_inv\_mod(), mpi\_is\_prime(), mpi\_mul\_mpi(), mpi\_read\_string(), mpi\_sub\_abs(), mpi\_write\_string(), rsa\_check\_privkey(), rsa\_gen\_key(), rsa\_private(), and rsa\_public().

**8.89.4.16 int mpi\_inv\_mod (mpi \* X, mpi \* A, mpi \* N)**

Modular inverse:  $X = A^{-1} \bmod N$ .

**Returns:**

0 if successful, 1 if memory allocation failed, ERR\_MPI\_INVALID\_PARAMETER if N is negative or nil ERR\_MPI\_NOT\_ACCEPTABLE if A has no inverse mod N

Definition at line 1491 of file bignum.c.

References CHK, ERR\_MPI\_INVALID\_PARAMETER, ERR\_MPI\_NOT\_ACCEPTABLE, mpi\_add\_mpi(), mpi\_cmp\_int(), mpi\_cmp\_mpi(), mpi\_copy(), mpi\_free(), mpi\_gcd(), mpi\_init(), mpi\_lset(), mpi\_mod\_mpi(), mpi\_shift\_r(), mpi\_sub\_mpi(), and mpi::p.

Referenced by main(), and rsa\_gen\_key().

**8.89.4.17 int mpi\_is\_prime (mpi \* X)**

Miller-Rabin primality test.

**Returns:**

0 if successful (probably prime), 1 if memory allocation failed, ERR\_MPI\_NOT\_ACCEPTABLE if X is not prime

Definition at line 1617 of file bignum.c.

References CHK, ERR\_MPI\_NOT\_ACCEPTABLE, mpi\_cmp\_int(), mpi\_cmp\_mpi(), mpi\_copy(), mpi\_exp\_mod(), mpi\_free(), mpi\_grow(), mpi\_init(), mpi\_lsb(), mpi\_mod\_int(), mpi\_mod\_mpi(), mpi\_msb(), mpi\_mul\_mpi(), mpi\_shift\_r(), mpi\_sub\_int(), mpi::n, mpi::p, R, mpi::s, and small\_prime.

Referenced by main(), and mpi\_gen\_prime().

**8.89.4.18 int mpi\_lsb (mpi \* X)**

Return the number of least significant bits.

Definition at line 461 of file bignum.c.

References biL, int, mpi::n, and mpi::p.

Referenced by mpi\_gcd(), and mpi\_is\_prime().

**8.89.4.19 int mpi\_lset (mpi \* X, int z)**

Set value from integer.

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 157 of file bignum.c.

References CHK, ciL, mpi\_grow(), mpi::n, mpi::p, and mpi::s.

Referenced by mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_gcd(), mpi\_gen\_prime(), mpi\_inv\_mod(), mpi\_mul\_mpi(), mpi\_read\_binary(), mpi\_read\_string(), and rsa\_gen\_key().

#### 8.89.4.20 int mpi\_mod\_int (t\_int \* r, mpi \* A, int b)

Modulo:  $r = A \bmod b$ .

##### Returns:

0 if successful, 1 if memory allocation failed, ERR\_MPI\_DIVISION\_BY\_ZERO if  $b == 0$ ,

Definition at line 1156 of file bignum.c.

References biH, ERR\_MPI\_DIVISION\_BY\_ZERO, mpi::n, and mpi::p.

Referenced by mpi\_is\_prime(), and mpi\_write\_hlp().

#### 8.89.4.21 int mpi\_mod\_mpi (mpi \* R, mpi \* A, mpi \* B)

Modulo:  $R = A \bmod B$ .

##### Returns:

0 if successful, 1 if memory allocation failed, ERR\_MPI\_DIVISION\_BY\_ZERO if  $B == 0$

Definition at line 1136 of file bignum.c.

References CHK, mpi\_add\_mpi(), mpi\_cmp\_int(), mpi\_cmp\_mpi(), mpi\_div\_mpi(), and mpi\_sub\_mpi().

Referenced by mpi\_exp\_mod(), mpi\_inv\_mod(), mpi\_is\_prime(), rsa\_gen\_key(), and rsa\_private().

#### 8.89.4.22 int mpi\_msb (mpi \* X)

Return the total size in bits, without leading 0s.

Definition at line 443 of file bignum.c.

References biL, mpi::n, and mpi::p.

Referenced by dhm\_calc\_secret(), dhm\_make\_params(), dhm\_read\_params(), mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_gen\_prime(), mpi\_is\_prime(), mpi\_shift\_l(), mpi\_write\_binary(), mpi\_write\_string(), rsa\_check\_pubkey(), rsa\_gen\_key(), rsa\_read\_private(), rsa\_read\_public(), x509\_add\_certs(), and x509\_parse\_key().

#### 8.89.4.23 int mpi\_mul\_int (mpi \* X, mpi \* A, t\_int b)

Baseline multiplication:  $X = A * b$ .

##### Returns:

0 if successful, 1 if memory allocation failed

Definition at line 940 of file bignum.c.

References mpi\_mul\_mpi(), mpi::n, mpi::p, and mpi::s.

Referenced by mpi\_div\_mpi(), and mpi\_read\_string().

**8.89.4.24 int mpi\_mul\_mpi (mpi \* X, mpi \* A, mpi \* B)**

Baseline multiplication:  $X = A * B$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 905 of file bignum.c.

References CHK, mpi\_copy(), mpi\_free(), mpi\_grow(), mpi\_init(), mpi\_lset(), mpi\_mul\_hlp(), mpi::n, mpi::p, and mpi::s.

Referenced by main(), mpi\_gcd(), mpi\_is\_prime(), mpi\_mul\_int(), rsa\_check\_privkey(), rsa\_gen\_key(), and rsa\_private().

**8.89.4.25 int mpi\_read\_binary (mpi \* X, unsigned char \* buf, int buflen)**

Import X from unsigned binary data, big endian.

**Parameters:**

*X* destination [mpi](#)

*buf* input buffer

*buflen* input buffer size

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 398 of file bignum.c.

References CHARS\_TO\_LIMBS, CHK, ciL, mpi\_grow(), mpi\_lset(), and mpi::p.

Referenced by asn1\_get\_mpi(), dhm\_read\_bignum(), dhm\_read\_public(), rsa\_private(), and rsa\_public().

**8.89.4.26 int mpi\_read\_file (mpi \* X, int radix, FILE \* fin)**

Read X from an opened file.

**Parameters:**

*X* destination [mpi](#)

*radix* input numeric base

*fin* input file handle

**Returns:**

0 if successful, or an ERR\_MPI\_XXX error code

Definition at line 335 of file bignum.c.

References ERR\_MPI\_FILE\_IO\_ERROR, mpi\_get\_digit(), and mpi\_read\_string().

Referenced by main(), rsa\_read\_private(), and rsa\_read\_public().

#### **8.89.4.27 int mpi\_read\_string (mpi \* X, int radix, char \* s)**

Import X from an ASCII string.

##### **Parameters:**

*X* destination [mpi](#)  
*radix* input numeric base  
*s* null-terminated string buffer

##### **Returns:**

0 if successful, or an ERR\_MPI\_XXX error code

Definition at line 191 of file bignum.c.

References [BITS\\_TO\\_LIMBS](#), [CHK](#), [ciL](#), [ERR\\_MPI\\_INVALID\\_PARAMETER](#), [int](#), [mpi\\_add\\_int\(\)](#), [mpi\\_free\(\)](#), [mpi\\_get\\_digit\(\)](#), [mpi\\_grow\(\)](#), [mpi\\_init\(\)](#), [mpi\\_lset\(\)](#), [mpi\\_mul\\_int\(\)](#), [mpi::p](#), and [mpi::s](#).

Referenced by [main\(\)](#), and [mpi\\_read\\_file\(\)](#).

#### **8.89.4.28 int mpi\_self\_test (int verbose)**

Checkup routine.

##### **Returns:**

0 if successful, or 1 if the test failed

Definition at line 1915 of file bignum.c.

Referenced by [main\(\)](#).

#### **8.89.4.29 int mpi\_shift\_l (mpi \* X, int count)**

Left-shift:  $X \ll= \text{count}$ .

##### **Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 476 of file bignum.c.

References [biL](#), [BITS\\_TO\\_LIMBS](#), [CHK](#), [mpi\\_grow\(\)](#), [mpi\\_msb\(\)](#), [mpi::n](#), and [mpi::p](#).

Referenced by [mpi\\_div\\_mpi\(\)](#), [mpi\\_exp\\_mod\(\)](#), [mpi\\_gcd\(\)](#), and [mpi\\_gen\\_prime\(\)](#).

#### **8.89.4.30 int mpi\_shift\_r (mpi \* X, int count)**

Right-shift:  $X \gg= \text{count}$ .

##### **Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 525 of file bignum.c.

References `biL`, `mpi::n`, and `mpi::p`.

Referenced by `dhm_make_params()`, `dhm_make_public()`, `mpi_div_mpi()`, `mpi_gcd()`, `mpi_gen_prime()`, `mpi_inv_mod()`, and `mpi_is_prime()`.

#### 8.89.4.31 `int mpi_sub_abs (mpi * X, mpi * A, mpi * B)`

Unsigned subtraction:  $X = |A| - |B|$ .

##### Returns:

0 if successful, `ERR_MPI_NEGATIVE_VALUE` if B is greater than A

Definition at line 711 of file bignum.c.

References `CHK`, `ERR_MPI_NEGATIVE_VALUE`, `mpi_cmp_abs()`, `mpi_copy()`, `mpi_free()`, `mpi_init()`, `mpi_sub_hlp()`, `mpi::n`, and `mpi::p`.

Referenced by `mpi_add_mpi()`, `mpi_gcd()`, and `mpi_sub_mpi()`.

#### 8.89.4.32 `int mpi_sub_int (mpi * X, mpi * A, int b)`

Signed subtraction:  $X = A - b$ .

##### Returns:

0 if successful, 1 if memory allocation failed

Definition at line 825 of file bignum.c.

References `mpi_sub_mpi()`, `mpi::n`, `mpi::p`, and `mpi::s`.

Referenced by `main()`, `mpi_gen_prime()`, `mpi_is_prime()`, `rsa_check_privkey()`, and `rsa_gen_key()`.

#### 8.89.4.33 `int mpi_sub_mpi (mpi * X, mpi * A, mpi * B)`

Signed subtraction:  $X = A - B$ .

##### Returns:

0 if successful, 1 if memory allocation failed

Definition at line 778 of file bignum.c.

References `CHK`, `mpi_add_abs()`, `mpi_cmp_abs()`, `mpi_sub_abs()`, and `mpi::s`.

Referenced by `mpi_div_mpi()`, `mpi_inv_mod()`, `mpi_mod_mpi()`, `mpi_sub_int()`, and `rsa_private()`.

#### 8.89.4.34 `void mpi_swap (mpi * X, mpi * Y)`

Swap the contents of X and Y.

Definition at line 145 of file bignum.c.

Referenced by `rsa_gen_key()`.

#### 8.89.4.35 `int mpi_write_binary (mpi * X, unsigned char * buf, int * buflen)`

Export *X* into unsigned binary data, big endian.

##### Parameters:

*X* source [mpi](#)  
*buf* output buffer  
*buflen* output buffer size

##### Returns:

0 if successful, ERR\_MPI\_BUFFER\_TOO\_SMALL if *buf* isn't large enough

##### Note:

Call this function with *\*buflen* = 0 to obtain the minimum required buffer size in *\*buflen*.

Definition at line 420 of file `bignum.c`.

References `ciL`, `ERR_MPI_BUFFER_TOO_SMALL`, `mpi_msb()`, and `mpi::p`.

Referenced by `dhm_calc_secret()`, `dhm_make_public()`, `rsa_private()`, and `rsa_public()`.

#### 8.89.4.36 `int mpi_write_file (char * p, mpi * X, int radix, FILE * fout)`

Write *X* into an opened file, or stdout.

##### Parameters:

*p* prefix, can be NULL  
*X* source [mpi](#)  
*radix* output numeric base  
*fout* output file handle

##### Returns:

0 if successful, or an ERR\_MPI\_XXX error code

##### Note:

Set *fout* == NULL to print *X* on the console.

Definition at line 360 of file `bignum.c`.

References `CHK`, `ERR_MPI_FILE_IO_ERROR`, and `mpi_write_string()`.

Referenced by `main()`, `rsa_write_private()`, and `rsa_write_public()`.

#### 8.89.4.37 `int mpi_write_string (mpi * X, int radix, char * s, int * slen)`

Export *X* into an ASCII string.

##### Parameters:

*X* source [mpi](#)

*radix* output numeric base

*s* string buffer

*slen* string buffer size

**Returns:**

0 if successful, or an ERR\_MPI\_XXX error code

**Note:**

Call this function with *\*slen* = 0 to obtain the minimum required buffer size in *\*slen*.

Definition at line 273 of file bignum.c.

References CHK, ciL, ERR\_MPI\_BUFFER\_TOO\_SMALL, ERR\_MPI\_INVALID\_PARAMETER, mpi\_copy(), mpi\_free(), mpi\_init(), mpi\_msb(), mpi\_write\_hlp(), mpi::n, mpi::p, and mpi::s.

Referenced by mpi\_write\_file().



## 8.90 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/bn\_asm.h File Reference

### Defines

- #define [MULADDC\\_INIT](#)
- #define [MULADDC\\_CORE](#)
- #define [MULADDC\\_STOP](#) }

### 8.90.1 Detailed Description

Multiply source vector [s] with b, add result to destination vector [d] and set carry c.

Currently supports:

. IA-32 (386+) . IA-32 (SSE2) . AMD64 / EM64T . Motorola 68000 . PowerPC, 32-bit . PowerPC, 64-bit  
 . SPARC v8 . MicroBlaze . TriCore . ARM v3+ . Alpha . MIPS32 . C, generic . C, longlong

Definition in file [bn\\_asm.h](#).

### 8.90.2 Define Documentation

#### 8.90.2.1 #define MULADDC\_CORE

##### Value:

```
s0 = ( *s << biH ) >> biH;          \
    s1 = ( *s >> biH ); s++;          \
    rx = s0 * b1; r0 = s0 * b0;        \
    ry = s1 * b0; r1 = s1 * b1;        \
    r1 += ( rx >> biH );               \
    r1 += ( ry >> biH );               \
    rx <<= biH; ry <<= biH;           \
    r0 += rx; r1 += (r0 < rx);          \
    r0 += ry; r1 += (r0 < ry);          \
    r0 += c; r1 += (r0 < c);            \
    r0 += *d; r1 += (r0 < *d);          \
    c = r1; *(d++) = r0;
```

Definition at line 597 of file [bn\\_asm.h](#).

Referenced by [mpi\\_mul\\_hlp\(\)](#).

#### 8.90.2.2 #define MULADDC\_INIT

##### Value:

```
{                                     \
    t_int s0, s1, b0, b1;            \
    t_int r0, r1, rx, ry;             \
    b0 = ( b << biH ) >> biH;         \
    b1 = ( b >> biH );
```

Definition at line 590 of file [bn\\_asm.h](#).

Referenced by [mpi\\_mul\\_hlp\(\)](#).

**8.90.2.3 #define MULADDC\_STOP }**

Definition at line 611 of file bn\_asm.h.

Referenced by mpi\_mul\_hlp().

## 8.91 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/certs.h File Reference

### Variables

- char [test\\_ca\\_cert](#) []
- char [test\\_ca\\_key](#) []
- char [test\\_ca\\_pwd](#) []
- char [test\\_srv\\_cert](#) []
- char [test\\_srv\\_key](#) []
- char [test\\_cli\\_cert](#) []
- char [test\\_cli\\_key](#) []
- char [xyssl\\_ca\\_cert](#) []

### 8.91.1 Detailed Description

Definition in file [certs.h](#).

### 8.91.2 Variable Documentation

#### 8.91.2.1 char test\_ca\_cert[]

Definition at line 24 of file [certs.c](#).

Referenced by [main\(\)](#).

#### 8.91.2.2 char test\_ca\_key[]

Definition at line 48 of file [certs.c](#).

#### 8.91.2.3 char test\_ca\_pwd[]

Definition at line 80 of file [certs.c](#).

#### 8.91.2.4 char test\_cli\_cert[]

Definition at line 133 of file [certs.c](#).

Referenced by [main\(\)](#).

#### 8.91.2.5 char test\_cli\_key[]

Definition at line 155 of file [certs.c](#).

Referenced by [main\(\)](#).

**8.91.2.6 char test\_srv\_crt[ ]**

Definition at line 82 of file certs.c.

Referenced by main().

**8.91.2.7 char test\_srv\_key[ ]**

Definition at line 104 of file certs.c.

Referenced by main().

**8.91.2.8 char xyssl\_ca\_crt[ ]**

Definition at line 184 of file certs.c.

Referenced by main().

## 8.92 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/des.h File Reference

### Data Structures

- struct [des\\_context](#)  
*DES context structure.*
- struct [des3\\_context](#)  
*Triple-DES context structure.*

### Functions

- void [des\\_set\\_key](#) ([des\\_context](#) \*ctx, unsigned char key[8])  
*DES key schedule (56-bit).*
- void [des\\_encrypt](#) ([des\\_context](#) \*ctx, unsigned char input[8], unsigned char output[8])  
*DES block encryption (ECB mode).*
- void [des\\_decrypt](#) ([des\\_context](#) \*ctx, unsigned char input[8], unsigned char output[8])  
*DES block decryption (ECB mode).*
- void [des\\_cbc\\_encrypt](#) ([des\\_context](#) \*ctx, unsigned char iv[8], unsigned char \*input, unsigned char \*output, [int](#) len)  
*DES-CBC buffer encryption.*
- void [des\\_cbc\\_decrypt](#) ([des\\_context](#) \*ctx, unsigned char iv[8], unsigned char \*input, unsigned char \*output, [int](#) len)  
*DES-CBC buffer decryption.*
- void [des3\\_set\\_2keys](#) ([des3\\_context](#) \*ctx, unsigned char key[16])  
*Triple-DES key schedule (112-bit).*
- void [des3\\_set\\_3keys](#) ([des3\\_context](#) \*ctx, unsigned char key[24])  
*Triple-DES key schedule (168-bit).*
- void [des3\\_encrypt](#) ([des3\\_context](#) \*ctx, unsigned char input[8], unsigned char output[8])  
*Triple-DES block encryption (ECB mode).*
- void [des3\\_decrypt](#) ([des3\\_context](#) \*ctx, unsigned char input[8], unsigned char output[8])  
*Triple-DES block decryption (ECB mode).*
- void [des3\\_cbc\\_encrypt](#) ([des3\\_context](#) \*ctx, unsigned char iv[8], unsigned char \*input, unsigned char \*output, [int](#) len)  
*3DES-CBC buffer encryption*
- void [des3\\_cbc\\_decrypt](#) ([des3\\_context](#) \*ctx, unsigned char iv[8], unsigned char \*input, unsigned char \*output, [int](#) len)

*3DES-CBC buffer decryption*

- [int des\\_self\\_test](#) ([int](#) verbose)

### 8.92.1 Detailed Description

Definition in file [des.h](#).

### 8.92.2 Function Documentation

#### 8.92.2.1 **void des3\_cbc\_decrypt** (**des3\_context** \* *ctx*, unsigned char *iv*[8], unsigned char \* *input*, unsigned char \* *output*, int *len*)

3DES-CBC buffer decryption

##### Parameters:

- ctx* 3DES context
- iv* initialization vector (modified after use)
- input* buffer holding the ciphertext
- output* buffer holding the plaintext
- len* length of the data to be decrypted

Definition at line 614 of file [des.c](#).

References [des3\\_crypt\(\)](#), and [des3\\_context::dsk](#).

Referenced by [ssl\\_decrypt\\_buf\(\)](#), and [x509\\_des3\\_decrypt\(\)](#).

#### 8.92.2.2 **void des3\_cbc\_encrypt** (**des3\_context** \* *ctx*, unsigned char *iv*[8], unsigned char \* *input*, unsigned char \* *output*, int *len*)

3DES-CBC buffer encryption

##### Parameters:

- ctx* 3DES context
- iv* initialization vector (modified after use)
- input* buffer holding the plaintext
- output* buffer holding the ciphertext
- len* length of the data to be encrypted

Definition at line 589 of file [des.c](#).

References [des3\\_crypt\(\)](#), and [des3\\_context::esk](#).

Referenced by [main\(\)](#), and [ssl\\_encrypt\\_buf\(\)](#).

### 8.92.2.3 void des3\_decrypt (des3\_context \* ctx, unsigned char input[8], unsigned char output[8])

Triple-DES block decryption (ECB mode).

#### Parameters:

*ctx* 3DES context  
*input* ciphertext block  
*output* plaintext block

Definition at line 579 of file des.c.

References des3\_crypt(), and des3\_context::dsk.

### 8.92.2.4 void des3\_encrypt (des3\_context \* ctx, unsigned char input[8], unsigned char output[8])

Triple-DES block encryption (ECB mode).

#### Parameters:

*ctx* 3DES context  
*input* plaintext block  
*output* ciphertext block

Definition at line 569 of file des.c.

References des3\_crypt(), and des3\_context::esk.

### 8.92.2.5 void des3\_set\_2keys (des3\_context \* ctx, unsigned char key[16])

Triple-DES key schedule (112-bit).

#### Parameters:

*ctx* 3DES context to be initialized  
*key* 16-byte secret key

Definition at line 475 of file des.c.

References des\_main\_ks(), des3\_context::dsk, and des3\_context::esk.

### 8.92.2.6 void des3\_set\_3keys (des3\_context \* ctx, unsigned char key[24])

Triple-DES key schedule (168-bit).

#### Parameters:

*ctx* 3DES context to be initialized  
*key* 24-byte secret key

Definition at line 501 of file des.c.

References des\_main\_ks(), des3\_context::dsk, and des3\_context::esk.

Referenced by main(), ssl\_derive\_keys(), and x509\_des3\_decrypt().

**8.92.2.7 void des\_cbc\_decrypt (des\_context \* ctx, unsigned char iv[8], unsigned char \* input, unsigned char \* output, int len)**

DES-CBC buffer decryption.

**Parameters:**

*ctx* DES context  
*iv* initialization vector (modified after use)  
*input* buffer holding the ciphertext  
*output* buffer holding the plaintext  
*len* length of the data to be decrypted

Definition at line 447 of file des.c.

References des\_crypt(), and des\_context::dsk.

**8.92.2.8 void des\_cbc\_encrypt (des\_context \* ctx, unsigned char iv[8], unsigned char \* input, unsigned char \* output, int len)**

DES-CBC buffer encryption.

**Parameters:**

*ctx* DES context  
*iv* initialization vector (modified after use)  
*input* buffer holding the plaintext  
*output* buffer holding the ciphertext  
*len* length of the data to be encrypted

Definition at line 422 of file des.c.

References des\_crypt(), and des\_context::esk.

Referenced by main().

**8.92.2.9 void des\_decrypt (des\_context \* ctx, unsigned char input[8], unsigned char output[8])**

DES block decryption (ECB mode).

**Parameters:**

*ctx* DES context  
*input* ciphertext block  
*output* plaintext block

Definition at line 412 of file des.c.

References des\_crypt(), and des\_context::dsk.



#### 8.92.2.10 void des\_encrypt (des\_context \* *ctx*, unsigned char *input*[8], unsigned char *output*[8])

DES block encryption (ECB mode).

##### Parameters:

*ctx* DES context  
*input* plaintext block  
*output* ciphertext block

Definition at line 402 of file des.c.

References des\_crypt(), and des\_context::esk.

#### 8.92.2.11 int des\_self\_test (int *verbose*)

Definition at line 736 of file des.c.

Referenced by main().

#### 8.92.2.12 void des\_set\_key (des\_context \* *ctx*, unsigned char *key*[8])

DES key schedule (56-bit).

##### Parameters:

*ctx* DES context to be initialized  
*key* 8-byte secret key

Definition at line 360 of file des.c.

References des\_main\_ks(), des\_context::dsk, and des\_context::esk.

Referenced by main().

## 8.93 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/dhm.h File Reference

```
#include "bignum.h"
```

### Data Structures

- struct [dhm\\_context](#)

### Defines

- #define [ERR\\_DHM\\_BAD\\_INPUT\\_DATA](#) 0x0380
- #define [ERR\\_DHM\\_READ\\_PARAMS\\_FAILED](#) 0x0390
- #define [ERR\\_DHM\\_MAKE\\_PARAMS\\_FAILED](#) 0x03A0
- #define [ERR\\_DHM\\_READ\\_PUBLIC\\_FAILED](#) 0x03B0
- #define [ERR\\_DHM\\_MAKE\\_PUBLIC\\_FAILED](#) 0x03C0
- #define [ERR\\_DHM\\_CALC\\_SECRET\\_FAILED](#) 0x03D0

### Functions

- [int dhm\\_read\\_params](#) ([dhm\\_context](#) \*ctx, unsigned char \*\*p, unsigned char \*end)  
*Parse the ServerKeyExchange parameters.*
- [int dhm\\_make\\_params](#) ([dhm\\_context](#) \*ctx, [int](#)(\*rng\_f)(void \*), void \*rng\_d, unsigned char \*output, [int](#) \*olen)  
*Setup and write the ServerKeyExchange parameters.*
- [int dhm\\_read\\_public](#) ([dhm\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*Import the peer's public value ( $G^Y$ ).*
- [int dhm\\_make\\_public](#) ([dhm\\_context](#) \*ctx, unsigned char \*output, [int](#) olen, [int](#)(\*rng\_f)(void \*), void \*rng\_d)  
*Create private value  $X$  and export  $G^X$ .*
- [int dhm\\_calc\\_secret](#) ([dhm\\_context](#) \*ctx, unsigned char \*output, [int](#) \*olen)  
*Derive and export the shared secret  $(G^Y)^X \bmod P$ .*
- void [dhm\\_free](#) ([dhm\\_context](#) \*ctx)
- [int dhm\\_self\\_test](#) ([int](#) verbose)  
*Checkup routine.*

#### 8.93.1 Detailed Description

Definition in file [dhm.h](#).

## 8.93.2 Define Documentation

### 8.93.2.1 #define ERR\_DHM\_BAD\_INPUT\_DATA 0x0380

Definition at line 16 of file dhm.h.

Referenced by dhm\_calc\_secret(), dhm\_make\_public(), dhm\_read\_bignum(), dhm\_read\_params(), and dhm\_read\_public().

### 8.93.2.2 #define ERR\_DHM\_CALC\_SECRET\_FAILED 0x03D0

Definition at line 21 of file dhm.h.

Referenced by dhm\_calc\_secret().

### 8.93.2.3 #define ERR\_DHM\_MAKE\_PARAMS\_FAILED 0x03A0

Definition at line 18 of file dhm.h.

Referenced by dhm\_make\_params().

### 8.93.2.4 #define ERR\_DHM\_MAKE\_PUBLIC\_FAILED 0x03C0

Definition at line 20 of file dhm.h.

Referenced by dhm\_make\_public().

### 8.93.2.5 #define ERR\_DHM\_READ\_PARAMS\_FAILED 0x0390

Definition at line 17 of file dhm.h.

Referenced by dhm\_read\_bignum().

### 8.93.2.6 #define ERR\_DHM\_READ\_PUBLIC\_FAILED 0x03B0

Definition at line 19 of file dhm.h.

Referenced by dhm\_read\_public().

## 8.93.3 Function Documentation

### 8.93.3.1 int dhm\_calc\_secret (dhm\_context \* *ctx*, unsigned char \* *output*, int \* *olen*)

Derive and export the shared secret  $(G^Y)^X \bmod P$ .

#### Parameters:

*ctx* DHM context

*output* destination buffer

*olen* number of chars written

**Returns:**

0 if successful, or ERR\_DHM\_MAKE\_PUBLIC\_FAILED

Definition at line 206 of file dhm.c.

References CHK, ERR\_DHM\_BAD\_INPUT\_DATA, ERR\_DHM\_CALC\_SECRET\_FAILED, dhm\_context::GY, dhm\_context::K, mpi\_exp\_mod(), mpi\_msb(), mpi\_write\_binary(), dhm\_context::P, dhm\_context::RP, and dhm\_context::X.

Referenced by main(), ssl\_parse\_client\_key\_exchange(), and ssl\_write\_client\_key\_exchange().

**8.93.3.2 void dhm\_free (dhm\_context \* ctx)**

Definition at line 232 of file dhm.c.

References dhm\_context::G, dhm\_context::GX, dhm\_context::GY, dhm\_context::K, mpi\_free(), dhm\_context::P, dhm\_context::RP, and dhm\_context::X.

Referenced by main(), and ssl\_free().

**8.93.3.3 int dhm\_make\_params (dhm\_context \* ctx, int(\*) (void \*) rng\_f, void \* rng\_d, unsigned char \* output, int \* olen)**

Setup and write the ServerKeyExchange parameters.

**Parameters:**

*ctx* DHM context  
*rng\_f* points to the RNG function  
*rng\_d* points to the RNG data  
*output* destination buffer  
*olen* number of chars written

**Note:**

This function assumes that ctx->P and ctx->G have already been properly set (for example using mpi\_read\_string).

**Returns:**

0 if successful, or an MPI error code

Definition at line 96 of file dhm.c.

References CHK, DHM\_MPI\_EXPORT, ERR\_DHM\_MAKE\_PARAMS\_FAILED, dhm\_context::G, dhm\_context::GX, dhm\_context::len, mpi\_cmp\_mpi(), mpi\_exp\_mod(), mpi\_grow(), mpi\_msb(), mpi\_shift\_r(), dhm\_context::P, mpi::p, dhm\_context::RP, and dhm\_context::X.

Referenced by main(), and ssl\_write\_server\_key\_exchange().

**8.93.3.4 int dhm\_make\_public (dhm\_context \* ctx, unsigned char \* output, int olen, int(\*) (void \*) rng\_f, void \* rng\_d)**

Create private value X and export  $G^X$ .

**Parameters:**

*ctx* DHM context  
*output* destination buffer  
*olen* must be == *ctx*->P.len  
*rng\_f* points to the RNG function  
*rng\_d* points to the RNG data

**Returns:**

0 if successful, or ERR\_DHM\_MAKE\_PUBLIC\_FAILED

Definition at line 167 of file dhm.c.

References CHK, ERR\_DHM\_BAD\_INPUT\_DATA, ERR\_DHM\_MAKE\_PUBLIC\_FAILED, dhm\_context::G, dhm\_context::GX, dhm\_context::len, mpi\_cmp\_mpi(), mpi\_exp\_mod(), mpi\_grow(), mpi\_shift\_r(), mpi\_write\_binary(), dhm\_context::P, mpi::p, dhm\_context::RP, and dhm\_context::X.

Referenced by main(), and ssl\_write\_client\_key\_exchange().

**8.93.3.5 int dhm\_read\_params (dhm\_context \* *ctx*, unsigned char \*\* *p*, unsigned char \* *end*)**

Parse the ServerKeyExchange parameters.

**Parameters:**

*ctx* DHM context  
*p* &(start of input buffer)  
*end* end of buffer

**Returns:**

0 if successful, or ERR\_DHM\_READ\_PARAMS\_FAILED

Definition at line 66 of file dhm.c.

References dhm\_read\_bignum(), ERR\_DHM\_BAD\_INPUT\_DATA, dhm\_context::G, dhm\_context::GY, dhm\_context::len, mpi\_msb(), and dhm\_context::P.

Referenced by main(), and ssl\_parse\_server\_key\_exchange().

**8.93.3.6 int dhm\_read\_public (dhm\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

Import the peer's public value ( $G^Y$ ).

**Parameters:**

*ctx* DHM context  
*input* input buffer  
*ilen* size of buffer

**Returns:**

0 if successful, or ERR\_DHM\_READ\_PUBLIC\_FAILED

Definition at line 150 of file dhm.c.

References `ERR_DHM_BAD_INPUT_DATA`, `ERR_DHM_READ_PUBLIC_FAILED`, `dhm_context::GY`, `dhm_context::len`, and `mpi_read_binary()`.

Referenced by `main()`, and `ssl_parse_client_key_exchange()`.

#### **8.93.3.7** `int dhm_self_test (int verbose)`

Checkup routine.

##### **Returns:**

0 if successful, or 1 if the test failed

Definition at line 246 of file dhm.c.

## 8.94 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/havege.h File Reference

### Data Structures

- struct [havege\\_state](#)  
*HAVEGE state structure.*

### Defines

- #define [COLLECT\\_SIZE](#) 1024

### Functions

- void [havege\\_init](#) ([havege\\_state](#) \*hs)  
*HAVEGE initialization.*
- int [havege\\_rand](#) (void \*rng\_d)  
*HAVEGE rand function.*

#### 8.94.1 Detailed Description

Definition in file [havege.h](#).

#### 8.94.2 Define Documentation

##### 8.94.2.1 #define COLLECT\_SIZE 1024

Definition at line 14 of file [havege.h](#).

Referenced by [havege\\_fill\(\)](#), and [havege\\_rand\(\)](#).

#### 8.94.3 Function Documentation

##### 8.94.3.1 void havege\_init (havege\_state \* hs)

HAVEGE initialization.

##### Parameters:

*hs* HAVEGE state to be initialized

Definition at line 191 of file [havege.c](#).

References [havege\\_fill\(\)](#).

Referenced by [main\(\)](#).

### 8.94.3.2 int havege\_rand (void \* *rng\_d*)

HAVEGE rand function.

**Parameters:**

*rng\_st* points to an HAVEGE state

**Returns:**

A random int

Definition at line 201 of file havege.c.

References COLLECT\_SIZE, havege\_fill(), havege\_state::offset, and havege\_state::pool.

Referenced by main().



## 8.95 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/md2.h File Reference

### Data Structures

- struct [md2\\_context](#)  
*MD2 context structure.*

### Functions

- void [md2\\_starts](#) ([md2\\_context](#) \*ctx)  
*MD2 context setup.*
- void [md2\\_update](#) ([md2\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*MD2 process buffer.*
- void [md2\\_finish](#) ([md2\\_context](#) \*ctx, unsigned char \*output)  
*MD2 final digest.*
- void [md2](#) (unsigned char \*input, [int](#) ilen, unsigned char \*output)  
*Output = MD2( input buffer ).*
- [int](#) [md2\\_file](#) (char \*path, unsigned char \*output)  
*Output = MD2( file contents ).*
- void [md2\\_hmac\\_starts](#) ([md2\\_context](#) \*ctx, unsigned char \*key, [int](#) keylen)  
*MD2 HMAC context setup.*
- void [md2\\_hmac\\_update](#) ([md2\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*MD2 HMAC process buffer.*
- void [md2\\_hmac\\_finish](#) ([md2\\_context](#) \*ctx, unsigned char \*output)  
*MD2 HMAC final digest.*
- void [md2\\_hmac](#) (unsigned char \*key, [int](#) keylen, unsigned char \*input, [int](#) ilen, unsigned char \*output)  
*Output = HMAC-MD2( hmac key, input buffer ).*
- [int](#) [md2\\_self\\_test](#) ([int](#) verbose)  
*Checkup routine.*

### 8.95.1 Detailed Description

Definition in file [md2.h](#).

## 8.95.2 Function Documentation

### 8.95.2.1 void md2 (unsigned char \* *input*, int *ilen*, unsigned char \* *output*)

Output = MD2( input buffer ).

**Parameters:**

*input* buffer holding the data  
*ilen* length of the input data  
*output* MD2 checksum result

Definition at line 154 of file md2.c.

References md2\_finish(), md2\_starts(), and md2\_update().

Referenced by x509\_hash().

### 8.95.2.2 int md2\_file (char \* *path*, unsigned char \* *output*)

Output = MD2( file contents ).

**Parameters:**

*path* input file name  
*output* MD2 checksum result

**Returns:**

0 if successful, 1 if fopen failed, or 2 if fread failed

Definition at line 169 of file md2.c.

References f, md2\_finish(), md2\_starts(), and md2\_update().

### 8.95.2.3 void md2\_finish (md2\_context \* *ctx*, unsigned char \* *output*)

MD2 final digest.

**Parameters:**

*ctx* MD2 context  
*output* MD2 checksum result

Definition at line 133 of file md2.c.

References md2\_context::buffer, md2\_context::cksum, md2\_context::left, md2\_process(), and md2\_context::state.

Referenced by md2(), md2\_file(), and md2\_hmac\_finish().

**8.95.2.4 void md2\_hmac (unsigned char \* *key*, int *keylen*, unsigned char \* *input*, int *ilen*, unsigned char \* *output*)**

Output = HMAC-MD2( hmac key, input buffer ).

**Parameters:**

*key* HMAC secret key

*keylen* length of the HMAC key

*input* buffer holding the data

*ilen* length of the input data

*output* HMAC-MD2 result

Definition at line 249 of file md2.c.

References md2\_hmac\_finish(), md2\_hmac\_starts(), and md2\_hmac\_update().

**8.95.2.5 void md2\_hmac\_finish (md2\_context \* *ctx*, unsigned char \* *output*)**

MD2 HMAC final digest.

**Parameters:**

*ctx* HMAC context

*output* MD2 HMAC checksum result

Definition at line 233 of file md2.c.

References md2\_finish(), md2\_starts(), md2\_update(), and md2\_context::opad.

Referenced by md2\_hmac().

**8.95.2.6 void md2\_hmac\_starts (md2\_context \* *ctx*, unsigned char \* *key*, int *keylen*)**

MD2 HMAC context setup.

**Parameters:**

*ctx* HMAC context to be initialized

*key* HMAC secret key

*keylen* length of the HMAC key

Definition at line 201 of file md2.c.

References md2\_context::ipad, md2\_starts(), md2\_update(), and md2\_context::opad.

Referenced by md2\_hmac().

**8.95.2.7 void md2\_hmac\_update (md2\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

MD2 HMAC process buffer.

**Parameters:**

*ctx* HMAC context  
*input* buffer holding the data  
*ilen* length of the input data

Definition at line 224 of file md2.c.

References md2\_update().

Referenced by md2\_hmac().

**8.95.2.8 int md2\_self\_test (int verbose)**

Checksum routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 330 of file md2.c.

Referenced by main().

**8.95.2.9 void md2\_starts (md2\_context \* ctx)**

MD2 context setup.

**Parameters:**

*ctx* context to be initialized

Definition at line 72 of file md2.c.

Referenced by md2(), md2\_file(), md2\_hmac\_finish(), and md2\_hmac\_starts().

**8.95.2.10 void md2\_update (md2\_context \* ctx, unsigned char \* input, int ilen)**

MD2 process buffer.

**Parameters:**

*ctx* MD2 context  
*input* buffer holding the data  
*ilen* length of the input data

Definition at line 105 of file md2.c.

References md2\_context::buffer, md2\_context::left, and md2\_process().

Referenced by md2(), md2\_file(), md2\_hmac\_finish(), md2\_hmac\_starts(), and md2\_hmac\_update().

## 8.96 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/md4.h File Reference

### Data Structures

- struct [md4\\_context](#)  
*MD4 context structure.*

### Functions

- void [md4\\_starts](#) ([md4\\_context](#) \*ctx)  
*MD4 context setup.*
- void [md4\\_update](#) ([md4\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*MD4 process buffer.*
- void [md4\\_finish](#) ([md4\\_context](#) \*ctx, unsigned char \*output)  
*MD4 final digest.*
- void [md4](#) (unsigned char \*input, [int](#) ilen, unsigned char \*output)  
*Output = MD4( input buffer ).*
- [int](#) [md4\\_file](#) (char \*path, unsigned char \*output)  
*Output = MD4( file contents ).*
- void [md4\\_hmac\\_starts](#) ([md4\\_context](#) \*ctx, unsigned char \*key, [int](#) keylen)  
*MD4 HMAC context setup.*
- void [md4\\_hmac\\_update](#) ([md4\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*MD4 HMAC process buffer.*
- void [md4\\_hmac\\_finish](#) ([md4\\_context](#) \*ctx, unsigned char \*output)  
*MD4 HMAC final digest.*
- void [md4\\_hmac](#) (unsigned char \*key, [int](#) keylen, unsigned char \*input, [int](#) ilen, unsigned char \*output)  
*Output = HMAC-MD4( hmac key, input buffer ).*
- [int](#) [md4\\_self\\_test](#) ([int](#) verbose)  
*Checkup routine.*

### 8.96.1 Detailed Description

Definition in file [md4.h](#).

## 8.96.2 Function Documentation

### 8.96.2.1 void md4 (unsigned char \* *input*, int *ilen*, unsigned char \* *output*)

Output = MD4( input buffer ).

**Parameters:**

*input* buffer holding the data  
*ilen* length of the input data  
*output* MD4 checksum result

Definition at line 262 of file md4.c.

References md4\_finish(), md4\_starts(), and md4\_update().

Referenced by main(), and x509\_hash().

### 8.96.2.2 int md4\_file (char \* *path*, unsigned char \* *output*)

Output = MD4( file contents ).

**Parameters:**

*path* input file name  
*output* MD4 checksum result

**Returns:**

0 if successful, 1 if fopen failed, or 2 if fread failed

Definition at line 277 of file md4.c.

References f, md4\_finish(), md4\_starts(), and md4\_update().

### 8.96.2.3 void md4\_finish (md4\_context \* *ctx*, unsigned char \* *output*)

MD4 final digest.

**Parameters:**

*ctx* MD4 context  
*output* MD4 checksum result

Definition at line 234 of file md4.c.

References md4\_padding, md4\_update(), PUT\_UINT32\_LE, md4\_context::state, and md4\_context::total.

Referenced by md4(), md4\_file(), and md4\_hmac\_finish().

### 8.96.2.4 void md4\_hmac (unsigned char \* *key*, int *keylen*, unsigned char \* *input*, int *ilen*, unsigned char \* *output*)

Output = HMAC-MD4( hmac key, input buffer ).

**Parameters:**

*key* HMAC secret key  
*keylen* length of the HMAC key  
*input* buffer holding the data  
*ilen* length of the input data  
*output* HMAC-MD4 result

Definition at line 357 of file md4.c.

References md4\_hmac\_finish(), md4\_hmac\_starts(), and md4\_hmac\_update().

**8.96.2.5 void md4\_hmac\_finish (md4\_context \* *ctx*, unsigned char \* *output*)**

MD4 HMAC final digest.

**Parameters:**

*ctx* HMAC context  
*output* MD4 HMAC checksum result

Definition at line 341 of file md4.c.

References md4\_finish(), md4\_starts(), md4\_update(), and md4\_context::opad.

Referenced by md4\_hmac().

**8.96.2.6 void md4\_hmac\_starts (md4\_context \* *ctx*, unsigned char \* *key*, int *keylen*)**

MD4 HMAC context setup.

**Parameters:**

*ctx* HMAC context to be initialized  
*key* HMAC secret key  
*keylen* length of the HMAC key

Definition at line 309 of file md4.c.

References md4\_context::ipad, md4\_starts(), md4\_update(), and md4\_context::opad.

Referenced by md4\_hmac().

**8.96.2.7 void md4\_hmac\_update (md4\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

MD4 HMAC process buffer.

**Parameters:**

*ctx* HMAC context  
*input* buffer holding the data  
*ilen* length of the input data

Definition at line 332 of file md4.c.

References md4\_update().

Referenced by md4\_hmac().

#### 8.96.2.8 int md4\_self\_test (int *verbose*)

Checkup routine.

##### Returns:

0 if successful, or 1 if the test failed

Definition at line 438 of file md4.c.

Referenced by main().

#### 8.96.2.9 void md4\_starts (md4\_context \* *ctx*)

MD4 context setup.

##### Parameters:

*ctx* context to be initialized

Definition at line 65 of file md4.c.

References md4\_context::state, and md4\_context::total.

Referenced by md4(), md4\_file(), md4\_hmac\_finish(), and md4\_hmac\_starts().

#### 8.96.2.10 void md4\_update (md4\_context \* *ctx*, unsigned char \* *input*, int *ilen*)

MD4 process buffer.

##### Parameters:

*ctx* MD4 context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 182 of file md4.c.

References md4\_context::buffer, md4\_process(), and md4\_context::total.

Referenced by md4(), md4\_file(), md4\_finish(), md4\_hmac\_finish(), md4\_hmac\_starts(), and md4\_hmac\_update().



## 8.97 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/md5.h File Reference

### Data Structures

- struct [md5\\_context](#)  
*MD5 context structure.*

### Functions

- void [md5\\_starts](#) ([md5\\_context](#) \*ctx)  
*MD5 context setup.*
- void [md5\\_update](#) ([md5\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*MD5 process buffer.*
- void [md5\\_finish](#) ([md5\\_context](#) \*ctx, unsigned char \*output)  
*MD5 final digest.*
- void [md5](#) (unsigned char \*input, [int](#) ilen, unsigned char \*output)  
*Output = MD5( input buffer ).*
- [int](#) [md5\\_file](#) (char \*path, unsigned char \*output)  
*Output = MD5( file contents ).*
- void [md5\\_hmac\\_starts](#) ([md5\\_context](#) \*ctx, unsigned char \*key, [int](#) keylen)  
*MD5 HMAC context setup.*
- void [md5\\_hmac\\_update](#) ([md5\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*MD5 HMAC process buffer.*
- void [md5\\_hmac\\_finish](#) ([md5\\_context](#) \*ctx, unsigned char \*output)  
*MD5 HMAC final digest.*
- void [md5\\_hmac](#) (unsigned char \*key, [int](#) keylen, unsigned char \*input, [int](#) ilen, unsigned char \*output)  
*Output = HMAC-MD5( hmac key, input buffer ).*
- [int](#) [md5\\_self\\_test](#) ([int](#) verbose)  
*Checkup routine.*

### 8.97.1 Detailed Description

Definition in file [md5.h](#).

## 8.97.2 Function Documentation

### 8.97.2.1 void md5 (unsigned char \* *input*, int *ilen*, unsigned char \* *output*)

Output = MD5( input buffer ).

#### Parameters:

*input* buffer holding the data  
*ilen* length of the input data  
*output* MD5 checksum result

Definition at line 281 of file md5.c.

References md5\_finish(), md5\_starts(), and md5\_update().

Referenced by main(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_md5(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_server\_key\_exchange(), and x509\_hash().

### 8.97.2.2 int md5\_file (char \* *path*, unsigned char \* *output*)

Output = MD5( file contents ).

#### Parameters:

*path* input file name  
*output* MD5 checksum result

#### Returns:

0 if successful, 1 if fopen failed, or 2 if fread failed

Definition at line 296 of file md5.c.

References f, md5\_finish(), md5\_starts(), and md5\_update().

Referenced by md5\_wrapper().

### 8.97.2.3 void md5\_finish (md5\_context \* *ctx*, unsigned char \* *output*)

MD5 final digest.

#### Parameters:

*ctx* MD5 context  
*output* MD5 checksum result

Definition at line 253 of file md5.c.

References md5\_padding, md5\_update(), PUT\_UINT32\_LE, md5\_context::state, and md5\_context::total.

Referenced by md5(), md5\_file(), md5\_hmac\_finish(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_md5(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_server\_key\_exchange(), and x509\_des3\_decrypt().

**8.97.2.4 void md5\_hmac (unsigned char \* *key*, int *keylen*, unsigned char \* *input*, int *ilen*, unsigned char \* *output*)**

Output = HMAC-MD5( hmac key, input buffer ).

**Parameters:**

*key* HMAC secret key

*keylen* length of the HMAC key

*input* buffer holding the data

*ilen* length of the input data

*output* HMAC-MD5 result

Definition at line 376 of file md5.c.

References md5\_hmac\_finish(), md5\_hmac\_starts(), and md5\_hmac\_update().

Referenced by ssl\_decrypt\_buf(), ssl\_encrypt\_buf(), and tls1\_prf().

**8.97.2.5 void md5\_hmac\_finish (md5\_context \* *ctx*, unsigned char \* *output*)**

MD5 HMAC final digest.

**Parameters:**

*ctx* HMAC context

*output* MD5 HMAC checksum result

Definition at line 360 of file md5.c.

References md5\_finish(), md5\_starts(), md5\_update(), and md5\_context::opad.

Referenced by md5\_hmac().

**8.97.2.6 void md5\_hmac\_starts (md5\_context \* *ctx*, unsigned char \* *key*, int *keylen*)**

MD5 HMAC context setup.

**Parameters:**

*ctx* HMAC context to be initialized

*key* HMAC secret key

*keylen* length of the HMAC key

Definition at line 328 of file md5.c.

References md5\_context::ipad, md5\_starts(), md5\_update(), and md5\_context::opad.

Referenced by md5\_hmac().

**8.97.2.7 void md5\_hmac\_update (md5\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

MD5 HMAC process buffer.

**Parameters:**

*ctx* HMAC context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 351 of file md5.c.

References md5\_update().

Referenced by md5\_hmac().

**8.97.2.8 int md5\_self\_test (int *verbose*)**

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 457 of file md5.c.

Referenced by main().

**8.97.2.9 void md5\_starts (md5\_context \* *ctx*)**

MD5 context setup.

**Parameters:**

*ctx* context to be initialized

Definition at line 64 of file md5.c.

References md5\_context::state, and md5\_context::total.

Referenced by md5(), md5\_file(), md5\_hmac\_finish(), md5\_hmac\_starts(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_md5(), ssl\_parse\_client\_hello(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_client\_hello(), ssl\_write\_server\_key\_exchange(), and x509\_des3\_decrypt().

**8.97.2.10 void md5\_update (md5\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

MD5 process buffer.

**Parameters:**

*ctx* MD5 context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 201 of file md5.c.

References md5\_context::buffer, md5\_process(), and md5\_context::total.

Referenced by md5(), md5\_file(), md5\_finish(), md5\_hmac\_finish(), md5\_hmac\_starts(), md5\_hmac\_update(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_md5(), ssl\_parse\_client\_hello(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_record(), ssl\_write\_server\_key\_exchange(), and x509\_des3\_decrypt().

## 8.98 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/net.h File Reference

### Defines

- `#define ERR_NET_UNKNOWN_HOST 0xFF10`
- `#define ERR_NET_SOCKET_FAILED 0xFF20`
- `#define ERR_NET_CONNECT_FAILED 0xFF20`
- `#define ERR_NET_BIND_FAILED 0xFF30`
- `#define ERR_NET_LISTEN_FAILED 0xFF40`
- `#define ERR_NET_ACCEPT_FAILED 0xFF50`
- `#define ERR_NET_RECV_FAILED 0xFF60`
- `#define ERR_NET_SEND_FAILED 0xFF70`
- `#define ERR_NET_CONN_RESET 0xFF80`
- `#define ERR_NET_WOULD_BLOCK 0xFF90`

### Functions

- `int net_connect (int *fd, char *host, unsigned int port)`  
*Initiate a TCP connection with host:port.*
- `int net_bind (int *fd, char *bind_ip, unsigned int port)`  
*Create a listening socket on bind\_ip:port. If bind\_ip == NULL, all interfaces are binded.*
- `int net_accept (int bind_fd, int *client_fd, unsigned char client_ip[4])`  
*Accept a connection from a remote client.*
- `int net_set_nonblock (int fd)`  
*Set the socket non-blocking.*
- `void net_usleep (unsigned long usec)`  
*Portable usleep helper.*
- `int net_recv (int fd, unsigned char *buf, int *len)`  
*Read at most 'len' characters. len is updated to reflect the actual number of characters read.*
- `int net_send (int fd, unsigned char *buf, int *len)`  
*Write at most 'len' characters. len is updated to reflect the number of characters \_not\_ written.*
- `void net_close (int fd)`  
*Gracefully shutdown the connection.*

### 8.98.1 Detailed Description

Definition in file `net.h`.

## 8.98.2 Define Documentation

### 8.98.2.1 **#define ERR\_NET\_ACCEPT\_FAILED 0xFF50**

Definition at line 19 of file net.h.

Referenced by net\_accept().

### 8.98.2.2 **#define ERR\_NET\_BIND\_FAILED 0xFF30**

Definition at line 17 of file net.h.

Referenced by net\_bind().

### 8.98.2.3 **#define ERR\_NET\_CONN\_RESET 0xFF80**

Definition at line 22 of file net.h.

Referenced by main(), net\_recv(), and net\_send().

### 8.98.2.4 **#define ERR\_NET\_CONNECT\_FAILED 0xFF20**

Definition at line 16 of file net.h.

Referenced by net\_connect().

### 8.98.2.5 **#define ERR\_NET\_LISTEN\_FAILED 0xFF40**

Definition at line 18 of file net.h.

Referenced by net\_bind().

### 8.98.2.6 **#define ERR\_NET\_RECV\_FAILED 0xFF60**

Definition at line 20 of file net.h.

Referenced by net\_recv().

### 8.98.2.7 **#define ERR\_NET\_SEND\_FAILED 0xFF70**

Definition at line 21 of file net.h.

Referenced by net\_send().

### 8.98.2.8 **#define ERR\_NET\_SOCKET\_FAILED 0xFF20**

Definition at line 15 of file net.h.

Referenced by net\_bind(), and net\_connect().

### 8.98.2.9 `#define ERR_NET_UNKNOWN_HOST 0xFF10`

Definition at line 14 of file net.h.

Referenced by net\_connect().

### 8.98.2.10 `#define ERR_NET_WOULD_BLOCK 0xFF90`

Definition at line 23 of file net.h.

Referenced by main(), net\_accept(), net\_recv(), and net\_send().

## 8.98.3 Function Documentation

### 8.98.3.1 `int net_accept (int bind_fd, int * client_fd, unsigned char client_ip[4])`

Accept a connection from a remote client.

#### Returns:

0 if successful, ERR\_NET\_ACCEPT\_FAILED, or ERR\_NET\_WOULD\_BLOCK is bind\_fd was set to non-blocking and accept() is blocking.

Definition at line 203 of file net.c.

References ERR\_NET\_ACCEPT\_FAILED, ERR\_NET\_WOULD\_BLOCK, int, and net\_is\_blocking().

Referenced by main().

### 8.98.3.2 `int net_bind (int * fd, char * bind_ip, unsigned int port)`

Create a listening socket on bind\_ip:port. If bind\_ip == NULL, all interfaces are binded.

#### Returns:

0 if successful, or one of: ERR\_NET\_SOCKET\_FAILED, ERR\_NET\_BIND\_FAILED, ERR\_NET\_LISTEN\_FAILED

Definition at line 115 of file net.c.

References ERR\_NET\_BIND\_FAILED, ERR\_NET\_LISTEN\_FAILED, and ERR\_NET\_SOCKET\_FAILED.

Referenced by main().

### 8.98.3.3 `void net_close (int fd)`

Gracefully shutdown the connection.

Definition at line 340 of file net.c.

Referenced by main().



#### 8.98.3.4 int net\_connect (int \*fd, char \*host, unsigned int port)

Initiate a TCP connection with host:port.

##### Returns:

0 if successful, or one of: ERR\_NET\_SOCKET\_FAILED, ERR\_NET\_UNKNOWN\_HOST, ERR\_NET\_CONNECT\_FAILED

Definition at line 70 of file net.c.

References ERR\_NET\_CONNECT\_FAILED, ERR\_NET\_SOCKET\_FAILED, and ERR\_NET\_UNKNOWN\_HOST.

Referenced by main().

#### 8.98.3.5 int net\_recv (int fd, unsigned char \*buf, int \*len)

Read at most 'len' characters. len is updated to reflect the actual number of characters read.

##### Returns:

0 if successful, ERR\_NET\_CONN\_RESET if the connection was closed from the other side, or ERR\_NET\_WOULD\_BLOCK if read() is blocking.

Definition at line 268 of file net.c.

References ERR\_NET\_CONN\_RESET, ERR\_NET\_RECV\_FAILED, ERR\_NET\_WOULD\_BLOCK, and net\_is\_blocking().

Referenced by main(), and ssl\_parse\_client\_hello().

#### 8.98.3.6 int net\_send (int fd, unsigned char \*buf, int \*len)

Write at most 'len' characters. len is updated to reflect the number of characters \_not\_ written.

##### Returns:

0 if successful, ERR\_NET\_CONN\_RESET if the connection was closed from the other side, or ERR\_NET\_WOULD\_BLOCK if write() is blocking.

Definition at line 305 of file net.c.

References ERR\_NET\_CONN\_RESET, ERR\_NET\_SEND\_FAILED, ERR\_NET\_WOULD\_BLOCK, and net\_is\_blocking().

Referenced by main(), ssl\_flush\_output(), and ssl\_write\_record().

#### 8.98.3.7 int net\_set\_nonblock (int fd)

Set the socket non-blocking.

##### Returns:

0 if successful, or 1 if the operation failed

Definition at line 244 of file net.c.

**8.98.3.8 void net\_usleep (unsigned long *usec*)**

Portable usleep helper.

**Note:**

Real amount of time slept will not be less than select()'s timeout granularity (typically, 10ms).

Definition at line 257 of file net.c.

## 8.99 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/rsa.h File Reference

```
#include "bignum.h"
```

### Data Structures

- struct [rsa\\_context](#)

### Defines

- #define [ERR\\_RSA\\_BAD\\_INPUT\\_DATA](#) 0x0300
- #define [ERR\\_RSA\\_INVALID\\_PADDING](#) 0x0310
- #define [ERR\\_RSA\\_KEY\\_GEN\\_FAILED](#) 0x0320
- #define [ERR\\_RSA\\_KEY\\_CHK\\_FAILED](#) 0x0330
- #define [ERR\\_RSA\\_KEY\\_RD\\_FAILED](#) 0x0340
- #define [ERR\\_RSA\\_KEY\\_WR\\_FAILED](#) 0x0350
- #define [ERR\\_RSA\\_PUBLIC\\_FAILED](#) 0x0360
- #define [ERR\\_RSA\\_PRIVATE\\_FAILED](#) 0x0370
- #define [ERR\\_RSA\\_VERIFY\\_FAILED](#) 0x0380
- #define [RSA\\_RAW](#) 0
- #define [RSA\\_MD2](#) 2
- #define [RSA\\_MD4](#) 3
- #define [RSA\\_MD5](#) 4
- #define [RSA\\_SHA1](#) 5
- #define [RSA\\_SIGN](#) 0x01
- #define [RSA\\_CRYPT](#) 0x02
- #define [ASN1\\_HASH\\_MDX](#)
- #define [ASN1\\_HASH\\_SHA1](#)
- #define [KEY\\_LEN](#) 128
- #define [RSA\\_N](#)
- #define [RSA\\_E](#) "10001"
- #define [RSA\\_D](#)
- #define [RSA\\_P](#)
- #define [RSA\\_Q](#)
- #define [RSA\\_DP](#)
- #define [RSA\\_DQ](#)
- #define [RSA\\_QP](#)

### Functions

- [int rsa\\_gen\\_key](#) ([rsa\\_context](#) \*ctx, [int](#) nbits, [int](#) exponent, [int](#)(\*rng\_f)(void \*), void \*rng\_d)  
*Generate an RSA keypair.*
- [int rsa\\_read\\_public](#) ([rsa\\_context](#) \*ctx, FILE \*f)  
*Read the public key from a file.*
- [int rsa\\_read\\_private](#) ([rsa\\_context](#) \*ctx, FILE \*f)

*Read the private key from a file.*

- `int rsa_write_public (rsa_context *ctx, FILE *f)`

*Write the public key into a file.*

- `int rsa_write_private (rsa_context *ctx, FILE *f)`

*Write the private key into a file.*

- `int rsa_public (rsa_context *ctx, unsigned char *input, int ilen, unsigned char *output, int olen)`

*Perform an RSA public key operation.*

- `int rsa_private (rsa_context *ctx, unsigned char *input, int ilen, unsigned char *output, int olen)`

*Perform an RSA private key operation.*

- `int rsa_check_pubkey (rsa_context *ctx)`

*Return 0 if the public key is valid, or ERR\_RSA\_KEY\_CHECK\_FAILED.*

- `int rsa_check_privkey (rsa_context *ctx)`

*Return 0 if the private key is valid, or ERR\_RSA\_KEY\_CHECK\_FAILED.*

- `int rsa_pkcs1_encrypt (rsa_context *ctx, unsigned char *input, int ilen, unsigned char *output, int olen)`

*Add the PKCS#1 v1.5 padding and do a public RSA.*

- `int rsa_pkcs1_decrypt (rsa_context *ctx, unsigned char *input, int ilen, unsigned char *output, int *olen)`

*Do a private RSA, removes the PKCS#1 v1.5 padding.*

- `int rsa_pkcs1_sign (rsa_context *ctx, int alg_id, unsigned char *hash, int hashlen, unsigned char *sig, int siglen)`

*Perform a private RSA to sign a message digest.*

- `int rsa_pkcs1_verify (rsa_context *ctx, int alg_id, unsigned char *hash, int hashlen, unsigned char *sig, int siglen)`

*Perform a public RSA and check the message digest.*

- `void rsa_free (rsa_context *ctx)`

*Free the components of an RSA key.*

- `int rsa_self_test (int verbose)`

*Checkup routine.*

## 8.99.1 Detailed Description

Definition in file [rsa.h](#).

## 8.99.2 Define Documentation

### 8.99.2.1 #define ASN1\_HASH\_MDX

**Value:**

```
"\x30\x20\x30\x0C\x06\x08\x2A\x86\x48" \
"\x86\xF7\x0D\x02\x00\x05\x00\x04\x10"
```

Definition at line 47 of file rsa.h.

Referenced by rsa\_pkcs1\_sign(), and rsa\_pkcs1\_verify().

### 8.99.2.2 #define ASN1\_HASH\_SHA1

**Value:**

```
"\x30\x21\x30\x09\x06\x05\x2B\x0E\x03" \
"\x02\x1A\x05\x00\x04\x14"
```

Definition at line 51 of file rsa.h.

Referenced by rsa\_pkcs1\_sign(), and rsa\_pkcs1\_verify().

### 8.99.2.3 #define ERR\_RSA\_BAD\_INPUT\_DATA 0x0300

Definition at line 16 of file rsa.h.

Referenced by rsa\_gen\_key(), rsa\_pkcs1\_decrypt(), rsa\_pkcs1\_encrypt(), rsa\_pkcs1\_sign(), rsa\_pkcs1\_verify(), rsa\_private(), and rsa\_public().

### 8.99.2.4 #define ERR\_RSA\_INVALID\_PADDING 0x0310

Definition at line 17 of file rsa.h.

Referenced by rsa\_pkcs1\_decrypt(), and rsa\_pkcs1\_verify().

### 8.99.2.5 #define ERR\_RSA\_KEY\_CHK\_FAILED 0x0330

Definition at line 19 of file rsa.h.

Referenced by rsa\_check\_privkey(), and rsa\_check\_pubkey().

### 8.99.2.6 #define ERR\_RSA\_KEY\_GEN\_FAILED 0x0320

Definition at line 18 of file rsa.h.

Referenced by rsa\_gen\_key().

### 8.99.2.7 #define ERR\_RSA\_KEY\_RD\_FAILED 0x0340

Definition at line 20 of file rsa.h.

Referenced by rsa\_read\_private(), and rsa\_read\_public().

**8.99.2.8 #define ERR\_RSA\_KEY\_WR\_FAILED 0x0350**

Definition at line 21 of file rsa.h.

Referenced by rsa\_write\_private(), and rsa\_write\_public().

**8.99.2.9 #define ERR\_RSA\_PRIVATE\_FAILED 0x0370**

Definition at line 23 of file rsa.h.

Referenced by rsa\_private().

**8.99.2.10 #define ERR\_RSA\_PUBLIC\_FAILED 0x0360**

Definition at line 22 of file rsa.h.

Referenced by rsa\_public().

**8.99.2.11 #define ERR\_RSA\_VERIFY\_FAILED 0x0380**

Definition at line 24 of file rsa.h.

Referenced by rsa\_pkcs1\_verify().

**8.99.2.12 #define KEY\_LEN 128**

Definition at line 252 of file rsa.h.

Referenced by main().

**8.99.2.13 #define RSA\_CRYPT 0x02**

Definition at line 36 of file rsa.h.

Referenced by rsa\_pkcs1\_decrypt(), and rsa\_pkcs1\_encrypt().

**8.99.2.14 #define RSA\_D**

**Value:**

```
"24BF6185468786FDD303083D25E64EFC" \
    "66CA472BC44D253102F8B4A9D3BFA750" \
    "91386C0077937FE33FA3252D28855837" \
    "AE1B484A8A9A45F7EE8C0C634F99E8CD" \
    "DF79C5CE07EE72C7F123142198164234" \
    "CABB724CF78B8173B9F880FC86322407" \
    "AF1FEDFDDE2BEB674CA15F3E81A1521E" \
    "071513A1E85B5DFA031F21ECAE91A34D"
```

Definition at line 265 of file rsa.h.

Referenced by main().

#### 8.99.2.15 #define RSA\_DP

**Value:**

```
"C1ACF567564274FB07A0BBAD5D26E298" \
    "3C94D22288ACD763FD8E5600ED4A702D" \
    "F84198A5F06C2E72236AE490C93F07F8" \
    "3CC559CD27BC2D1CA488811730BB5725"
```

Definition at line 284 of file rsa.h.

Referenced by main().

#### 8.99.2.16 #define RSA\_DQ

**Value:**

```
"4959CBF6F8FEF750AEE6977C155579C7" \
    "D8AAEA56749EA28623272E4F7D0592AF" \
    "7C1F1313CAC9471B5C523BFE592F517B" \
    "407A1BD76C164B93DA2D32A383E58357"
```

Definition at line 289 of file rsa.h.

Referenced by main().

#### 8.99.2.17 #define RSA\_E "10001"

Definition at line 263 of file rsa.h.

Referenced by main().

#### 8.99.2.18 #define RSA\_MD2 2

Definition at line 30 of file rsa.h.

Referenced by rsa\_pkcs1\_sign(), rsa\_pkcs1\_verify(), x509\_cert\_info(), and x509\_hash().

#### 8.99.2.19 #define RSA\_MD4 3

Definition at line 31 of file rsa.h.

Referenced by rsa\_pkcs1\_sign(), rsa\_pkcs1\_verify(), x509\_cert\_info(), and x509\_hash().

#### 8.99.2.20 #define RSA\_MD5 4

Definition at line 32 of file rsa.h.

Referenced by rsa\_pkcs1\_sign(), rsa\_pkcs1\_verify(), x509\_cert\_info(), and x509\_hash().

#### 8.99.2.21 #define RSA\_N

**Value:**

```
"9292758453063D803DD603D5E777D788" \
    "8ED1D5BF35786190FA2F23EBC0848AEA" \
    "DDA92CA6C3D80B32C4D109BE0F36D6AE" \
    "7130B9CED7ACDF54CFC7555AC14EEBAB" \
    "93A89813FBF3C4F8066D2D800F7C38A8" \
    "1AE31942917403FF4946B0A83D3D3E05" \
    "EE57C6F5F5606FB5D4BC6CD34EE0801A" \
    "5E94BB77B07507233A0BC7BAC8F90F79"
```

Definition at line 254 of file rsa.h.

Referenced by main().

#### 8.99.2.22 #define RSA\_P

**Value:**

```
"C36D0EB7FCD285223CFB5AABA5BDA3D8" \
    "2C01CAD19EA484A87EA4377637E75500" \
    "FCB2005C5C7DD6EC4AC023CDA285D796" \
    "C3D9E75E1EFC42488BB4F1D13AC30A57"
```

Definition at line 274 of file rsa.h.

Referenced by main().

#### 8.99.2.23 #define RSA\_Q

**Value:**

```
"C000DF51A7C77AE8D7C7370C1FF55B69" \
    "E211C2B9E5DB1ED0BF61D0D9899620F4" \
    "910E4168387E3C30AA1E00C339A79508" \
    "8452DD96A9A5EA5D9DCA68DA636032AF"
```

Definition at line 279 of file rsa.h.

Referenced by main().

#### 8.99.2.24 #define RSA\_QP

**Value:**

```
"9AE7FBC99546432DF71896FC239EADAE" \
    "F38D18D2B2F0E2DD275AA977E2BF4411" \
    "F5A3B2A5D33605AEBBCCBA7FEB9F2D2F" \
    "A74206CEC169D74BF5A8C50D6F48EA08"
```

Definition at line 294 of file rsa.h.

Referenced by main().

#### 8.99.2.25 #define RSA\_RAW 0

Definition at line 29 of file rsa.h.

Referenced by rsa\_pkcs1\_sign(), rsa\_pkcs1\_verify(), ssl\_parse\_certificate\_verify(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_certificate\_verify(), and ssl\_write\_server\_key\_exchange().



**8.99.2.26 #define RSA\_SHA1 5**

Definition at line 33 of file rsa.h.

Referenced by main(), rsa\_pkcs1\_sign(), rsa\_pkcs1\_verify(), x509\_cert\_info(), and x509\_hash().

**8.99.2.27 #define RSA\_SIGN 0x01**

Definition at line 35 of file rsa.h.

Referenced by rsa\_pkcs1\_sign(), and rsa\_pkcs1\_verify().

**8.99.3 Function Documentation****8.99.3.1 int rsa\_check\_privkey (rsa\_context \* ctx)**

Return 0 if the private key is valid, or ERR\_RSA\_KEY\_CHECK\_FAILED.

Definition at line 231 of file dsa.c.

References CHK, rsa\_context::E, ERR\_RSA\_KEY\_CHK\_FAILED, mpi\_cmp\_int(), mpi\_cmp\_mpi(), mpi\_free(), mpi\_gcd(), mpi\_init(), mpi\_mul\_mpi(), mpi\_sub\_int(), rsa\_context::N, rsa\_context::P, and rsa\_context::Q.

Referenced by x509\_parse\_key().

**8.99.3.2 int rsa\_check\_pubkey (rsa\_context \* ctx)**

Return 0 if the public key is valid, or ERR\_RSA\_KEY\_CHECK\_FAILED.

Definition at line 211 of file dsa.c.

References rsa\_context::E, ERR\_RSA\_KEY\_CHK\_FAILED, mpi\_msb(), rsa\_context::N, and mpi::p.

Referenced by x509\_add\_certs().

**8.99.3.3 void rsa\_free (rsa\_context \* ctx)**

Free the components of an RSA key.

Definition at line 474 of file dsa.c.

References rsa\_context::D, rsa\_context::DP, rsa\_context::DQ, rsa\_context::E, mpi\_free(), rsa\_context::N, rsa\_context::P, rsa\_context::Q, rsa\_context::QP, rsa\_context::RN, rsa\_context::RP, and rsa\_context::RQ.

Referenced by main(), rsa\_gen\_key(), rsa\_read\_private(), rsa\_read\_public(), x509\_free\_cert(), and x509\_parse\_key().

**8.99.3.4 int rsa\_gen\_key (rsa\_context \* ctx, int nbits, int exponent, int(\*) (void \*) rng\_f, void \* rng\_d)**

Generate an RSA keypair.

**Parameters:**

*ctx* RSA context to be initialized

*nbits* size of the public key in bits  
*exponent* public exponent (e.g., 65537)  
*rng\_f* points to the RNG function  
*rng\_d* points to the RNG data

**Returns:**

0 if successful, or an ERR\_RSA\_XXX error code

Definition at line 44 of file dsa.c.

References CHK, rsa\_context::D, rsa\_context::DP, rsa\_context::DQ, rsa\_context::E, ERR\_RSA\_BAD\_INPUT\_DATA, ERR\_RSA\_KEY\_GEN\_FAILED, rsa\_context::len, mpi\_cmp\_int(), mpi\_cmp\_mpi(), mpi\_free(), mpi\_gcd(), mpi\_gen\_prime(), mpi\_init(), mpi\_inv\_mod(), mpi\_lset(), mpi\_mod\_mpi(), mpi\_msb(), mpi\_mul\_mpi(), mpi\_sub\_int(), mpi\_swap(), rsa\_context::N, rsa\_context::P, rsa\_context::Q, rsa\_context::QP, and rsa\_free().

Referenced by main().

### 8.99.3.5 `int rsa_pkcs1_decrypt(rsa_context * ctx, unsigned char * input, int ilen, unsigned char * output, int * olen)`

Do a private RSA, removes the PKCS#1 v1.5 padding.

**Parameters:**

*ctx* RSA context  
*input* buffer holding the encrypted data  
*ilen* must be the same as the modulus size  
*output* buffer that will hold the plaintext  
*olen* size of output buffer, will be updated to contain the length of the plaintext

**Returns:**

0 if successful, or an ERR\_RSA\_XXX error code

Definition at line 290 of file dsa.c.

References ERR\_RSA\_BAD\_INPUT\_DATA, ERR\_RSA\_INVALID\_PADDING, int, rsa\_context::len, RSA\_CRYPT, and rsa\_private().

Referenced by ssl\_parse\_client\_key\_exchange().

### 8.99.3.6 `int rsa_pkcs1_encrypt(rsa_context * ctx, unsigned char * input, int ilen, unsigned char * output, int olen)`

Add the PKCS#1 v1.5 padding and do a public RSA.

**Parameters:**

*ctx* RSA context  
*input* buffer holding the data to be encrypted  
*ilen* length of the plaintext; cannot be longer than the modulus, minus 3+8 for padding

*output* buffer that will hold the ciphertext

*olen* must be the same as the modulus size (for example, 128 if RSA-1024 is used)

#### Returns:

0 if successful, or an ERR\_RSA\_XXX error code

Definition at line 260 of file dsa.c.

References ERR\_RSA\_BAD\_INPUT\_DATA, rsa\_context::len, RSA\_CRYPT, and rsa\_public().

Referenced by ssl\_write\_client\_key\_exchange().

#### 8.99.3.7 int rsa\_pkcs1\_sign (rsa\_context \* ctx, int alg\_id, unsigned char \* hash, int hashlen, unsigned char \* sig, int siglen)

Perform a private RSA to sign a message digest.

#### Parameters:

*ctx* RSA context

*alg\_id* RSA\_RAW, RSA\_MD2/4/5 or RSA\_SHA1

*hash* buffer holding the message digest

*hashlen* message digest length

*sig* buffer that will hold the ciphertext

*siglen* must be the same as the modulus size (for example, 128 if RSA-1024 is used)

#### Returns:

0 if the signing operation was successful, or an ERR\_RSA\_XXX error code

Definition at line 328 of file dsa.c.

References ASN1\_HASH\_MDX, ASN1\_HASH\_SHA1, ERR\_RSA\_BAD\_INPUT\_DATA, rsa\_context::len, RSA\_MD2, RSA\_MD4, RSA\_MD5, rsa\_private(), RSA\_RAW, RSA\_SHA1, and RSA\_SIGN.

Referenced by main(), ssl\_write\_certificate\_verify(), and ssl\_write\_server\_key\_exchange().

#### 8.99.3.8 int rsa\_pkcs1\_verify (rsa\_context \* ctx, int alg\_id, unsigned char \* hash, int hashlen, unsigned char \* sig, int siglen)

Perform a public RSA and check the message digest.

#### Parameters:

*ctx* points to an RSA public key

*alg\_id* RSA\_RAW, RSA\_MD2/4/5 or RSA\_SHA1

*hash* buffer holding the message digest

*hashlen* message digest length

*sig* buffer holding the ciphertext

*siglen* must be the same as the modulus size

**Returns:**

0 if the verify operation was successful, or an ERR\_RSA\_XXX error code

Definition at line 404 of file dsa.c.

References ASN1\_HASH\_MD5, ASN1\_HASH\_SHA1, ERR\_RSA\_BAD\_INPUT\_DATA, ERR\_RSA\_INVALID\_PADDING, ERR\_RSA\_VERIFY\_FAILED, int, rsa\_context::len, RSA\_MD2, RSA\_MD4, RSA\_MD5, rsa\_public(), RSA\_RAW, RSA\_SHA1, and RSA\_SIGN.

Referenced by main(), ssl\_parse\_certificate\_verify(), ssl\_parse\_server\_key\_exchange(), and x509\_verify\_cert().

### 8.99.3.9 int rsa\_private (rsa\_context \* ctx, unsigned char \* input, int ilen, unsigned char \* output, int olen)

Perform an RSA private key operation.

**Returns:**

0 if successful, or an ERR\_RSA\_XXX error code

**Note:**

This function does not take care of message padding: both ilen and olen must be equal to the modulus size (ctx->len). Also, be sure to set input[0] = 0.

Definition at line 150 of file dsa.c.

References CHK, rsa\_context::D, rsa\_context::DP, rsa\_context::DQ, ERR\_RSA\_BAD\_INPUT\_DATA, ERR\_RSA\_PRIVATE\_FAILED, rsa\_context::len, mpi\_add\_mpi(), mpi\_cmp\_mpi(), mpi\_exp\_mod(), mpi\_free(), mpi\_init(), mpi\_mod\_mpi(), mpi\_mul\_mpi(), mpi\_read\_binary(), mpi\_sub\_mpi(), mpi\_write\_binary(), rsa\_context::N, rsa\_context::P, rsa\_context::Q, rsa\_context::QP, rsa\_context::RN, rsa\_context::RP, and rsa\_context::RQ.

Referenced by main(), rsa\_pkcs1\_decrypt(), and rsa\_pkcs1\_sign().

### 8.99.3.10 int rsa\_public (rsa\_context \* ctx, unsigned char \* input, int ilen, unsigned char \* output, int olen)

Perform an RSA public key operation.

**Returns:**

0 if successful, or an ERR\_RSA\_XXX error code

**Note:**

This function does not take care of message padding: both ilen and olen must be equal to the modulus size (ctx->len). Also, be sure to set input[0] = 0.

Definition at line 114 of file dsa.c.

References CHK, rsa\_context::E, ERR\_RSA\_BAD\_INPUT\_DATA, ERR\_RSA\_PUBLIC\_FAILED, rsa\_context::len, mpi\_cmp\_mpi(), mpi\_exp\_mod(), mpi\_free(), mpi\_init(), mpi\_read\_binary(), mpi\_write\_binary(), rsa\_context::N, and rsa\_context::RN.

Referenced by main(), rsa\_pkcs1\_encrypt(), and rsa\_pkcs1\_verify().

### 8.99.3.11 `int rsa_read_private (rsa_context * ctx, FILE * f)`

Read the private key from a file.

#### Parameters:

*ctx* RSA context to be initialized

*f* Handle of the source file

#### Returns:

0 if successful, or an ERR\_RSA\_XXX error code

Definition at line 139 of file rsa.c.

References CHK, rsa\_context::D, rsa\_context::DP, rsa\_context::DQ, rsa\_context::E, ERR\_RSA\_KEY\_RD\_FAILED, rsa\_context::len, mpi\_msb(), mpi\_read\_file(), rsa\_context::N, rsa\_context::P, rsa\_context::Q, rsa\_context::QP, and rsa\_free().

Referenced by main().

### 8.99.3.12 `int rsa_read_public (rsa_context * ctx, FILE * f)`

Read the public key from a file.

#### Parameters:

*ctx* RSA context to be initialized

*f* Handle of the source file

#### Returns:

0 if successful, or an ERR\_RSA\_XXX error code

Definition at line 114 of file rsa.c.

References CHK, rsa\_context::E, ERR\_RSA\_KEY\_RD\_FAILED, rsa\_context::len, mpi\_msb(), mpi\_read\_file(), rsa\_context::N, and rsa\_free().

Referenced by main().

### 8.99.3.13 `int rsa_self_test (int verbose)`

Checkup routine.

#### Returns:

0 if successful, or 1 if the test failed

Definition at line 588 of file dsa.c.

Referenced by main().

**8.99.3.14 int rsa\_write\_private (rsa\_context \* *ctx*, FILE \* *f*)**

Write the private key into a file.

**Parameters:**

*ctx* RSA context holding the key  
*f* Handle of the destination file

**Returns:**

0 if successful, or an ERR\_RSA\_XXX error code

Definition at line 188 of file rsa.c.

References CHK, rsa\_context::D, rsa\_context::DP, rsa\_context::DQ, rsa\_context::E, ERR\_RSA\_KEY\_WR\_FAILED, mpi\_write\_file(), rsa\_context::N, rsa\_context::P, rsa\_context::Q, and rsa\_context::QP.

Referenced by main().

**8.99.3.15 int rsa\_write\_public (rsa\_context \* *ctx*, FILE \* *f*)**

Write the public key into a file.

**Parameters:**

*ctx* RSA context holding the key  
*f* Handle of the destination file

**Returns:**

0 if successful, or an ERR\_RSA\_XXX error code

Definition at line 170 of file rsa.c.

References CHK, rsa\_context::E, ERR\_RSA\_KEY\_WR\_FAILED, mpi\_write\_file(), and rsa\_context::N.

Referenced by main().

## 8.100 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/sha1.h File Reference

### Data Structures

- struct [sha1\\_context](#)  
*SHA-1 context structure.*

### Functions

- void [sha1\\_starts](#) ([sha1\\_context](#) \*ctx)  
*SHA-1 context setup.*
- void [sha1\\_update](#) ([sha1\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*SHA-1 process buffer.*
- void [sha1\\_finish](#) ([sha1\\_context](#) \*ctx, unsigned char \*output)  
*SHA-1 final digest.*
- void [sha1](#) (unsigned char \*input, [int](#) ilen, unsigned char \*output)  
*Output = SHA-1( input buffer ).*
- [int](#) [sha1\\_file](#) (char \*path, unsigned char \*output)  
*Output = SHA-1( file contents ).*
- void [sha1\\_hmac\\_starts](#) ([sha1\\_context](#) \*ctx, unsigned char \*key, [int](#) keylen)  
*SHA-1 HMAC context setup.*
- void [sha1\\_hmac\\_update](#) ([sha1\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*SHA-1 HMAC process buffer.*
- void [sha1\\_hmac\\_finish](#) ([sha1\\_context](#) \*ctx, unsigned char \*output)  
*SHA-1 HMAC final digest.*
- void [sha1\\_hmac](#) (unsigned char \*key, [int](#) keylen, unsigned char \*input, [int](#) ilen, unsigned char \*output)  
*Output = HMAC-SHA-1( hmac key, input buffer ).*
- [int](#) [sha1\\_self\\_test](#) ([int](#) verbose)  
*Checkup routine.*

### 8.100.1 Detailed Description

Definition in file [sha1.h](#).

## 8.100.2 Function Documentation

### 8.100.2.1 void sha1 (unsigned char \* *input*, int *ilen*, unsigned char \* *output*)

Output = SHA-1( input buffer ).

#### Parameters:

*input* buffer holding the data  
*ilen* length of the input data  
*output* SHA-1 checksum result

Definition at line 316 of file sha1.c.

References sha1\_finish(), sha1\_starts(), and sha1\_update().

Referenced by main(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_get\_session(), ssl\_mac\_sha1(), ssl\_parse\_server\_key\_exchange(), ssl\_set\_session(), ssl\_write\_server\_key\_exchange(), and x509\_hash().

### 8.100.2.2 int sha1\_file (char \* *path*, unsigned char \* *output*)

Output = SHA-1( file contents ).

#### Parameters:

*path* input file name  
*output* SHA-1 checksum result

#### Returns:

0 if successful, 1 if fopen failed, or 2 if fread failed

Definition at line 331 of file sha1.c.

References f, sha1\_finish(), sha1\_starts(), and sha1\_update().

Referenced by main(), and sha1\_wrapper().

### 8.100.2.3 void sha1\_finish (sha1\_context \* *ctx*, unsigned char \* *output*)

SHA-1 final digest.

#### Parameters:

*ctx* SHA-1 context  
*output* SHA-1 checksum result

Definition at line 287 of file sha1.c.

References PUT\_UINT32\_BE, sha1\_padding, sha1\_update(), sha1\_context::state, and sha1\_context::total.

Referenced by sha1(), sha1\_file(), sha1\_hmac\_finish(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_sha1(), ssl\_parse\_server\_key\_exchange(), and ssl\_write\_server\_key\_exchange().



**8.100.2.4** `void sha1_hmac (unsigned char * key, int keylen, unsigned char * input, int ilen, unsigned char * output)`

Output = HMAC-SHA-1( hmac key, input buffer ).

**Parameters:**

*key* HMAC secret key

*keylen* length of the HMAC key

*input* buffer holding the data

*ilen* length of the input data

*output* HMAC-SHA-1 result

Definition at line 411 of file sha1.c.

References sha1\_hmac\_finish(), sha1\_hmac\_starts(), and sha1\_hmac\_update().

Referenced by ssl\_decrypt\_buf(), ssl\_encrypt\_buf(), and tls1\_prf().

**8.100.2.5** `void sha1_hmac_finish (sha1_context * ctx, unsigned char * output)`

SHA-1 HMAC final digest.

**Parameters:**

*ctx* HMAC context

*output* SHA-1 HMAC checksum result

Definition at line 395 of file sha1.c.

References sha1\_context::opad, sha1\_finish(), sha1\_starts(), and sha1\_update().

Referenced by sha1\_hmac().

**8.100.2.6** `void sha1_hmac_starts (sha1_context * ctx, unsigned char * key, int keylen)`

SHA-1 HMAC context setup.

**Parameters:**

*ctx* HMAC context to be initialized

*key* HMAC secret key

*keylen* length of the HMAC key

Definition at line 363 of file sha1.c.

References sha1\_context::ipad, sha1\_context::opad, sha1\_starts(), and sha1\_update().

Referenced by sha1\_hmac().

**8.100.2.7 void sha1\_hmac\_update (sha1\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

SHA-1 HMAC process buffer.

**Parameters:**

*ctx* HMAC context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 386 of file sha1.c.

References sha1\_update().

Referenced by sha1\_hmac().

**8.100.2.8 int sha1\_self\_test (int *verbose*)**

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 492 of file sha1.c.

Referenced by main().

**8.100.2.9 void sha1\_starts (sha1\_context \* *ctx*)**

SHA-1 context setup.

**Parameters:**

*ctx* context to be initialized

Definition at line 64 of file sha1.c.

References sha1\_context::state, and sha1\_context::total.

Referenced by sha1(), sha1\_file(), sha1\_hmac\_finish(), sha1\_hmac\_starts(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_sha1(), ssl\_parse\_client\_hello(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_client\_hello(), and ssl\_write\_server\_key\_exchange().

**8.100.2.10 void sha1\_update (sha1\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

SHA-1 process buffer.

**Parameters:**

*ctx* SHA-1 context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 235 of file sha1.c.

References sha1\_context::buffer, sha1\_process(), and sha1\_context::total.

Referenced by sha1(), sha1\_file(), sha1\_finish(), sha1\_hmac\_finish(), sha1\_hmac\_starts(), sha1\_hmac\_update(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_sha1(), ssl\_parse\_client\_hello(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_record(), and ssl\_write\_server\_key\_exchange().

## 8.101 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/sha2.h File Reference

### Data Structures

- struct [sha2\\_context](#)  
*SHA-256 context structure.*

### Functions

- void [sha2\\_starts](#) ([sha2\\_context](#) \*ctx, [int](#) is224)  
*SHA-256 context setup.*
- void [sha2\\_update](#) ([sha2\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*SHA-256 process buffer.*
- void [sha2\\_finish](#) ([sha2\\_context](#) \*ctx, unsigned char \*output)  
*SHA-256 final digest.*
- void [sha2](#) (unsigned char \*input, [int](#) ilen, unsigned char \*output, [int](#) is224)  
*Output = SHA-256( input buffer ).*
- [int](#) [sha2\\_file](#) (char \*path, unsigned char \*output, [int](#) is224)  
*Output = SHA-256( file contents ).*
- void [sha2\\_hmac\\_starts](#) ([sha2\\_context](#) \*ctx, [int](#) is224, unsigned char \*key, [int](#) keylen)  
*SHA-256 HMAC context setup.*
- void [sha2\\_hmac\\_update](#) ([sha2\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*SHA-256 HMAC process buffer.*
- void [sha2\\_hmac\\_finish](#) ([sha2\\_context](#) \*ctx, unsigned char \*output)  
*SHA-256 HMAC final digest.*
- void [sha2\\_hmac](#) (unsigned char \*key, [int](#) keylen, unsigned char \*input, [int](#) ilen, unsigned char \*output, [int](#) is224)  
*Output = HMAC-SHA-256( hmac key, input buffer ).*
- [int](#) [sha2\\_self\\_test](#) ([int](#) verbose)  
*Checkup routine.*

### 8.101.1 Detailed Description

Definition in file [sha2.h](#).

## 8.101.2 Function Documentation

### 8.101.2.1 void sha2 (unsigned char \* *input*, int *ilen*, unsigned char \* *output*, int *is224*)

Output = SHA-256( input buffer ).

#### Parameters:

*input* buffer holding the data  
*ilen* length of the input data  
*output* SHA-224/256 checksum result  
*is224* 0 = use SHA256, 1 = use SHA224

Definition at line 317 of file sha2.c.

References sha2\_finish(), sha2\_starts(), and sha2\_update().

Referenced by main().

### 8.101.2.2 int sha2\_file (char \* *path*, unsigned char \* *output*, int *is224*)

Output = SHA-256( file contents ).

#### Parameters:

*path* input file name  
*output* SHA-224/256 checksum result  
*is224* 0 = use SHA256, 1 = use SHA224

#### Returns:

0 if successful, 1 if fopen failed, or 2 if fread failed

Definition at line 332 of file sha2.c.

References f, sha2\_finish(), sha2\_starts(), and sha2\_update().

Referenced by sha2\_wrapper().

### 8.101.2.3 void sha2\_finish (sha2\_context \* *ctx*, unsigned char \* *output*)

SHA-256 final digest.

#### Parameters:

*ctx* SHA-256 context  
*output* SHA-224/256 checksum result

Definition at line 283 of file sha2.c.

References sha2\_context::is224, PUT\_UINT32\_BE, sha2\_padding, sha2\_update(), sha2\_context::state, and sha2\_context::total.

Referenced by main(), sha2(), sha2\_file(), and sha2\_hmac\_finish().

**8.101.2.4 void sha2\_hmac (unsigned char \* *key*, int *keylen*, unsigned char \* *input*, int *ilen*, unsigned char \* *output*, int *is224*)**

Output = HMAC-SHA-256( hmac key, input buffer ).

**Parameters:**

*key* HMAC secret key  
*keylen* length of the HMAC key  
*input* buffer holding the data  
*ilen* length of the input data  
*output* HMAC-SHA-224/256 result  
*is224* 0 = use SHA256, 1 = use SHA224

Definition at line 416 of file sha2.c.

References sha2\_hmac\_finish(), sha2\_hmac\_starts(), and sha2\_hmac\_update().

**8.101.2.5 void sha2\_hmac\_finish (sha2\_context \* *ctx*, unsigned char \* *output*)**

SHA-256 HMAC final digest.

**Parameters:**

*ctx* HMAC context  
*output* SHA-224/256 HMAC checksum result

Definition at line 396 of file sha2.c.

References sha2\_context::is224, sha2\_context::opad, sha2\_finish(), sha2\_starts(), and sha2\_update().

Referenced by main(), and sha2\_hmac().

**8.101.2.6 void sha2\_hmac\_starts (sha2\_context \* *ctx*, int *is224*, unsigned char \* *key*, int *keylen*)**

SHA-256 HMAC context setup.

**Parameters:**

*ctx* HMAC context to be initialized  
*is224* 0 = use SHA256, 1 = use SHA224  
*key* HMAC secret key  
*keylen* length of the HMAC key

Definition at line 364 of file sha2.c.

References sha2\_context::ipad, sha2\_context::opad, sha2\_starts(), and sha2\_update().

Referenced by main(), and sha2\_hmac().

**8.101.2.7 void sha2\_hmac\_update (sha2\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

SHA-256 HMAC process buffer.

**Parameters:**

*ctx* HMAC context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 387 of file sha2.c.

References sha2\_update().

Referenced by main(), and sha2\_hmac().

**8.101.2.8 int sha2\_self\_test (int *verbose*)**

Checksum routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 527 of file sha2.c.

Referenced by main().

**8.101.2.9 void sha2\_starts (sha2\_context \* *ctx*, int *is224*)**

SHA-256 context setup.

**Parameters:**

*ctx* context to be initialized

*is224* 0 = use SHA256, 1 = use SHA224

Definition at line 64 of file sha2.c.

References sha2\_context::is224, sha2\_context::state, and sha2\_context::total.

Referenced by main(), sha2(), sha2\_file(), sha2\_hmac\_finish(), and sha2\_hmac\_starts().

**8.101.2.10 void sha2\_update (sha2\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

SHA-256 process buffer.

**Parameters:**

*ctx* SHA-256 context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 231 of file sha2.c.

References sha2\_context::buffer, sha2\_process(), and sha2\_context::total.

Referenced by main(), sha2(), sha2\_file(), sha2\_finish(), sha2\_hmac\_finish(), sha2\_hmac\_starts(), and sha2\_hmac\_update().



## 8.102 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/sha4.h File Reference

### Data Structures

- struct [sha4\\_context](#)  
*SHA-512 context structure.*

### Defines

- #define [UL64](#)(x) x##ULL
- #define [uint64](#) unsigned long long

### Functions

- void [sha4\\_starts](#) ([sha4\\_context](#) \*ctx, [int](#) is384)  
*SHA-512 context setup.*
- void [sha4\\_update](#) ([sha4\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*SHA-512 process buffer.*
- void [sha4\\_finish](#) ([sha4\\_context](#) \*ctx, unsigned char \*output)  
*SHA-512 final digest.*
- void [sha4](#) (unsigned char \*input, [int](#) ilen, unsigned char \*output, [int](#) is384)  
*Output = SHA-512( input buffer ).*
- [int](#) [sha4\\_file](#) (char \*path, unsigned char \*output, [int](#) is384)  
*Output = SHA-512( file contents ).*
- void [sha4\\_hmac\\_starts](#) ([sha4\\_context](#) \*ctx, [int](#) is384, unsigned char \*key, [int](#) keylen)  
*SHA-512 HMAC context setup.*
- void [sha4\\_hmac\\_update](#) ([sha4\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*SHA-512 HMAC process buffer.*
- void [sha4\\_hmac\\_finish](#) ([sha4\\_context](#) \*ctx, unsigned char \*output)  
*SHA-512 HMAC final digest.*
- void [sha4\\_hmac](#) (unsigned char \*key, [int](#) keylen, unsigned char \*input, [int](#) ilen, unsigned char \*output, [int](#) is384)  
*Output = HMAC-SHA-512( hmac key, input buffer ).*
- [int](#) [sha4\\_self\\_test](#) ([int](#) verbose)  
*Checkup routine.*

### 8.102.1 Detailed Description

Definition in file [sha4.h](#).

### 8.102.2 Define Documentation

#### 8.102.2.1 #define uint64 unsigned long long

Definition at line 19 of file sha4.h.

Referenced by sha4\_finish(), sha4\_process(), and sha4\_update().

#### 8.102.2.2 #define UL64(x) x##ULL

Definition at line 18 of file sha4.h.

Referenced by sha4\_starts().

### 8.102.3 Function Documentation

#### 8.102.3.1 void sha4 (unsigned char \* *input*, int *ilen*, unsigned char \* *output*, int *is384*)

Output = SHA-512( input buffer ).

##### Parameters:

*input* buffer holding the data  
*ilen* length of the input data  
*output* SHA-384/512 checksum result  
*is384* 0 = use SHA512, 1 = use SHA384

Definition at line 315 of file sha4.c.

References sha4\_finish(), sha4\_starts(), and sha4\_update().

#### 8.102.3.2 int sha4\_file (char \* *path*, unsigned char \* *output*, int *is384*)

Output = SHA-512( file contents ).

##### Parameters:

*path* input file name  
*output* SHA-384/512 checksum result  
*is384* 0 = use SHA512, 1 = use SHA384

##### Returns:

0 if successful, 1 if fopen failed, or 2 if fread failed

Definition at line 330 of file sha4.c.

References f, sha4\_finish(), sha4\_starts(), and sha4\_update().

### 8.102.3.3 void sha4\_finish (sha4\_context \* ctx, unsigned char \* output)

SHA-512 final digest.

#### Parameters:

*ctx* SHA-512 context

*output* SHA-384/512 checksum result

Definition at line 279 of file sha4.c.

References int, sha4\_context::is384, PUT\_UINT64\_BE, sha4\_padding, sha4\_update(), sha4\_context::state, sha4\_context::total, and uint64.

Referenced by sha4(), sha4\_file(), and sha4\_hmac\_finish().

### 8.102.3.4 void sha4\_hmac (unsigned char \* key, int keylen, unsigned char \* input, int ilen, unsigned char \* output, int is384)

Output = HMAC-SHA-512( hmac key, input buffer ).

#### Parameters:

*key* HMAC secret key

*keylen* length of the HMAC key

*input* buffer holding the data

*ilen* length of the input data

*output* HMAC-SHA-384/512 result

*is384* 0 = use SHA512, 1 = use SHA384

Definition at line 414 of file sha4.c.

References sha4\_hmac\_finish(), sha4\_hmac\_starts(), and sha4\_hmac\_update().

### 8.102.3.5 void sha4\_hmac\_finish (sha4\_context \* ctx, unsigned char \* output)

SHA-512 HMAC final digest.

#### Parameters:

*ctx* HMAC context

*output* SHA-384/512 HMAC checksum result

Definition at line 394 of file sha4.c.

References sha4\_context::is384, sha4\_context::opad, sha4\_finish(), sha4\_starts(), and sha4\_update().

Referenced by sha4\_hmac().

**8.102.3.6 void sha4\_hmac\_starts (sha4\_context \* ctx, int is384, unsigned char \* key, int keylen)**

SHA-512 HMAC context setup.

**Parameters:**

*ctx* HMAC context to be initialized  
*is384* 0 = use SHA512, 1 = use SHA384  
*key* HMAC secret key  
*keylen* length of the HMAC key

Definition at line 362 of file sha4.c.

References sha4\_context::ipad, sha4\_context::opad, sha4\_starts(), and sha4\_update().

Referenced by sha4\_hmac().

**8.102.3.7 void sha4\_hmac\_update (sha4\_context \* ctx, unsigned char \* input, int ilen)**

SHA-512 HMAC process buffer.

**Parameters:**

*ctx* HMAC context  
*input* buffer holding the data  
*ilen* length of the input data

Definition at line 385 of file sha4.c.

References sha4\_update().

Referenced by sha4\_hmac().

**8.102.3.8 int sha4\_self\_test (int verbose)**

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 544 of file sha4.c.

**8.102.3.9 void sha4\_starts (sha4\_context \* ctx, int is384)**

SHA-512 context setup.

**Parameters:**

*ctx* context to be initialized  
*is384* 0 = use SHA512, 1 = use SHA384

Definition at line 119 of file sha4.c.

References sha4\_context::is384, sha4\_context::state, sha4\_context::total, and UL64.

Referenced by sha4(), sha4\_file(), sha4\_hmac\_finish(), and sha4\_hmac\_starts().

**8.102.3.10 void sha4\_update (sha4\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

SHA-512 process buffer.

**Parameters:**

*ctx* SHA-512 context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 224 of file sha4.c.

References sha4\_context::buffer, int, sha4\_process(), sha4\_context::total, and uint64.

Referenced by sha4(), sha4\_file(), sha4\_finish(), sha4\_hmac\_finish(), sha4\_hmac\_starts(), and sha4\_hmac\_update().

## 8.103 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/ssl.h File Reference

```
#include "x509.h"
#include "rsa.h"
#include "dhm.h"
#include "md5.h"
#include "sha1.h"
```

### Data Structures

- struct [ssl\\_context](#)

### Defines

- #define [ERR\\_SSL\\_FEATURE\\_UNAVAILABLE](#) 0x1000
- #define [ERR\\_SSL\\_INVALID\\_MAC](#) 0x1800
- #define [ERR\\_SSL\\_INVALID\\_RECORD](#) 0x2000
- #define [ERR\\_SSL\\_INVALID\\_MODULUS\\_SIZE](#) 0x2800
- #define [ERR\\_SSL\\_UNKNOWN\\_CIPHER](#) 0x3000
- #define [ERR\\_SSL\\_NO\\_CIPHER\\_CHOSEN](#) 0x3800
- #define [ERR\\_SSL\\_NO\\_SESSION\\_FOUND](#) 0x4000
- #define [ERR\\_SSL\\_NO\\_CLIENT\\_CERTIFICATE](#) 0x4800
- #define [ERR\\_SSL\\_CERTIFICATE\\_TOO\\_LARGE](#) 0x5000
- #define [ERR\\_SSL\\_CERTIFICATE\\_REQUIRED](#) 0x5800
- #define [ERR\\_SSL\\_PRIVATE\\_KEY\\_REQUIRED](#) 0x6000
- #define [ERR\\_SSL\\_CA\\_CHAIN\\_REQUIRED](#) 0x6800
- #define [ERR\\_SSL\\_UNEXPECTED\\_MESSAGE](#) 0x7000
- #define [ERR\\_SSL\\_FATAL\\_ALERT\\_MESSAGE](#) 0x7800
- #define [ERR\\_SSL\\_PEER\\_VERIFY\\_FAILED](#) 0x8000
- #define [ERR\\_SSL\\_PEER\\_CLOSE\\_NOTIFY](#) 0x8800
- #define [ERR\\_SSL\\_BAD\\_HS\\_CLIENT\\_HELLO](#) 0x9000
- #define [ERR\\_SSL\\_BAD\\_HS\\_SERVER\\_HELLO](#) 0x9800
- #define [ERR\\_SSL\\_BAD\\_HS\\_CERTIFICATE](#) 0xA000
- #define [ERR\\_SSL\\_BAD\\_HS\\_CERTIFICATE\\_REQUEST](#) 0xA800
- #define [ERR\\_SSL\\_BAD\\_HS\\_SERVER\\_KEY\\_EXCHANGE](#) 0xB000
- #define [ERR\\_SSL\\_BAD\\_HS\\_SERVER\\_HELLO\\_DONE](#) 0xB800
- #define [ERR\\_SSL\\_BAD\\_HS\\_CLIENT\\_KEY\\_EXCHANGE](#) 0xC000
- #define [ERR\\_SSL\\_BAD\\_HS\\_CERTIFICATE\\_VERIFY](#) 0xC800
- #define [ERR\\_SSL\\_BAD\\_HS\\_CHANGE\\_CIPHER\\_SPEC](#) 0xD000
- #define [ERR\\_SSL\\_BAD\\_HS\\_FINISHED](#) 0xD800
- #define [SSLV3\\_MAJOR\\_VERSION](#) 3
- #define [SSLV3\\_MINOR\\_VERSION](#) 0
- #define [TLS10\\_MINOR\\_VERSION](#) 1
- #define [TLS11\\_MINOR\\_VERSION](#) 2
- #define [SSL\\_IS\\_CLIENT](#) 0
- #define [SSL\\_IS\\_SERVER](#) 1

- #define [SSL\\_COMPRESS\\_NULL](#) 0
- #define [SSL\\_VERIFY\\_NONE](#) 0
- #define [SSL\\_VERIFY\\_OPTIONAL](#) 1
- #define [SSL\\_VERIFY\\_REQUIRED](#) 2
- #define [SSL\\_SESSION\\_TBL\\_LEN](#) 8192
- #define [SSL\\_MAX\\_CONTENT\\_LEN](#) 16384
- #define [SSL\\_EXPIRATION\\_TIME](#) 86400
- #define [SSL\\_BUFFER\\_LEN](#) (SSL\_MAX\_CONTENT\_LEN + 512)
- #define [SSL3\\_RSA\\_RC4\\_128\\_MD5](#) 4
- #define [SSL3\\_RSA\\_RC4\\_128\\_SHA](#) 5
- #define [SSL3\\_RSA\\_DES\\_168\\_SHA](#) 10
- #define [SSL3\\_EDH\\_RSA\\_DES\\_168\\_SHA](#) 22
- #define [TLS1\\_RSA\\_AES\\_256\\_SHA](#) 53
- #define [TLS1\\_EDH\\_RSA\\_AES\\_256\\_SHA](#) 57
- #define [SSL\\_MSG\\_CHANGE\\_CIPHER\\_SPEC](#) 20
- #define [SSL\\_MSG\\_ALERT](#) 21
- #define [SSL\\_MSG\\_HANDSHAKE](#) 22
- #define [SSL\\_MSG\\_APPLICATION\\_DATA](#) 23
- #define [SSL\\_ALERT\\_CLOSE\\_NOTIFY](#) 0
- #define [SSL\\_ALERT\\_WARNING](#) 1
- #define [SSL\\_ALERT\\_FATAL](#) 2
- #define [SSL\\_ALERT\\_NO\\_CERTIFICATE](#) 41
- #define [SSL\\_HS\\_HELLO\\_REQUEST](#) 0
- #define [SSL\\_HS\\_CLIENT\\_HELLO](#) 1
- #define [SSL\\_HS\\_SERVER\\_HELLO](#) 2
- #define [SSL\\_HS\\_CERTIFICATE](#) 11
- #define [SSL\\_HS\\_SERVER\\_KEY\\_EXCHANGE](#) 12
- #define [SSL\\_HS\\_CERTIFICATE\\_REQUEST](#) 13
- #define [SSL\\_HS\\_SERVER\\_HELLO\\_DONE](#) 14
- #define [SSL\\_HS\\_CERTIFICATE\\_VERIFY](#) 15
- #define [SSL\\_HS\\_CLIENT\\_KEY\\_EXCHANGE](#) 16
- #define [SSL\\_HS\\_FINISHED](#) 20

## Enumerations

- enum [ssl\\_states](#) {  
    [SSL\\_HELLO\\_REQUEST](#), [SSL\\_CLIENT\\_HELLO](#), [SSL\\_SERVER\\_HELLO](#), [SSL\\_SERVER\\_CERTIFICATE](#),  
    [SSL\\_SERVER\\_KEY\\_EXCHANGE](#), [SSL\\_CERTIFICATE\\_REQUEST](#), [SSL\\_SERVER\\_HELLO\\_DONE](#), [SSL\\_CLIENT\\_CERTIFICATE](#),  
    [SSL\\_CLIENT\\_KEY\\_EXCHANGE](#), [SSL\\_CERTIFICATE\\_VERIFY](#), [SSL\\_CLIENT\\_CHANGE\\_CIPHER\\_SPEC](#), [SSL\\_CLIENT\\_FINISHED](#),  
    [SSL\\_SERVER\\_CHANGE\\_CIPHER\\_SPEC](#), [SSL\\_SERVER\\_FINISHED](#), [SSL\\_HANDSHAKE\\_OVER](#) }

## Functions

- `int ssl_client_start (ssl_context *ssl)`
- `int ssl_server_start (ssl_context *ssl)`
- `int ssl_derive_keys (ssl_context *ssl)`
- `int ssl_calc_verify (ssl_context *ssl, unsigned char hash[36])`
- `int ssl_read_record (ssl_context *ssl, int do_crypt)`
- `int ssl_write_record (ssl_context *ssl, int do_crypt)`
- `int ssl_flush_output (ssl_context *ssl)`
- `int ssl_write_certificate (ssl_context *ssl)`
- `int ssl_parse_certificate (ssl_context *ssl)`
- `int ssl_write_change_cipher_spec (ssl_context *ssl)`
- `int ssl_parse_change_cipher_spec (ssl_context *ssl)`
- `int ssl_write_finished (ssl_context *ssl)`
- `int ssl_parse_finished (ssl_context *ssl)`
- `int ssl_init (ssl_context *ssl, int client_resume)`  
*Initialize the SSL context. If client\_resume is not null, the session id and premaster secret are preserved (client-side only).*
- `void ssl_set_endpoint (ssl_context *ssl, int endpoint)`  
*Set the current endpoint type, SSL\_IS\_CLIENT or SSL\_IS\_SERVER.*
- `void ssl_set_authmode (ssl_context *ssl, int authmode)`  
*Set the certificate verification mode.*
- `void ssl_set_rng_func (ssl_context *ssl, int(*rng_f)(void *), void *rng_d)`  
*Set the random number generator function.*
- `void ssl_set_io_files (ssl_context *ssl, int read_fd, int write_fd)`  
*Set the read and write file descriptors.*
- `void ssl_set_ciphlist (ssl_context *ssl, int *ciphers)`  
*Set the list of allowed ciphersuites.*
- `void ssl_set_ca_chain (ssl_context *ssl, x509_cert *ca, char *cn)`  
*Set the CA certificate chain used to verify peer cert, and the peer's expected CommonName (or NULL).*
- `void ssl_set_rsa_cert (ssl_context *ssl, x509_cert *own_cert, rsa_context *own_key)`  
*Set own certificate and private RSA key.*
- `void ssl_set_sidtable (ssl_context *ssl, unsigned char *sidtable)`  
*Set the global session ID table (server-side only).*
- `int ssl_set_dhm_vals (ssl_context *ssl, char *dhm_P, char *dhm_G)`  
*Set the Diffie-Hellman public P and G values, provided as hexadecimal strings (server-side only).*
- `int ssl_get_verify_result (ssl_context *ssl)`  
*Return the result of the certificate verification.*
- `char * ssl_get_cipher_name (ssl_context *ssl)`



*Return the name of the current cipher.*

- [int ssl\\_handshake](#) ([ssl\\_context](#) \*ssl)

*Perform the SSL handshake.*

- [int ssl\\_read](#) ([ssl\\_context](#) \*ssl, unsigned char \*buf, [int](#) \*len)

*Read at most 'len' application data bytes.*

- [int ssl\\_write](#) ([ssl\\_context](#) \*ssl, unsigned char \*buf, [int](#) len)

*Write 'len' application data bytes.*

- [int ssl\\_close\\_notify](#) ([ssl\\_context](#) \*ssl)

*Notify the peer that the connection is being closed.*

- void [ssl\\_free](#) ([ssl\\_context](#) \*ssl)

*Free an SSL context.*

## Variables

- [int ssl\\_default\\_ciphers](#) []

### 8.103.1 Detailed Description

Definition in file [ssl.h](#).

### 8.103.2 Define Documentation

#### 8.103.2.1 **#define ERR\_SSL\_BAD\_HS\_CERTIFICATE 0xA000**

Definition at line 38 of file [ssl.h](#).

#### 8.103.2.2 **#define ERR\_SSL\_BAD\_HS\_CERTIFICATE\_REQUEST 0xA800**

Definition at line 39 of file [ssl.h](#).

#### 8.103.2.3 **#define ERR\_SSL\_BAD\_HS\_CERTIFICATE\_VERIFY 0xC800**

Definition at line 43 of file [ssl.h](#).

Referenced by [ssl\\_parse\\_certificate\\_verify\(\)](#).

#### 8.103.2.4 **#define ERR\_SSL\_BAD\_HS\_CHANGE\_CIPHER\_SPEC 0xD000**

Definition at line 44 of file [ssl.h](#).

**8.103.2.5 #define ERR\_SSL\_BAD\_HS\_CLIENT\_HELLO 0x9000**

Definition at line 36 of file ssl.h.

Referenced by ssl\_parse\_client\_hello().

**8.103.2.6 #define ERR\_SSL\_BAD\_HS\_CLIENT\_KEY\_EXCHANGE 0xC000**

Definition at line 42 of file ssl.h.

Referenced by ssl\_parse\_client\_key\_exchange().

**8.103.2.7 #define ERR\_SSL\_BAD\_HS\_FINISHED 0xD800**

Definition at line 45 of file ssl.h.

**8.103.2.8 #define ERR\_SSL\_BAD\_HS\_SERVER\_HELLO 0x9800**

Definition at line 37 of file ssl.h.

Referenced by ssl\_parse\_server\_hello().

**8.103.2.9 #define ERR\_SSL\_BAD\_HS\_SERVER\_HELLO\_DONE 0xB800**

Definition at line 41 of file ssl.h.

Referenced by ssl\_parse\_server\_hello\_done().

**8.103.2.10 #define ERR\_SSL\_BAD\_HS\_SERVER\_KEY\_EXCHANGE 0xB000**

Definition at line 40 of file ssl.h.

Referenced by ssl\_parse\_server\_key\_exchange().

**8.103.2.11 #define ERR\_SSL\_CA\_CHAIN\_REQUIRED 0x6800**

Definition at line 31 of file ssl.h.

**8.103.2.12 #define ERR\_SSL\_CERTIFICATE\_REQUIRED 0x5800**

Definition at line 29 of file ssl.h.

Referenced by ssl\_parse\_certificate\_request(), and ssl\_write\_certificate().

**8.103.2.13 #define ERR\_SSL\_CERTIFICATE\_TOO\_LARGE 0x5000**

Definition at line 28 of file ssl.h.

Referenced by ssl\_write\_certificate().

**8.103.2.14 #define ERR\_SSL\_FATAL\_ALERT\_MESSAGE 0x7800**

Definition at line 33 of file ssl.h.

**8.103.2.15 #define ERR\_SSL\_FEATURE\_UNAVAILABLE 0x1000**

Definition at line 20 of file ssl.h.

Referenced by ssl\_decrypt\_buf(), ssl\_derive\_keys(), ssl\_encrypt\_buf(), ssl\_handshake(), ssl\_parse\_client\_key\_exchange(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_client\_key\_exchange(), and ssl\_write\_server\_key\_exchange().

**8.103.2.16 #define ERR\_SSL\_INVALID\_MAC 0x1800**

Definition at line 21 of file ssl.h.

Referenced by ssl\_decrypt\_buf().

**8.103.2.17 #define ERR\_SSL\_INVALID\_MODULUS\_SIZE 0x2800**

Definition at line 23 of file ssl.h.

**8.103.2.18 #define ERR\_SSL\_INVALID\_RECORD 0x2000**

Definition at line 22 of file ssl.h.

**8.103.2.19 #define ERR\_SSL\_NO\_CIPHER\_CHOSEN 0x3800**

Definition at line 25 of file ssl.h.

Referenced by ssl\_parse\_client\_hello(), and ssl\_parse\_server\_hello().

**8.103.2.20 #define ERR\_SSL\_NO\_CLIENT\_CERTIFICATE 0x4800**

Definition at line 27 of file ssl.h.

**8.103.2.21 #define ERR\_SSL\_NO\_SESSION\_FOUND 0x4000**

Definition at line 26 of file ssl.h.

Referenced by ssl\_get\_session().

**8.103.2.22 #define ERR\_SSL\_PEER\_CLOSE\_NOTIFY 0x8800**

Definition at line 35 of file ssl.h.

Referenced by main().

**8.103.2.23 #define ERR\_SSL\_PEER\_VERIFY\_FAILED 0x8000**

Definition at line 34 of file ssl.h.

**8.103.2.24 #define ERR\_SSL\_PRIVATE\_KEY\_REQUIRED 0x6000**

Definition at line 30 of file ssl.h.

Referenced by ssl\_parse\_certificate\_request().

**8.103.2.25 #define ERR\_SSL\_UNEXPECTED\_MESSAGE 0x7000**

Definition at line 32 of file ssl.h.

Referenced by ssl\_parse\_certificate\_request(), ssl\_parse\_server\_hello(), ssl\_parse\_server\_hello\_done(), ssl\_parse\_server\_key\_exchange(), and ssl\_read().

**8.103.2.26 #define ERR\_SSL\_UNKNOWN\_CIPHER 0x3000**

Definition at line 24 of file ssl.h.

**8.103.2.27 #define SSL3\_EDH\_RSA\_DES\_168\_SHA 22**

Definition at line 79 of file ssl.h.

Referenced by ssl\_derive\_keys(), ssl\_parse\_client\_key\_exchange(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_client\_key\_exchange(), and ssl\_write\_server\_key\_exchange().

**8.103.2.28 #define SSL3\_RSA\_DES\_168\_SHA 10**

Definition at line 78 of file ssl.h.

Referenced by ssl\_derive\_keys().

**8.103.2.29 #define SSL3\_RSA\_RC4\_128\_MD5 4**

Definition at line 76 of file ssl.h.

Referenced by ssl\_derive\_keys().

**8.103.2.30 #define SSL3\_RSA\_RC4\_128\_SHA 5**

Definition at line 77 of file ssl.h.

Referenced by ssl\_derive\_keys().

**8.103.2.31 #define SSL\_ALERT\_CLOSE\_NOTIFY 0**

Definition at line 93 of file ssl.h.

Referenced by ssl\_close\_notify().

**8.103.2.32 #define SSL\_ALERT\_FATAL 2**

Definition at line 95 of file ssl.h.

**8.103.2.33 #define SSL\_ALERT\_NO\_CERTIFICATE 41**

Definition at line 96 of file ssl.h.

Referenced by ssl\_write\_certificate().

**8.103.2.34 #define SSL\_ALERT\_WARNING 1**

Definition at line 94 of file ssl.h.

Referenced by ssl\_close\_notify(), and ssl\_write\_certificate().

**8.103.2.35 #define SSL\_BUFFER\_LEN (SSL\_MAX\_CONTENT\_LEN + 512)**

Definition at line 71 of file ssl.h.

Referenced by ssl\_free(), and ssl\_init().

**8.103.2.36 #define SSL\_COMPRESS\_NULL 0**

Definition at line 57 of file ssl.h.

Referenced by ssl\_parse\_server\_hello(), ssl\_write\_client\_hello(), and ssl\_write\_server\_hello().

**8.103.2.37 #define SSL\_EXPIRATION\_TIME 86400**

Definition at line 65 of file ssl.h.

Referenced by ssl\_get\_session().

**8.103.2.38 #define SSL\_HS\_CERTIFICATE 11**

Definition at line 101 of file ssl.h.

Referenced by ssl\_write\_certificate().

**8.103.2.39 #define SSL\_HS\_CERTIFICATE\_REQUEST 13**

Definition at line 103 of file ssl.h.

Referenced by ssl\_parse\_certificate\_request(), and ssl\_write\_certificate\_request().

**8.103.2.40 #define SSL\_HS\_CERTIFICATE\_VERIFY 15**

Definition at line 105 of file ssl.h.

Referenced by ssl\_parse\_certificate\_verify(), and ssl\_write\_certificate\_verify().

**8.103.2.41 #define SSL\_HS\_CLIENT\_HELLO 1**

Definition at line 99 of file ssl.h.

Referenced by ssl\_parse\_client\_hello(), and ssl\_write\_client\_hello().

**8.103.2.42 #define SSL\_HS\_CLIENT\_KEY\_EXCHANGE 16**

Definition at line 106 of file ssl.h.

Referenced by ssl\_parse\_client\_key\_exchange(), and ssl\_write\_client\_key\_exchange().

**8.103.2.43 #define SSL\_HS\_FINISHED 20**

Definition at line 107 of file ssl.h.

**8.103.2.44 #define SSL\_HS\_HELLO\_REQUEST 0**

Definition at line 98 of file ssl.h.

**8.103.2.45 #define SSL\_HS\_SERVER\_HELLO 2**

Definition at line 100 of file ssl.h.

Referenced by ssl\_parse\_server\_hello(), and ssl\_write\_server\_hello().

**8.103.2.46 #define SSL\_HS\_SERVER\_HELLO\_DONE 14**

Definition at line 104 of file ssl.h.

Referenced by ssl\_parse\_server\_hello\_done(), and ssl\_write\_server\_hello\_done().

**8.103.2.47 #define SSL\_HS\_SERVER\_KEY\_EXCHANGE 12**

Definition at line 102 of file ssl.h.

Referenced by ssl\_parse\_server\_key\_exchange(), and ssl\_write\_server\_key\_exchange().

**8.103.2.48 #define SSL\_IS\_CLIENT 0**

Definition at line 55 of file ssl.h.

Referenced by main(), ssl\_derive\_keys(), ssl\_handshake(), and ssl\_write\_certificate().

**8.103.2.49 #define SSL\_IS\_SERVER 1**

Definition at line 56 of file ssl.h.

Referenced by main(), and ssl\_handshake().

**8.103.2.50 #define SSL\_MAX\_CONTENT\_LEN 16384**

Definition at line 64 of file ssl.h.

Referenced by ssl\_write(), and ssl\_write\_certificate().

**8.103.2.51 #define SSL\_MSG\_ALERT 21**

Definition at line 89 of file ssl.h.

Referenced by ssl\_close\_notify(), and ssl\_write\_certificate().

**8.103.2.52 #define SSL\_MSG\_APPLICATION\_DATA 23**

Definition at line 91 of file ssl.h.

Referenced by ssl\_read(), and ssl\_write().

**8.103.2.53 #define SSL\_MSG\_CHANGE\_CIPHER\_SPEC 20**

Definition at line 88 of file ssl.h.

**8.103.2.54 #define SSL\_MSG\_HANDSHAKE 22**

Definition at line 90 of file ssl.h.

Referenced by ssl\_parse\_certificate\_request(), ssl\_parse\_certificate\_verify(), ssl\_parse\_client\_hello(), ssl\_parse\_client\_key\_exchange(), ssl\_parse\_server\_hello(), ssl\_parse\_server\_hello\_done(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_certificate(), ssl\_write\_certificate\_request(), ssl\_write\_certificate\_verify(), ssl\_write\_client\_hello(), ssl\_write\_client\_key\_exchange(), ssl\_write\_record(), ssl\_write\_server\_hello(), ssl\_write\_server\_hello\_done(), and ssl\_write\_server\_key\_exchange().

**8.103.2.55 #define SSL\_SESSION\_TBL\_LEN 8192**

Definition at line 63 of file ssl.h.

Referenced by ssl\_get\_session(), and ssl\_set\_session().

**8.103.2.56 #define SSL\_VERIFY\_NONE 0**

Definition at line 59 of file ssl.h.

Referenced by main(), and ssl\_write\_certificate\_request().

**8.103.2.57 #define SSL\_VERIFY\_OPTIONAL 1**

Definition at line 60 of file ssl.h.

Referenced by main().

**8.103.2.58 #define SSL\_VERIFY\_REQUIRED 2**

Definition at line 61 of file ssl.h.

Referenced by ssl\_parse\_certificate\_verify().

**8.103.2.59 #define SSLV3\_MAJOR\_VERSION 3**

Definition at line 50 of file ssl.h.

Referenced by ssl\_parse\_client\_hello(), ssl\_parse\_server\_hello(), and ssl\_write\_client\_hello().

**8.103.2.60 #define SSLV3\_MINOR\_VERSION 0**

Definition at line 51 of file ssl.h.

Referenced by ssl\_calc\_verify(), ssl\_decrypt\_buf(), ssl\_derive\_keys(), ssl\_encrypt\_buf(), ssl\_parse\_client\_key\_exchange(), ssl\_parse\_server\_hello(), ssl\_write\_certificate(), ssl\_write\_client\_hello(), and ssl\_write\_client\_key\_exchange().

**8.103.2.61 #define TLS10\_MINOR\_VERSION 1**

Definition at line 52 of file ssl.h.

Referenced by ssl\_parse\_client\_hello(), ssl\_parse\_server\_hello(), and ssl\_write\_client\_hello().

**8.103.2.62 #define TLS11\_MINOR\_VERSION 2**

Definition at line 53 of file ssl.h.

**8.103.2.63 #define TLS1\_EDH\_RSA\_AES\_256\_SHA 57**

Definition at line 81 of file ssl.h.

Referenced by ssl\_derive\_keys(), ssl\_parse\_client\_key\_exchange(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_client\_key\_exchange(), and ssl\_write\_server\_key\_exchange().

**8.103.2.64 #define TLS1\_RSA\_AES\_256\_SHA 53**

Definition at line 80 of file ssl.h.

Referenced by ssl\_derive\_keys().

**8.103.3 Enumeration Type Documentation****8.103.3.1 enum ssl\_states**

Enumerator:

*SSL\_HELLO\_REQUEST*

*SSL\_CLIENT\_HELLO*

*SSL\_SERVER\_HELLO*



***SSL\_SERVER\_CERTIFICATE***  
***SSL\_SERVER\_KEY\_EXCHANGE***  
***SSL\_CERTIFICATE\_REQUEST***  
***SSL\_SERVER\_HELLO\_DONE***  
***SSL\_CLIENT\_CERTIFICATE***  
***SSL\_CLIENT\_KEY\_EXCHANGE***  
***SSL\_CERTIFICATE\_VERIFY***  
***SSL\_CLIENT\_CHANGE\_CIPHER\_SPEC***  
***SSL\_CLIENT\_FINISHED***  
***SSL\_SERVER\_CHANGE\_CIPHER\_SPEC***  
***SSL\_SERVER\_FINISHED***  
***SSL\_HANDSHAKE\_OVER***

Definition at line 112 of file ssl.h.

## 8.103.4 Function Documentation

### 8.103.4.1 `int ssl_calc_verify (ssl_context * ssl, unsigned char hash[36])`

Definition at line 329 of file ssl\_tls.c.

References `ssl_context::hs_md5`, `ssl_context::hs_sha1`, `ssl_context::master`, `md5()`, `md5_finish()`, `md5_starts()`, `md5_update()`, `ssl_context::minor_ver`, `sha1()`, `sha1_finish()`, `sha1_starts()`, `sha1_update()`, and `SSLV3_MINOR_VERSION`.

Referenced by `ssl_parse_certificate_verify()`, and `ssl_write_certificate_verify()`.

### 8.103.4.2 `int ssl_client_start (ssl_context * ssl)`

Definition at line 441 of file ssl\_cli.c.

References `SSL_CERTIFICATE_REQUEST`, `SSL_CERTIFICATE_VERIFY`, `SSL_CLIENT_CERTIFICATE`, `SSL_CLIENT_CHANGE_CIPHER_SPEC`, `SSL_CLIENT_FINISHED`, `SSL_CLIENT_HELLO`, `SSL_CLIENT_KEY_EXCHANGE`, `ssl_flush_output()`, `SSL_HELLO_REQUEST`, `ssl_parse_certificate()`, `ssl_parse_certificate_request()`, `ssl_parse_change_cipher_spec()`, `ssl_parse_finished()`, `ssl_parse_server_hello()`, `ssl_parse_server_hello_done()`, `ssl_parse_server_key_exchange()`, `SSL_SERVER_CERTIFICATE`, `SSL_SERVER_CHANGE_CIPHER_SPEC`, `SSL_SERVER_FINISHED`, `SSL_SERVER_HELLO`, `SSL_SERVER_HELLO_DONE`, `SSL_SERVER_KEY_EXCHANGE`, `ssl_write_certificate()`, `ssl_write_certificate_verify()`, `ssl_write_change_cipher_spec()`, `ssl_write_client_hello()`, `ssl_write_client_key_exchange()`, `ssl_write_finished()`, and `ssl_context::state`.

Referenced by `ssl_handshake()`.

### 8.103.4.3 `int ssl_close_notify (ssl_context * ssl)`

Notify the peer that the connection is being closed.

Definition at line 1453 of file ssl\_tls.c.

References `ssl_context::out_msg`, `ssl_context::out_msglen`, `ssl_context::out_msgtype`, `SSL_ALERT_CLOSE_NOTIFY`, `SSL_ALERT_WARNING`, `ssl_flush_output()`, `SSL_HANDSHAKE_OVER`, `SSL_MSG_ALERT`, `ssl_write_record()`, and `ssl_context::state`.

Referenced by `main()`.

#### 8.103.4.4 `int ssl_derive_keys (ssl_context * ssl)`

Definition at line 104 of file `ssl_tls.c`.

References `aes_set_key()`, `arc4_setup()`, `ssl_context::cipher`, `ssl_context::ctx_dec`, `ssl_context::ctx_enc`, `ssl_context::ctxlen`, `des3_set_3keys()`, `ssl_context::endpoint`, `ERR_SSL_FEATURE_UNAVAILABLE`, `ssl_context::iv_dec`, `ssl_context::iv_enc`, `ssl_context::ivlen`, `ssl_context::keylen`, `ssl_context::mac_dec`, `ssl_context::mac_enc`, `ssl_context::maclen`, `ssl_context::master`, `md5()`, `md5_finish()`, `md5_starts()`, `md5_update()`, `ssl_context::minlen`, `ssl_context::minor_ver`, `ssl_context::pmslen`, `ssl_context::premaster`, `ssl_context::randbytes`, `ssl_context::resumed`, `sha1()`, `sha1_finish()`, `sha1_starts()`, `sha1_update()`, `SSL3_EDH_RSA_DES_168_SHA`, `SSL3_RSA_DES_168_SHA`, `SSL3_RSA_RC4_128_MD5`, `SSL3_RSA_RC4_128_SHA`, `SSL_IS_CLIENT`, `SSLV3_MINOR_VERSION`, `TLS1_EDH_RSA_AES_256_SHA`, `tls1_prf()`, and `TLS1_RSA_AES_256_SHA`.

Referenced by `ssl_parse_client_key_exchange()`, `ssl_parse_server_hello()`, `ssl_write_client_key_exchange()`, and `ssl_write_server_hello()`.

#### 8.103.4.5 `int ssl_flush_output (ssl_context * ssl)`

Definition at line 826 of file `ssl_tls.c`.

References `net_send()`, `ssl_context::out_hdr`, `ssl_context::out_left`, `ssl_context::out_msglen`, and `ssl_context::write_fd`.

Referenced by `ssl_client_start()`, `ssl_close_notify()`, and `ssl_server_start()`.

#### 8.103.4.6 `void ssl_free (ssl_context * ssl)`

Free an SSL context.

Definition at line 1476 of file `ssl_tls.c`.

References `ssl_context::ctx_dec`, `ssl_context::ctx_enc`, `ssl_context::ctxlen`, `ssl_context::dhm_ctx`, `dhm_free()`, `ssl_context::in_ctr`, `ssl_context::out_ctr`, `ssl_context::peer_cert`, `SSL_BUFFER_LEN`, and `x509_free_cert()`.

Referenced by `main()`.

#### 8.103.4.7 `char* ssl_get_cipher_name (ssl_context * ssl)`

Return the name of the current cipher.

Definition at line 1311 of file `ssl_tls.c`.

Referenced by `main()`.

#### 8.103.4.8 `int ssl_get_verify_result (ssl_context * ssl)`

Return the result of the certificate verification.

Definition at line 1306 of file `ssl_tls.c`.

References `ssl_context::verify_result`.

Referenced by `main()`.

#### **8.103.4.9 int ssl\_handshake (ssl\_context \* ssl)**

Perform the SSL handshake.

##### **Returns:**

0 if successful, ERR\_NET\_WOULD\_BLOCK (only when the socket is set to non-blocking), or a specific SSL error code.

Definition at line 1372 of file ssl\_tls.c.

References ssl\_context::endpoint, ERR\_SSL\_FEATURE\_UNAVAILABLE, ssl\_client\_start(), SSL\_IS\_CLIENT, SSL\_IS\_SERVER, and ssl\_server\_start().

Referenced by main(), ssl\_read(), and ssl\_write().

#### **8.103.4.10 int ssl\_init (ssl\_context \* ssl, int client\_resume)**

Initialize the SSL context. If client\_resume is not null, the session id and premaster secret are preserved (client-side only).

##### **Returns:**

0 if successful, or 1 if memory allocation failed

Definition at line 1202 of file ssl\_tls.c.

References ssl\_context::in\_ctr, ssl\_context::in\_hdr, ssl\_context::in\_msg, ssl\_context::master, ssl\_context::out\_ctr, ssl\_context::out\_hdr, ssl\_context::out\_msg, ssl\_context::sessid, ssl\_context::sidlen, and SSL\_BUFFER\_LEN.

Referenced by main().

#### **8.103.4.11 int ssl\_parse\_certificate (ssl\_context \* ssl)**

Definition at line 917 of file ssl\_tls.c.

Referenced by ssl\_client\_start(), and ssl\_server\_start().

#### **8.103.4.12 int ssl\_parse\_change\_cipher\_spec (ssl\_context \* ssl)**

Definition at line 1031 of file ssl\_tls.c.

Referenced by ssl\_client\_start(), and ssl\_server\_start().

#### **8.103.4.13 int ssl\_parse\_finished (ssl\_context \* ssl)**

Definition at line 1157 of file ssl\_tls.c.

Referenced by ssl\_client\_start(), and ssl\_server\_start().

#### **8.103.4.14 int ssl\_read (ssl\_context \* ssl, unsigned char \* buf, int \* len)**

Read at most 'len' application data bytes.

**Returns:**

0 if successful, ERR\_NET\_WOULD\_BLOCK (only when the socket is set to non-blocking), or a specific SSL error code.

**Note:**

len is updated to reflect the actual number of data bytes read.

Definition at line 1390 of file ssl\_tls.c.

References ERR\_SSL\_UNEXPECTED\_MESSAGE, ssl\_context::in\_msg, ssl\_context::in\_msglen, ssl\_context::in\_msgtype, ssl\_context::in\_offt, ssl\_handshake(), SSL\_MSG\_APPLICATION\_DATA, and ssl\_read\_record().

Referenced by main().

**8.103.4.15 int ssl\_read\_record (ssl\_context \* ssl, int do\_crypt)**

Definition at line 685 of file ssl\_tls.c.

Referenced by ssl\_parse\_certificate\_request(), ssl\_parse\_certificate\_verify(), ssl\_parse\_client\_key\_exchange(), ssl\_parse\_server\_hello(), ssl\_parse\_server\_hello\_done(), ssl\_parse\_server\_key\_exchange(), and ssl\_read().

**8.103.4.16 int ssl\_server\_start (ssl\_context \* ssl)**

Definition at line 666 of file ssl\_srv.c.

References SSL\_CERTIFICATE\_REQUEST, SSL\_CERTIFICATE\_VERIFY, SSL\_CLIENT\_CERTIFICATE, SSL\_CLIENT\_CHANGE\_CIPHER\_SPEC, SSL\_CLIENT\_FINISHED, SSL\_CLIENT\_HELLO, SSL\_CLIENT\_KEY\_EXCHANGE, ssl\_flush\_output(), SSL\_HELLO\_REQUEST, ssl\_parse\_certificate(), ssl\_parse\_certificate\_verify(), ssl\_parse\_change\_cipher\_spec(), ssl\_parse\_client\_hello(), ssl\_parse\_client\_key\_exchange(), ssl\_parse\_finished(), SSL\_SERVER\_CERTIFICATE, SSL\_SERVER\_CHANGE\_CIPHER\_SPEC, SSL\_SERVER\_FINISHED, SSL\_SERVER\_HELLO, SSL\_SERVER\_HELLO\_DONE, SSL\_SERVER\_KEY\_EXCHANGE, ssl\_write\_certificate(), ssl\_write\_certificate\_request(), ssl\_write\_change\_cipher\_spec(), ssl\_write\_finished(), ssl\_write\_server\_hello(), ssl\_write\_server\_hello\_done(), ssl\_write\_server\_key\_exchange(), and ssl\_context::state.

Referenced by main(), and ssl\_handshake().

**8.103.4.17 void ssl\_set\_authmode (ssl\_context \* ssl, int authmode)**

Set the certificate verification mode.

**Parameters:**

*mode* can be:

SSL\_VERIFY\_NONE: peer certificate is not checked (default).

SSL\_VERIFY\_OPTIONAL: peer certificate is checked, however the handshake continues even if verification failed; you may want to check ssl->verify\_result after.

SSL\_VERIFY\_REQUIRED: peer *must* present a valid certificate, handshake is aborted if verification failed.

Definition at line 1252 of file ssl\_tls.c.

Referenced by main().

**8.103.4.18 void ssl\_set\_ca\_chain (ssl\_context \* ssl, x509\_cert \* ca, char \* cn)**

Set the CA certificate chain used to verify peer cert, and the peer's expected CommonName (or NULL).

Definition at line 1276 of file ssl\_tls.c.

Referenced by main().

**8.103.4.19 void ssl\_set\_ciphlist (ssl\_context \* ssl, int \* ciphers)**

Set the list of allowed ciphersuites.

Definition at line 1271 of file ssl\_tls.c.

Referenced by main().

**8.103.4.20 int ssl\_set\_dhm\_vals (ssl\_context \* ssl, char \* dhm\_P, char \* dhm\_G)**

Set the Diffie-Hellman public P and G values, provided as hexadecimal strings (server-side only).

**Returns:**

0 if successful, or 1 if the values could not be read

Definition at line 1294 of file ssl\_tls.c.

Referenced by main().

**8.103.4.21 void ssl\_set\_endpoint (ssl\_context \* ssl, int endpoint)**

Set the current endpoint type, SSL\_IS\_CLIENT or SSL\_IS\_SERVER.

Definition at line 1247 of file ssl\_tls.c.

References ssl\_context::endpoint.

Referenced by main().

**8.103.4.22 void ssl\_set\_io\_files (ssl\_context \* ssl, int read\_fd, int write\_fd)**

Set the read and write file descriptors.

Definition at line 1265 of file ssl\_tls.c.

Referenced by main().

**8.103.4.23 void ssl\_set\_rng\_func (ssl\_context \* ssl, int(\*) (void \*) rng\_f, void \* rng\_d)**

Set the random number generator function.

Definition at line 1257 of file ssl\_tls.c.

Referenced by main().

**8.103.4.24 void ssl\_set\_rsa\_cert (ssl\_context \* *ssl*, x509\_cert \* *own\_cert*, rsa\_context \* *own\_key*)**

Set own certificate and private RSA key.

Definition at line 1282 of file ssl\_tls.c.

Referenced by main().

**8.103.4.25 void ssl\_set\_sidtable (ssl\_context \* *ssl*, unsigned char \* *sidtable*)**

Set the global session ID table (server-side only).

Definition at line 1289 of file ssl\_tls.c.

Referenced by main().

**8.103.4.26 int ssl\_write (ssl\_context \* *ssl*, unsigned char \* *buf*, int *len*)**

Write 'len' application data bytes.

**Returns:**

0 if successful, ERR\_NET\_WOULD\_BLOCK (only when the socket is set to non-blocking), or a specific SSL error code.

**Note:**

When the socket is set to non-blocking and this function returns ERR\_NET\_WOULD\_BLOCK, it should be called again with the *\*same\** arguments until it returns 0.

Definition at line 1425 of file ssl\_tls.c.

References ssl\_context::out\_msg, ssl\_context::out\_msglen, ssl\_context::out\_msgtype, ssl\_context::out\_uoff, ssl\_handshake(), SSL\_MAX\_CONTENT\_LEN, SSL\_MSG\_APPLICATION\_DATA, and ssl\_write\_record().

Referenced by main().

**8.103.4.27 int ssl\_write\_certificate (ssl\_context \* *ssl*)**

Definition at line 846 of file ssl\_tls.c.

References ssl\_context::client\_auth, ssl\_context::endpoint, ERR\_SSL\_CERTIFICATE\_REQUIRED, ERR\_SSL\_CERTIFICATE\_TOO\_LARGE, \_x509\_buf::len, ssl\_context::minor\_ver, \_x509\_cert::next, ssl\_context::out\_msg, ssl\_context::out\_msglen, ssl\_context::out\_msgtype, ssl\_context::own\_cert, \_x509\_buf::p, \_x509\_cert::raw, SSL\_ALERT\_NO\_CERTIFICATE, SSL\_ALERT\_WARNING, SSL\_HS\_CERTIFICATE, SSL\_IS\_CLIENT, SSL\_MAX\_CONTENT\_LEN, SSL\_MSG\_ALERT, SSL\_MSG\_HANDSHAKE, ssl\_write\_record(), SSLV3\_MINOR\_VERSION, and ssl\_context::state.

Referenced by ssl\_client\_start(), and ssl\_server\_start().

**8.103.4.28 int ssl\_write\_change\_cipher\_spec (ssl\_context \* *ssl*)**

Definition at line 1022 of file ssl\_tls.c.

Referenced by ssl\_client\_start(), and ssl\_server\_start().

#### **8.103.4.29 int ssl\_write\_finished (ssl\_context \* ssl)**

Definition at line 1120 of file ssl\_tls.c.

Referenced by ssl\_client\_start(), and ssl\_server\_start().

#### **8.103.4.30 int ssl\_write\_record (ssl\_context \* ssl, int do\_crypt)**

Definition at line 648 of file ssl\_tls.c.

References ssl\_context::hs\_md5, ssl\_context::hs\_sha1, ssl\_context::major\_ver, md5\_update(), ssl\_context::minor\_ver, net\_send(), ssl\_context::out\_hdr, ssl\_context::out\_left, ssl\_context::out\_msg, ssl\_context::out\_msglen, ssl\_context::out\_msgtype, sha1\_update(), ssl\_encrypt\_buf(), SSL\_MSG\_HANDSHAKE, and ssl\_context::write\_fd.

Referenced by ssl\_close\_notify(), ssl\_write(), ssl\_write\_certificate(), ssl\_write\_certificate\_request(), ssl\_write\_certificate\_verify(), ssl\_write\_client\_hello(), ssl\_write\_client\_key\_exchange(), ssl\_write\_server\_hello(), ssl\_write\_server\_hello\_done(), and ssl\_write\_server\_key\_exchange().

### **8.103.5 Variable Documentation**

#### **8.103.5.1 int ssl\_default\_ciphers[ ]**

Definition at line 1346 of file ssl\_tls.c.

Referenced by main().

## 8.104 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/ssl\_conf.h File Reference

### 8.104.1 Detailed Description

Defines the set of features `_not_` available, this is mostly useful for embedded systems.

Both MD5 and SHA-1 are required for SSL and therefore cannot be disabled here.

When compiling for ARMv3 / ARMv4T with gcc 3.4.6 -Os:

NO\_GENPRIME saves 2196 bytes NO\_DHM saves 2220 bytes NO\_MD2 saves 2056 bytes NO\_MD4 saves 3948 bytes NO\_ARC4 saves 840 bytes NO\_DES saves 18016 bytes NO\_AES saves 19228 bytes NO\_SSL\_CLI saves 2644 bytes NO\_SSL\_SRV saves 3348 bytes

Definition in file [ssl\\_conf.h](#).



## 8.105 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/timing.h File Reference

### Data Structures

- struct [hr\\_time](#)  
*timer structure*

### Functions

- unsigned long [hardclock](#) (void)  
*Return the CPU cycle counter value.*
- unsigned long [set\\_timer](#) (struct [hr\\_time](#) \*val, int reset)  
*Return the elapsed time in milliseconds.*
- void [set\\_alarm](#) (int seconds)  
*Setup an alarm clock.*

### Variables

- int [alarmed](#)

#### 8.105.1 Detailed Description

Definition in file [timing.h](#).

#### 8.105.2 Function Documentation

##### 8.105.2.1 unsigned long hardclock (void)

Return the CPU cycle counter value.

Definition at line 138 of file timing.c.

References [tv\\_init](#).

Referenced by [main\(\)](#).

##### 8.105.2.2 void set\_alarm (int seconds)

Setup an alarm clock.

#### Parameters:

*seconds* delay before the "alarmed" flag is set

Definition at line 230 of file timing.c.

References sighandler().

Referenced by main().

#### **8.105.2.3 unsigned long set\_timer (struct hr\_time \* *val*, int *reset*)**

Return the elapsed time in milliseconds.

##### **Parameters:**

*val* points to a timer structure

*reset* if set to 1, the timer is restarted

Definition at line 204 of file timing.c.

References \_hr\_time::start.

### **8.105.3 Variable Documentation**

#### **8.105.3.1 int alarmed**

Definition at line 163 of file timing.c.

Referenced by main().

## 8.106 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/x509.h File Reference

```
#include "rsa.h"
```

### Data Structures

- struct [\\_x509\\_buf](#)
- struct [\\_x509\\_name](#)
- struct [\\_x509\\_time](#)
- struct [\\_x509\\_cert](#)

### Defines

- #define [ERR\\_ASN1\\_OUT\\_OF\\_DATA](#) 0x0014
- #define [ERR\\_ASN1\\_UNEXPECTED\\_TAG](#) 0x0016
- #define [ERR\\_ASN1\\_INVALID\\_LENGTH](#) 0x0018
- #define [ERR\\_ASN1\\_LENGTH\\_MISMATCH](#) 0x001A
- #define [ERR\\_ASN1\\_INVALID\\_DATA](#) 0x001C
- #define [ERR\\_X509\\_FEATURE\\_UNAVAILABLE](#) 0x0020
- #define [ERR\\_X509\\_CERT\\_INVALID\\_PEM](#) 0x0040
- #define [ERR\\_X509\\_CERT\\_INVALID\\_FORMAT](#) 0x0060
- #define [ERR\\_X509\\_CERT\\_INVALID\\_VERSION](#) 0x0080
- #define [ERR\\_X509\\_CERT\\_INVALID\\_SERIAL](#) 0x00A0
- #define [ERR\\_X509\\_CERT\\_INVALID\\_ALG](#) 0x00C0
- #define [ERR\\_X509\\_CERT\\_INVALID\\_NAME](#) 0x00E0
- #define [ERR\\_X509\\_CERT\\_INVALID\\_DATE](#) 0x0100
- #define [ERR\\_X509\\_CERT\\_INVALID\\_PUBKEY](#) 0x0120
- #define [ERR\\_X509\\_CERT\\_INVALID\\_SIGNATURE](#) 0x0140
- #define [ERR\\_X509\\_CERT\\_INVALID\\_EXTENSIONS](#) 0x0160
- #define [ERR\\_X509\\_CERT\\_UNKNOWN\\_VERSION](#) 0x0180
- #define [ERR\\_X509\\_CERT\\_UNKNOWN\\_SIG\\_ALG](#) 0x01A0
- #define [ERR\\_X509\\_CERT\\_UNKNOWN\\_PK\\_ALG](#) 0x01C0
- #define [ERR\\_X509\\_CERT\\_SIG\\_MISMATCH](#) 0x01E0
- #define [ERR\\_X509\\_KEY\\_INVALID\\_PEM](#) 0x0200
- #define [ERR\\_X509\\_KEY\\_INVALID\\_VERSION](#) 0x0220
- #define [ERR\\_X509\\_KEY\\_INVALID\\_FORMAT](#) 0x0240
- #define [ERR\\_X509\\_KEY\\_INVALID\\_ENC\\_IV](#) 0x0260
- #define [ERR\\_X509\\_KEY\\_UNKNOWN\\_ENC\\_ALG](#) 0x0280
- #define [ERR\\_X509\\_KEY\\_PASSWORD\\_REQUIRED](#) 0x02A0
- #define [ERR\\_X509\\_KEY\\_PASSWORD\\_MISMATCH](#) 0x02C0
- #define [ERR\\_X509\\_SIG\\_VERIFY\\_FAILED](#) 0x02E0
- #define [BADCERT\\_HAS\\_EXPIRED](#) 1
- #define [BADCERT\\_CN\\_MISMATCH](#) 2
- #define [BADCERT\\_NOT\\_TRUSTED](#) 4
- #define [ASN1\\_BOOLEAN](#) 0x01
- #define [ASN1\\_INTEGER](#) 0x02
- #define [ASN1\\_BIT\\_STRING](#) 0x03
- #define [ASN1\\_OCTET\\_STRING](#) 0x04

- #define `ASN1_NULL` 0x05
- #define `ASN1_OID` 0x06
- #define `ASN1_UTF8_STRING` 0x0C
- #define `ASN1_SEQUENCE` 0x10
- #define `ASN1_SET` 0x11
- #define `ASN1_PRINTABLE_STRING` 0x13
- #define `ASN1_T61_STRING` 0x14
- #define `ASN1_IA5_STRING` 0x16
- #define `ASN1_UTC_TIME` 0x17
- #define `ASN1_UNIVERSAL_STRING` 0x1C
- #define `ASN1_BMP_STRING` 0x1E
- #define `ASN1_PRIMITIVE` 0x00
- #define `ASN1_CONSTRUCTED` 0x20
- #define `ASN1_CONTEXT_SPECIFIC` 0x80
- #define `X520_COMMON_NAME` 3
- #define `X520_COUNTRY` 6
- #define `X520_LOCALITY` 7
- #define `X520_STATE` 8
- #define `X520_ORGANIZATION` 10
- #define `X520_ORG_UNIT` 11
- #define `PKCS9_EMAIL` 1
- #define `OID_X520` "\x55\x04"
- #define `OID_PKCS1` "\x2A\x86\x48\x86\xF7\x0D\x01\x01"
- #define `OID_PKCS1_RSA` "\x2A\x86\x48\x86\xF7\x0D\x01\x01\x01"
- #define `OID_PKCS9` "\x2A\x86\x48\x86\xF7\x0D\x01\x09"

## Typedefs

- typedef struct `_x509_buf` `x509_buf`
- typedef struct `_x509_name` `x509_name`
- typedef struct `_x509_time` `x509_time`
- typedef struct `_x509_cert` `x509_cert`

## Functions

- `int x509_add_certs` (`x509_cert` \*chain, unsigned char \*buf, `int` buflen)  
*Parse one or more certificates and add them to the chain.*
- `int x509_read_crtfile` (`x509_cert` \*chain, char \*path)  
*Load one or more certificates and add them to the chain.*
- `int x509_parse_key` (`rsa_context` \*rsa, unsigned char \*buf, `int` buflen, unsigned char \*pwd, `int` pwrlen)  
*Parse a private RSA key.*
- `int x509_read_keyfile` (`rsa_context` \*rsa, char \*path, char \*password)  
*Load and parse a private RSA key.*
- `int x509_dn_gets` (char \*buf, char \*end, `x509_name` \*dn)

*Store the certificate DN in printable form into buf; no more than (end - buf) characters will be written.*

- `char * x509_cert_info (x509_cert *crt)`

*Return an informational string about the certificate, or NULL if memory allocation failed.*

- `int x509_is_cert_expired (x509_cert *crt)`

*Return 0 if the certificate is still valid, or BADCERT\_HAS\_EXPIRED.*

- `int x509_verify_cert (x509_cert *crt, x509_cert *trust_ca, char *cn, int *flags)`

*Verify the certificate signature.*

- `void x509_free_cert (x509_cert *crt)`

*Unallocate all certificate data.*

- `int x509_self_test (int verbose)`

*Checkup routine.*

## **8.106.1 Detailed Description**

Definition in file [x509.h](#).

## **8.106.2 Define Documentation**

### **8.106.2.1 #define ASN1\_BIT\_STRING 0x03**

Definition at line 55 of file [x509.h](#).

Referenced by [x509\\_get\\_pubkey\(\)](#), and [x509\\_get\\_sig\(\)](#).

### **8.106.2.2 #define ASN1\_BMP\_STRING 0x1E**

Definition at line 67 of file [x509.h](#).

Referenced by [x509\\_get\\_name\(\)](#).

### **8.106.2.3 #define ASN1\_BOOLEAN 0x01**

Definition at line 53 of file [x509.h](#).

Referenced by [asn1\\_get\\_bool\(\)](#).

### **8.106.2.4 #define ASN1\_CONSTRUCTED 0x20**

Definition at line 69 of file [x509.h](#).

Referenced by [x509\\_add\\_certs\(\)](#), [x509\\_get\\_alg\(\)](#), [x509\\_get\\_dates\(\)](#), [x509\\_get\\_ext\(\)](#), [x509\\_get\\_name\(\)](#), [x509\\_get\\_pubkey\(\)](#), [x509\\_get\\_uid\(\)](#), [x509\\_get\\_version\(\)](#), and [x509\\_parse\\_key\(\)](#).

**8.106.2.5 #define ASN1\_CONTEXT\_SPECIFIC 0x80**

Definition at line 70 of file x509.h.

Referenced by x509\_get\_ext(), x509\_get\_serial(), x509\_get\_uid(), and x509\_get\_version().

**8.106.2.6 #define ASN1\_IA5\_STRING 0x16**

Definition at line 64 of file x509.h.

Referenced by x509\_get\_name().

**8.106.2.7 #define ASN1\_INTEGER 0x02**

Definition at line 54 of file x509.h.

Referenced by asn1\_get\_int(), asn1\_get\_mpi(), and x509\_get\_serial().

**8.106.2.8 #define ASN1\_NULL 0x05**

Definition at line 57 of file x509.h.

Referenced by x509\_get\_alg().

**8.106.2.9 #define ASN1\_OCTET\_STRING 0x04**

Definition at line 56 of file x509.h.

Referenced by x509\_get\_ext().

**8.106.2.10 #define ASN1\_OID 0x06**

Definition at line 58 of file x509.h.

Referenced by x509\_get\_alg(), and x509\_get\_name().

**8.106.2.11 #define ASN1\_PRIMITIVE 0x00**

Definition at line 68 of file x509.h.

Referenced by x509\_get\_serial().

**8.106.2.12 #define ASN1\_PRINTABLE\_STRING 0x13**

Definition at line 62 of file x509.h.

Referenced by x509\_get\_name().

**8.106.2.13 #define ASN1\_SEQUENCE 0x10**

Definition at line 60 of file x509.h.

Referenced by x509\_add\_certs(), x509\_get\_alg(), x509\_get\_dates(), x509\_get\_ext(), x509\_get\_name(), x509\_get\_pubkey(), and x509\_parse\_key().

**8.106.2.14 #define ASN1\_SET 0x11**

Definition at line 61 of file x509.h.

Referenced by x509\_get\_name().

**8.106.2.15 #define ASN1\_T61\_STRING 0x14**

Definition at line 63 of file x509.h.

Referenced by x509\_get\_name().

**8.106.2.16 #define ASN1\_UNIVERSAL\_STRING 0x1C**

Definition at line 66 of file x509.h.

Referenced by x509\_get\_name().

**8.106.2.17 #define ASN1.UTC\_TIME 0x17**

Definition at line 65 of file x509.h.

Referenced by x509\_get\_dates().

**8.106.2.18 #define ASN1\_UTF8\_STRING 0x0C**

Definition at line 59 of file x509.h.

Referenced by x509\_get\_name().

**8.106.2.19 #define BADCERT\_CN\_MISMATCH 2**

Definition at line 47 of file x509.h.

Referenced by main(), and x509\_verify\_cert().

**8.106.2.20 #define BADCERT\_HAS\_EXPIRED 1**

Definition at line 46 of file x509.h.

Referenced by main(), and x509\_is\_cert\_expired().

**8.106.2.21 #define BADCERT\_NOT\_TRUSTED 4**

Definition at line 48 of file x509.h.

Referenced by main(), and x509\_verify\_cert().

**8.106.2.22 #define ERR\_ASN1\_INVALID\_DATA 0x001C**

Definition at line 20 of file x509.h.

**8.106.2.23 #define ERR\_ASN1\_INVALID\_LENGTH 0x0018**

Definition at line 18 of file x509.h.

Referenced by `asn1_get_bool()`, `asn1_get_int()`, and `asn1_get_len()`.

**8.106.2.24 #define ERR\_ASN1\_LENGTH\_MISMATCH 0x001A**

Definition at line 19 of file x509.h.

Referenced by `x509_add_certs()`, `x509_get_alg()`, `x509_get_dates()`, `x509_get_ext()`, `x509_get_name()`, `x509_get_pubkey()`, `x509_get_version()`, and `x509_parse_key()`.

**8.106.2.25 #define ERR\_ASN1\_OUT\_OF\_DATA 0x0014**

Definition at line 16 of file x509.h.

Referenced by `asn1_get_len()`, `asn1_get_tag()`, `x509_get_name()`, `x509_get_pubkey()`, and `x509_get_serial()`.

**8.106.2.26 #define ERR\_ASN1\_UNEXPECTED\_TAG 0x0016**

Definition at line 17 of file x509.h.

Referenced by `asn1_get_tag()`, `x509_get_ext()`, `x509_get_name()`, `x509_get_serial()`, `x509_get_uid()`, and `x509_get_version()`.

**8.106.2.27 #define ERR\_X509\_CERT\_INVALID\_ALG 0x00C0**

Definition at line 27 of file x509.h.

Referenced by `x509_get_alg()`.

**8.106.2.28 #define ERR\_X509\_CERT\_INVALID\_DATE 0x0100**

Definition at line 29 of file x509.h.

Referenced by `x509_get_dates()`.

**8.106.2.29 #define ERR\_X509\_CERT\_INVALID\_EXTENSIONS 0x0160**

Definition at line 32 of file x509.h.

Referenced by `x509_get_ext()`.



**8.106.2.30 #define ERR\_X509\_CERT\_INVALID\_FORMAT 0x0060**

Definition at line 24 of file x509.h.

Referenced by x509\_add\_certs().

**8.106.2.31 #define ERR\_X509\_CERT\_INVALID\_NAME 0x00E0**

Definition at line 28 of file x509.h.

Referenced by x509\_get\_name().

**8.106.2.32 #define ERR\_X509\_CERT\_INVALID\_PEM 0x0040**

Definition at line 23 of file x509.h.

Referenced by x509\_add\_certs().

**8.106.2.33 #define ERR\_X509\_CERT\_INVALID\_PUBKEY 0x0120**

Definition at line 30 of file x509.h.

Referenced by x509\_get\_pubkey().

**8.106.2.34 #define ERR\_X509\_CERT\_INVALID\_SERIAL 0x00A0**

Definition at line 26 of file x509.h.

Referenced by x509\_get\_serial().

**8.106.2.35 #define ERR\_X509\_CERT\_INVALID\_SIGNATURE 0x0140**

Definition at line 31 of file x509.h.

Referenced by x509\_get\_sig().

**8.106.2.36 #define ERR\_X509\_CERT\_INVALID\_VERSION 0x0080**

Definition at line 25 of file x509.h.

Referenced by x509\_get\_version().

**8.106.2.37 #define ERR\_X509\_CERT\_SIG\_MISMATCH 0x01E0**

Definition at line 36 of file x509.h.

Referenced by x509\_add\_certs().

**8.106.2.38 #define ERR\_X509\_CERT\_UNKNOWN\_PK\_ALG 0x01C0**

Definition at line 35 of file x509.h.

Referenced by x509\_get\_pubkey().

**8.106.2.39 #define ERR\_X509\_CERT\_UNKNOWN\_SIG\_ALG 0x01A0**

Definition at line 34 of file x509.h.

Referenced by x509\_add\_certs().

**8.106.2.40 #define ERR\_X509\_CERT\_UNKNOWN\_VERSION 0x0180**

Definition at line 33 of file x509.h.

Referenced by x509\_add\_certs().

**8.106.2.41 #define ERR\_X509\_FEATURE\_UNAVAILABLE 0x0020**

Definition at line 22 of file x509.h.

Referenced by x509\_parse\_key().

**8.106.2.42 #define ERR\_X509\_KEY\_INVALID\_ENC\_IV 0x0260**

Definition at line 40 of file x509.h.

Referenced by x509\_des3\_getiv(), and x509\_parse\_key().

**8.106.2.43 #define ERR\_X509\_KEY\_INVALID\_FORMAT 0x0240**

Definition at line 39 of file x509.h.

Referenced by x509\_parse\_key().

**8.106.2.44 #define ERR\_X509\_KEY\_INVALID\_PEM 0x0200**

Definition at line 37 of file x509.h.

Referenced by x509\_parse\_key().

**8.106.2.45 #define ERR\_X509\_KEY\_INVALID\_VERSION 0x0220**

Definition at line 38 of file x509.h.

Referenced by x509\_parse\_key().

**8.106.2.46 #define ERR\_X509\_KEY\_PASSWORD\_MISMATCH 0x02C0**

Definition at line 43 of file x509.h.

Referenced by x509\_parse\_key().

**8.106.2.47 #define ERR\_X509\_KEY\_PASSWORD\_REQUIRED 0x02A0**

Definition at line 42 of file x509.h.

Referenced by x509\_parse\_key().

**8.106.2.48 #define ERR\_X509\_KEY\_UNKNOWN\_ENC\_ALG 0x0280**

Definition at line 41 of file x509.h.

Referenced by x509\_parse\_key().

**8.106.2.49 #define ERR\_X509\_SIG\_VERIFY\_FAILED 0x02E0**

Definition at line 44 of file x509.h.

Referenced by x509\_verify\_cert().

**8.106.2.50 #define OID\_PKCS1 "\x2A\x86\x48\x86\xF7\x0D\x01\x01"**

Definition at line 84 of file x509.h.

Referenced by x509\_add\_certs().

**8.106.2.51 #define OID\_PKCS1\_RSA "\x2A\x86\x48\x86\xF7\x0D\x01\x01\x01"**

Definition at line 85 of file x509.h.

Referenced by x509\_get\_pubkey().

**8.106.2.52 #define OID\_PKCS9 "\x2A\x86\x48\x86\xF7\x0D\x01\x09"**

Definition at line 86 of file x509.h.

Referenced by x509\_dn\_gets().

**8.106.2.53 #define OID\_X520 "\x55\x04"**

Definition at line 83 of file x509.h.

Referenced by x509\_dn\_gets().

**8.106.2.54 #define PKCS9\_EMAIL 1**

Definition at line 81 of file x509.h.

Referenced by x509\_dn\_gets().

**8.106.2.55 #define X520\_COMMON\_NAME 3**

Definition at line 75 of file x509.h.

Referenced by x509\_dn\_gets().

**8.106.2.56 #define X520\_COUNTRY 6**

Definition at line 76 of file x509.h.

Referenced by x509\_dn\_gets().

**8.106.2.57 #define X520\_LOCALITY 7**

Definition at line 77 of file x509.h.

Referenced by x509\_dn\_gets().

**8.106.2.58 #define X520\_ORG\_UNIT 11**

Definition at line 80 of file x509.h.

Referenced by x509\_dn\_gets().

**8.106.2.59 #define X520\_ORGANIZATION 10**

Definition at line 79 of file x509.h.

Referenced by x509\_dn\_gets().

**8.106.2.60 #define X520\_STATE 8**

Definition at line 78 of file x509.h.

Referenced by x509\_dn\_gets().

**8.106.3 Typedef Documentation****8.106.3.1 typedef struct \_x509\_buf x509\_buf****8.106.3.2 typedef struct \_x509\_cert x509\_cert****8.106.3.3 typedef struct \_x509\_name x509\_name****8.106.3.4 typedef struct \_x509\_time x509\_time****8.106.4 Function Documentation****8.106.4.1 int x509\_add\_certs (x509\_cert \* *chain*, unsigned char \* *buf*, int *buflen*)**

Parse one or more certificates and add them to the chain.

**Parameters:**

*chain* points to the start of the chain

*buf* buffer holding the certificate data

*buflen* size of the buffer

**Returns:**

0 if successful, or a specific X509 error code

Definition at line 646 of file x509read.c.

References ASN1\_CONSTRUCTED, asn1\_get\_tag(), ASN1\_SEQUENCE, base64\_decode(), \_x509\_cert::ca\_istrue, rsa\_context::E, ERR\_ASN1\_LENGTH\_MISMATCH, ERR\_BASE64\_INVALID\_CHARACTER, ERR\_X509\_CERT\_INVALID\_FORMAT, ERR\_X509\_CERT\_INVALID\_PEM, ERR\_X509\_CERT\_SIG\_MISMATCH, ERR\_X509\_CERT\_UNKNOWN\_SIG\_ALG, ERR\_X509\_CERT\_UNKNOWN\_VERSION, \_x509\_cert::issuer, \_x509\_cert::issuer\_id, \_x509\_cert::issuer\_raw, rsa\_context::len, \_x509\_buf::len, \_x509\_cert::max\_pathlen, mpi\_msb(), rsa\_context::N, \_x509\_cert::next, OID\_PKCS1, \_x509\_buf::p, \_x509\_cert::pk\_oid, \_x509\_cert::raw, \_x509\_cert::rsa, rsa\_check\_pubkey(), \_x509\_cert::serial, \_x509\_cert::sig, \_x509\_cert::sig\_oid1, \_x509\_cert::sig\_oid2, \_x509\_cert::subject, \_x509\_cert::subject\_id, \_x509\_cert::subject\_raw, \_x509\_cert::tbs, \_x509\_cert::v3\_ext, \_x509\_cert::valid\_from, \_x509\_cert::valid\_to, \_x509\_cert::version, x509\_add\_certs(), x509\_free\_cert(), x509\_get\_alg(), x509\_get\_dates(), x509\_get\_ext(), x509\_get\_name(), x509\_get\_pubkey(), x509\_get\_serial(), x509\_get\_sig(), x509\_get\_uid(), and x509\_get\_version().

Referenced by main(), x509\_add\_certs(), and x509\_read\_crtfile().

#### 8.106.4.2 char\* x509\_cert\_info (x509\_cert \* crt)

Return an informational string about the certificate, or NULL if memory allocation failed.

Definition at line 1391 of file x509read.c.

References \_x509\_time::day, \_x509\_time::hour, \_x509\_cert::issuer, \_x509\_buf::len, \_x509\_time::min, \_x509\_time::mon, mpi::n, rsa\_context::N, \_x509\_buf::p, \_x509\_cert::rsa, RSA\_MD2, RSA\_MD4, RSA\_MD5, RSA\_SHA1, \_x509\_time::sec, \_x509\_cert::serial, \_x509\_cert::sig\_oid1, \_x509\_cert::subject, \_x509\_cert::valid\_from, \_x509\_cert::valid\_to, \_x509\_cert::version, x509\_dn\_gets(), and \_x509\_time::year.

#### 8.106.4.3 int x509\_dn\_gets (char \* buf, char \* end, x509\_name \* dn)

Store the certificate DN in printable form into buf; no more than (end - buf) characters will be written.

Definition at line 1308 of file x509read.c.

References \_x509\_buf::len, \_x509\_name::next, \_x509\_name::oid, OID\_PKCS9, OID\_X520, \_x509\_buf::p, PKCS9\_EMAIL, \_x509\_name::val, X520\_COMMON\_NAME, X520\_COUNTRY, X520\_LOCALITY, X520\_ORG\_UNIT, X520\_ORGANIZATION, and X520\_STATE.

Referenced by x509\_cert\_info().

#### 8.106.4.4 void x509\_free\_cert (x509\_cert \* crt)

Unallocate all certificate data.

Definition at line 1598 of file x509read.c.

References \_x509\_cert::issuer, \_x509\_cert::next, \_x509\_name::next, \_x509\_buf::p, \_x509\_cert::raw, \_x509\_cert::rsa, rsa\_free(), and \_x509\_cert::subject.

Referenced by main(), ssl\_free(), and x509\_add\_certs().

#### 8.106.4.5 int x509\_is\_cert\_expired (x509\_cert \* crt)

Return 0 if the certificate is still valid, or BADCERT\_HAS\_EXPIRED.

Definition at line 1452 of file x509read.c.

References BADCERT\_HAS\_EXPIRED, \_x509\_time::day, \_x509\_time::mon, \_x509\_cert::valid\_to, and \_x509\_time::year.

Referenced by x509\_verify\_cert().

#### 8.106.4.6 `int x509_parse_key(rsa_context *rsa, unsigned char *buf, int buflen, unsigned char *pwd, int pwrlen)`

Parse a private RSA key.

##### Parameters:

*rsa* RSA context to be initialized

*buf* input buffer

*buflen* size of the buffer

*pwd* password for decryption (optional)

*pwrlen* size of the password

##### Returns:

0 if successful, or a specific X509 error code

Definition at line 1077 of file x509read.c.

References ASN1\_CONSTRUCTED, asn1\_get\_int(), asn1\_get\_mpi(), asn1\_get\_tag(), ASN1\_SEQUENCE, base64\_decode(), rsa\_context::D, rsa\_context::DP, rsa\_context::DQ, rsa\_context::E, ERR\_ASN1\_LENGTH\_MISMATCH, ERR\_BASE64\_INVALID\_CHARACTER, ERR\_X509\_FEATURE\_UNAVAILABLE, ERR\_X509\_KEY\_INVALID\_ENC\_IV, ERR\_X509\_KEY\_INVALID\_FORMAT, ERR\_X509\_KEY\_INVALID\_PEM, ERR\_X509\_KEY\_INVALID\_VERSION, ERR\_X509\_KEY\_PASSWORD\_MISMATCH, ERR\_X509\_KEY\_PASSWORD\_REQUIRED, ERR\_X509\_KEY\_UNKNOWN\_ENC\_ALG, rsa\_context::len, mpi\_msb(), rsa\_context::N, rsa\_context::P, rsa\_context::Q, rsa\_context::QP, rsa\_check\_privkey(), rsa\_free(), rsa\_context::ver, x509\_des3\_decrypt(), and x509\_des3\_getiv().

Referenced by main(), and x509\_read\_keyfile().

#### 8.106.4.7 `int x509_read_crtfile(x509_cert *chain, char *path)`

Load one or more certificates and add them to the chain.

##### Parameters:

*chain* points to the start of the chain

*path* filename to read the certificates from

##### Returns:

0 if successful, or a specific X509 error code

Definition at line 974 of file x509read.c.

References f, and x509\_add\_certs().

#### 8.106.4.8 `int x509_read_keyfile (rsa_context * rsa, char * path, char * password)`

Load and parse a private RSA key.

##### Parameters:

*rsa* RSA context to be initialized  
*path* filename to read the private key from  
*pwd* password to decrypt the file (can be NULL)

##### Returns:

0 if successful, or a specific X509 error code

Definition at line 1261 of file x509read.c.

References `f`, and `x509_parse_key()`.

#### 8.106.4.9 `int x509_self_test (int verbose)`

Checkup routine.

##### Returns:

0 if successful, or 1 if the test failed

Definition at line 1726 of file x509read.c.

Referenced by `main()`.

#### 8.106.4.10 `int x509_verify_cert (x509_cert * crt, x509_cert * trust_ca, char * cn, int * flags)`

Verify the certificate signature.

##### Parameters:

*crt* a certificate to be verified  
*trust\_ca* the trusted CA chain  
*cn* expected Common Name (can be set to NULL if the CN must not be verified)  
*flags* result of the verification

##### Returns:

0 if successful or `ERR_X509_SIG_VERIFY_FAILED`, in which case *flags* will have one or more of the following values set: `BADCERT_HAS_EXPIRED` – `BADCERT_CN_MISMATCH` – `BADCERT_NOT_TRUSTED`

Definition at line 1497 of file x509read.c.

References `BADCERT_CN_MISMATCH`, `BADCERT_NOT_TRUSTED`, `_x509_cert::ca_istrue`, `ERR_X509_SIG_VERIFY_FAILED`, `_x509_cert::issuer_raw`, `_x509_buf::len`, `_x509_cert::max_pathlen`, `_x509_cert::next`, `_x509_name::next`, `_x509_name::oid`, `_x509_buf::p`, `_x509_cert::rsa`, `rsa_pkcs1_verify()`, `_x509_cert::sig`, `_x509_cert::sig_oid1`, `_x509_cert::subject`, `_x509_cert::subject_raw`, `_x509_cert::tbs`, `_x509_name::val`, `_x509_cert::version`, `x509_hash()`, and `x509_is_cert_expired()`.

## 8.107 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/aes.c File Reference

```
#include <string.h>
#include "xyssl/aes.h"
```

### Defines

- #define [\\_CRT\\_SECURE\\_NO\\_DEPRECATED](#) 1
- #define [uint8](#) unsigned char
- #define [uint32](#) unsigned long
- #define [GET\\_UINT32\\_BE](#)(n, b, i)
- #define [PUT\\_UINT32\\_BE](#)(n, b, i)
- #define [ROTR8](#)(x)
- #define [XTIME](#)(x) ( ( x < 1 ) ^ ( ( x & 0x80 ) ? 0x1B : 0x00 ) )
- #define [MUL](#)(x, y) ( ( x && y ) ? pow[(log[x] + log[y]) % 255] : 0 )
- #define [AES\\_FROUND](#)(X0, X1, X2, X3, Y0, Y1, Y2, Y3)
- #define [AES\\_RROUND](#)(X0, X1, X2, X3, Y0, Y1, Y2, Y3)

### Functions

- static void [aes\\_gen\\_tables](#) (void)
- void [aes\\_set\\_key](#) ([aes\\_context](#) \*ctx, uint8 \*key, int keysize)
- void [aes\\_encrypt](#) ([aes\\_context](#) \*ctx, unsigned char input[16], unsigned char output[16])  
*AES block encryption (ECB mode).*
- void [aes\\_decrypt](#) ([aes\\_context](#) \*ctx, unsigned char input[16], unsigned char output[16])  
*AES block decryption (ECB mode).*
- void [aes\\_cbc\\_encrypt](#) ([aes\\_context](#) \*ctx, unsigned char iv[16], unsigned char \*input, unsigned char \*output, int len)  
*AES-CBC buffer encryption.*
- void [aes\\_cbc\\_decrypt](#) ([aes\\_context](#) \*ctx, unsigned char iv[16], unsigned char \*input, unsigned char \*output, int len)  
*AES-CBC buffer decryption.*
- int [aes\\_self\\_test](#) (int verbose)  
*Checkup routine.*

### Variables

- static uint8 [FSb](#) [256]
- static uint32 [FT0](#) [256]
- static uint32 [FT1](#) [256]
- static uint32 [FT2](#) [256]
- static uint32 [FT3](#) [256]



- static uint8 [RSb](#) [256]
- static uint32 [RT0](#) [256]
- static uint32 [RT1](#) [256]
- static uint32 [RT2](#) [256]
- static uint32 [RT3](#) [256]
- static uint32 [RCON](#) [10]
- static uint32 [KT0](#) [256]
- static uint32 [KT1](#) [256]
- static uint32 [KT2](#) [256]
- static uint32 [KT3](#) [256]
- static const char [\\_aes\\_src](#) [] = "\_aes\_src"

## 8.107.1 Define Documentation

### 8.107.1.1 #define \_CRT\_SECURE\_NO\_DEPRECATED 1

Definition at line 31 of file aes.c.

### 8.107.1.2 #define AES\_FROUND(X0, X1, X2, X3, Y0, Y1, Y2, Y3)

**Value:**

```
{
    RK += 4;

    X0 = RK[0] ^ FT0[ (uint8) ( Y0 >> 24 ) ] ^
            FT1[ (uint8) ( Y1 >> 16 ) ] ^
            FT2[ (uint8) ( Y2 >>  8 ) ] ^
            FT3[ (uint8) ( Y3          ) ];

    X1 = RK[1] ^ FT0[ (uint8) ( Y1 >> 24 ) ] ^
            FT1[ (uint8) ( Y2 >> 16 ) ] ^
            FT2[ (uint8) ( Y3 >>  8 ) ] ^
            FT3[ (uint8) ( Y0          ) ];

    X2 = RK[2] ^ FT0[ (uint8) ( Y2 >> 24 ) ] ^
            FT1[ (uint8) ( Y3 >> 16 ) ] ^
            FT2[ (uint8) ( Y0 >>  8 ) ] ^
            FT3[ (uint8) ( Y1          ) ];

    X3 = RK[3] ^ FT0[ (uint8) ( Y3 >> 24 ) ] ^
            FT1[ (uint8) ( Y0 >> 16 ) ] ^
            FT2[ (uint8) ( Y1 >>  8 ) ] ^
            FT3[ (uint8) ( Y2          ) ];
}
```

### 8.107.1.3 #define AES\_RROUND(X0, X1, X2, X3, Y0, Y1, Y2, Y3)

**Value:**

```
{
    RK += 4;

    X0 = RK[0] ^ RT0[ (uint8) ( Y0 >> 24 ) ] ^
            RT1[ (uint8) ( Y3 >> 16 ) ] ^
```

```

        RT2[ (uint8) ( Y2 >> 8 ) ] ^ \
        RT3[ (uint8) ( Y1          ) ]; \
    \
X1 = RK[1] ^ RT0[ (uint8) ( Y1 >> 24 ) ] ^ \
        RT1[ (uint8) ( Y0 >> 16 ) ] ^ \
        RT2[ (uint8) ( Y3 >> 8 ) ] ^ \
        RT3[ (uint8) ( Y2          ) ]; \
    \
X2 = RK[2] ^ RT0[ (uint8) ( Y2 >> 24 ) ] ^ \
        RT1[ (uint8) ( Y1 >> 16 ) ] ^ \
        RT2[ (uint8) ( Y0 >> 8 ) ] ^ \
        RT3[ (uint8) ( Y3          ) ]; \
    \
X3 = RK[3] ^ RT0[ (uint8) ( Y3 >> 24 ) ] ^ \
        RT1[ (uint8) ( Y2 >> 16 ) ] ^ \
        RT2[ (uint8) ( Y1 >> 8 ) ] ^ \
        RT3[ (uint8) ( Y0          ) ]; \
}

```

#### 8.107.1.4 #define GET\_UINT32\_BE(n, b, i)

**Value:**

```

{
    (n) = ( (uint32) (b)[(i)      ] << 24 ) \
        | ( (uint32) (b)[(i) + 1] << 16 ) \
        | ( (uint32) (b)[(i) + 2] << 8 )  \
        | ( (uint32) (b)[(i) + 3]          ); \
}

```

Definition at line 50 of file aes.c.

Referenced by aes\_set\_key(), des3\_crypt(), des\_crypt(), des\_main\_ks(), sha1\_process(), and sha2\_process().

#### 8.107.1.5 #define MUL(x, y) ( ( x && y ) ? pow[(log[x] + log[y]) % 255] : 0 )

Definition at line 106 of file aes.c.

Referenced by aes\_gen\_tables().

#### 8.107.1.6 #define PUT\_UINT32\_BE(n, b, i)

**Value:**

```

{
    (b)[(i)      ] = (uint8) ( (n) >> 24 ); \
    (b)[(i) + 1] = (uint8) ( (n) >> 16 ); \
    (b)[(i) + 2] = (uint8) ( (n) >> 8 );  \
    (b)[(i) + 3] = (uint8) ( (n)          ); \
}

```

Definition at line 59 of file aes.c.

Referenced by des3\_crypt(), des\_crypt(), sha1\_finish(), and sha2\_finish().

**8.107.1.7 #define ROTR8(x)****Value:**

```
( ( ( x << 24 ) & 0xFFFFFFFF ) | \
    ( ( x & 0xFFFFFFFF ) >> 8 ) )
```

Definition at line 103 of file aes.c.

Referenced by aes\_gen\_tables().

**8.107.1.8 #define uint32 unsigned long**

Definition at line 43 of file aes.c.

Referenced by aes\_gen\_tables(), and aes\_set\_key().

**8.107.1.9 #define uint8 unsigned char**

Definition at line 39 of file aes.c.

Referenced by aes\_gen\_tables(), and aes\_set\_key().

**8.107.1.10 #define XTIME(x) ( ( x << 1 ) ^ ( ( x & 0x80 ) ? 0x1B : 0x00 ) )**

Definition at line 105 of file aes.c.

Referenced by aes\_gen\_tables().

**8.107.2 Function Documentation****8.107.2.1 void aes\_cbc\_decrypt (aes\_context \* ctx, unsigned char iv[16], unsigned char \* input, unsigned char \* output, int len)**

AES-CBC buffer decryption.

**Parameters:**

*ctx* AES context  
*iv* initialization vector (modified after use)  
*input* buffer holding the ciphertext  
*output* buffer holding the plaintext  
*len* length of the data to be decrypted

Definition at line 825 of file aes.c.

Referenced by ssl\_decrypt\_buf().

**8.107.2.2 void aes\_cbc\_encrypt (aes\_context \* ctx, unsigned char iv[16], unsigned char \* input, unsigned char \* output, int len)**

AES-CBC buffer encryption.

**Parameters:**

*ctx* AES context  
*iv* initialization vector (modified after use)  
*input* buffer holding the plaintext  
*output* buffer holding the ciphertext  
*len* length of the data to be encrypted

Definition at line 800 of file aes.c.

Referenced by `main()`, and `ssl_encrypt_buf()`.

**8.107.2.3 void aes\_decrypt (aes\_context \* ctx, unsigned char input[16], unsigned char output[16])**

AES block decryption (ECB mode).

**Parameters:**

*ctx* AES context  
*input* ciphertext block  
*output* plaintext block

Definition at line 709 of file aes.c.

Referenced by `main()`.

**8.107.2.4 void aes\_encrypt (aes\_context \* ctx, unsigned char input[16], unsigned char output[16])**

AES block encryption (ECB mode).

AES block encryption (ECB mode)

Definition at line 618 of file aes.c.

Referenced by `main()`.

**8.107.2.5 static void aes\_gen\_tables (void) [static]**

Definition at line 108 of file aes.c.

References `FSb`, `FT0`, `FT1`, `FT2`, `FT3`, `MUL`, `RCON`, `ROTR8`, `RSb`, `RT0`, `RT1`, `RT2`, `RT3`, `uint32`, `uint8`, and `XTIME`.

Referenced by `aes_set_key()`.

**8.107.2.6 int aes\_self\_test (int verbose)**

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 923 of file aes.c.

Referenced by `main()`.

**8.107.2.7 void aes\_set\_key (aes\_context \* ctx, uint8 \* key, int keysize)**

Definition at line 460 of file aes.c.

References aes\_gen\_tables(), aes\_context::drk, aes\_context::erk, FSb, GET\_UINT32\_BE, KT0, KT1, KT2, KT3, aes\_context::nr, RCON, RT0, RT1, RT2, RT3, uint32, and uint8.

**8.107.3 Variable Documentation****8.107.3.1 const char \_aes\_src[] = "\_aes\_src" [static]**

Definition at line 850 of file aes.c.

**8.107.3.2 uint8 FSb[256] [static]**

Definition at line 80 of file aes.c.

Referenced by aes\_gen\_tables(), and aes\_set\_key().

**8.107.3.3 uint32 FT0[256] [static]**

Definition at line 81 of file aes.c.

Referenced by aes\_gen\_tables().

**8.107.3.4 uint32 FT1[256] [static]**

Definition at line 82 of file aes.c.

Referenced by aes\_gen\_tables().

**8.107.3.5 uint32 FT2[256] [static]**

Definition at line 83 of file aes.c.

Referenced by aes\_gen\_tables().

**8.107.3.6 uint32 FT3[256] [static]**

Definition at line 84 of file aes.c.

Referenced by aes\_gen\_tables().

**8.107.3.7 uint32 KT0[256] [static]**

Definition at line 452 of file aes.c.

Referenced by aes\_set\_key().

**8.107.3.8 uint32 KT1[256] [static]**

Definition at line 453 of file aes.c.

Referenced by aes\_set\_key().

**8.107.3.9 uint32 KT2[256] [static]**

Definition at line 454 of file aes.c.

Referenced by aes\_set\_key().

**8.107.3.10 uint32 KT3[256] [static]**

Definition at line 455 of file aes.c.

Referenced by aes\_set\_key().

**8.107.3.11 uint32 RCON[10] [static]**

Definition at line 98 of file aes.c.

Referenced by aes\_gen\_tables(), and aes\_set\_key().

**8.107.3.12 uint8 RSb[256] [static]**

Definition at line 89 of file aes.c.

Referenced by aes\_gen\_tables().

**8.107.3.13 uint32 RT0[256] [static]**

Definition at line 90 of file aes.c.

Referenced by aes\_gen\_tables(), and aes\_set\_key().

**8.107.3.14 uint32 RT1[256] [static]**

Definition at line 91 of file aes.c.

Referenced by aes\_gen\_tables(), and aes\_set\_key().

**8.107.3.15 uint32 RT2[256] [static]**

Definition at line 92 of file aes.c.

Referenced by aes\_gen\_tables(), and aes\_set\_key().

**8.107.3.16 uint32 RT3[256] [static]**

Definition at line 93 of file aes.c.

Referenced by aes\_gen\_tables(), and aes\_set\_key().

## 8.108 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/arc4.c File Reference

```
#include "xyssl/arc4.h"
```

### Defines

- `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

### Functions

- `void arc4\_setup (arc4\_context *ctx, unsigned char *key, int keylen)`  
*ARC4 key schedule.*
- `void arc4\_crypt (arc4\_context *ctx, unsigned char *buf, int buflen)`  
*ARC4 cipher function.*
- `int arc4\_self\_test (int verbose)`

### Variables

- `static const char \_arc4\_src [] = "_arc4_src"`

## 8.108.1 Define Documentation

### 8.108.1.1 `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

Definition at line 30 of file arc4.c.

## 8.108.2 Function Documentation

### 8.108.2.1 `void arc4\_crypt (arc4\_context * ctx, unsigned char * buf, int buflen)`

ARC4 cipher function.

#### Parameters:

*ctx* ARC4 context

*buf* buffer to be processed

*buflen* amount of data in buf

Definition at line 63 of file arc4.c.

References `arc4_context::m`, `arc4_context::x`, and `arc4_context::y`.

Referenced by `main()`, `ssl_decrypt_buf()`, and `ssl_encrypt_buf()`.

### 8.108.2.2 `int arc4_self_test (int verbose)`

Definition at line 150 of file arc4.c.

Referenced by main().

### 8.108.2.3 `void arc4_setup (arc4_context * ctx, unsigned char * key, int keylen)`

ARC4 key schedule.

#### Parameters:

*ctx* ARC4 context to be initialized

*key* the secret key

*keylen* length of the key

Definition at line 38 of file arc4.c.

References arc4\_context::m, arc4\_context::x, and arc4\_context::y.

Referenced by main(), and ssl\_derive\_keys().

## 8.108.3 Variable Documentation

### 8.108.3.1 `const char _arc4_src[] = "_arc4_src" [static]`

Definition at line 84 of file arc4.c.



## 8.109 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/base64.c File Reference

```
#include "xyssl/base64.h"
```

### Defines

- `#define _CRT_SECURE_NO_DEPRECATED` 1

### Functions

- `int base64_encode` (unsigned char \*dst, int \*dlen, unsigned char \*src, int slen)  
*Encode a buffer into base64 format.*
- `int base64_decode` (unsigned char \*dst, int \*dlen, unsigned char \*src, int slen)  
*Decode a base64-formatted buffer.*
- `int base64_self_test` (int verbose)  
*Checkup routine.*

### Variables

- static const int base64\_enc\_map [64]
- static const int base64\_dec\_map [128]
- static const char \_base64\_src [] = "\_base64\_src"

### 8.109.1 Define Documentation

#### 8.109.1.1 `#define _CRT_SECURE_NO_DEPRECATED` 1

Definition at line 25 of file base64.c.

### 8.109.2 Function Documentation

#### 8.109.2.1 `int base64_decode` (unsigned char \*dst, int \*dlen, unsigned char \*src, int slen)

Decode a base64-formatted buffer.

##### Parameters:

- dst* destination buffer
- dlen* size of the buffer (updated after call)
- src* source buffer
- slen* amount of data to be decoded

**Returns:**

0 if successful, ERR\_BASE64\_BUFFER\_TOO\_SMALL, or ERR\_BASE64\_INVALID\_DATA if an invalid char is found. \*dlen is always updated to reflect to amount of data that was written (or would have been written)

**Note:**

Call this function with \*dlen = 0 to obtain the required buffer size in \*dlen

Definition at line 121 of file base64.c.

References base64\_dec\_map, ERR\_BASE64\_BUFFER\_TOO\_SMALL, and ERR\_BASE64\_INVALID\_CHARACTER.

Referenced by x509\_add\_certs(), and x509\_parse\_key().

**8.109.2.2 int base64\_encode (unsigned char \* *dst*, int \* *dlen*, unsigned char \* *src*, int *slen*)**

Encode a buffer into base64 format.

**Parameters:**

*dst* destination buffer  
*dlen* size of the buffer (updated after call)  
*src* source buffer  
*slen* amount of data to be encoded

**Returns:**

0 if successful, or ERR\_BASE64\_BUFFER\_TOO\_SMALL. \*dlen is always updated to reflect to amount of data that was written (or would have been written)

**Note:**

Call this function with \*dlen = 0 to obtain the required buffer size in \*dlen

Definition at line 61 of file base64.c.

References base64\_enc\_map, and ERR\_BASE64\_BUFFER\_TOO\_SMALL.

**8.109.2.3 int base64\_self\_test (int *verbose*)**

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 248 of file base64.c.

Referenced by main().

**8.109.3 Variable Documentation****8.109.3.1 const char \_base64\_src[] = "\_base64\_src" [static]**

Definition at line 182 of file base64.c.

**8.109.3.2 const int base64\_dec\_map[128] [static]****Initial value:**

```
{
    127, 127, 127, 127, 127, 127, 127, 127, 127, 127,
    127, 127, 127, 127, 127, 127, 127, 127, 127, 127,
    127, 127, 127, 127, 127, 127, 127, 127, 127, 127,
    127, 127, 127, 127, 127, 127, 127, 127, 127, 127,
    127, 127, 127, 62, 127, 127, 127, 63, 52, 53,
    54, 55, 56, 57, 58, 59, 60, 61, 127, 127,
    127, 64, 127, 127, 127, 0, 1, 2, 3, 4,
    5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
    15, 16, 17, 18, 19, 20, 21, 22, 23, 24,
    25, 127, 127, 127, 127, 127, 127, 26, 27, 28,
    29, 30, 31, 32, 33, 34, 35, 36, 37, 38,
    39, 40, 41, 42, 43, 44, 45, 46, 47, 48,
    49, 50, 51, 127, 127, 127, 127, 127, 127
}
```

Definition at line 41 of file base64.c.

Referenced by base64\_decode().

**8.109.3.3 const int base64\_enc\_map[64] [static]****Initial value:**

```
{
    'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J',
    'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T',
    'U', 'V', 'W', 'X', 'Y', 'Z', 'a', 'b', 'c', 'd',
    'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n',
    'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x',
    'y', 'z', '0', '1', '2', '3', '4', '5', '6', '7',
    '8', '9', '+', '/'
}
```

Definition at line 30 of file base64.c.

Referenced by base64\_encode().

## 8.110 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/bignum.c File Reference

```
#include <string.h>
#include <stdlib.h>
#include <stdarg.h>
#include <stdio.h>
#include "xyssl/bignum.h"
#include "xyssl/bn_asm.h"
```

### Defines

- #define [\\_CRT\\_SECURE\\_NO\\_DEPRECATED](#) 1
- #define [BITS\\_TO\\_LIMBS](#)(i) (((i) + biL - 1) / biL)
- #define [CHARS\\_TO\\_LIMBS](#)(i) (((i) + ciL - 1) / ciL)

### Functions

- void [mpi\\_init](#) ([mpi](#) \*X,...)  
*Initialize one or more [mpi](#).*
- void [mpi\\_free](#) ([mpi](#) \*X,...)  
*Unallocate one or more [mpi](#).*
- int [mpi\\_grow](#) ([mpi](#) \*X, int nblimbs)  
*Enlarge X to the specified number of limbs.*
- int [mpi\\_copy](#) ([mpi](#) \*X, [mpi](#) \*Y)  
*Copy the contents of Y into X.*
- void [mpi\\_swap](#) ([mpi](#) \*X, [mpi](#) \*Y)  
*Swap the contents of X and Y.*
- int [mpi\\_lset](#) ([mpi](#) \*X, int z)  
*Set value from integer.*
- static int [mpi\\_get\\_digit](#) (t\_int \*d, int radix, char c)
- int [mpi\\_read\\_string](#) ([mpi](#) \*X, int radix, char \*s)  
*Import X from an ASCII string.*
- static int [mpi\\_write\\_hlp](#) ([mpi](#) \*X, int radix, char \*\*p)
- int [mpi\\_write\\_string](#) ([mpi](#) \*X, int radix, char \*s, int \*slen)  
*Export X into an ASCII string.*
- int [mpi\\_read\\_file](#) ([mpi](#) \*X, int radix, FILE \*fin)  
*Read X from an opened file.*

- `int mpi_write_file` (`char *p`, `mpi *X`, `int radix`, `FILE *fout`)  
*Write X into an opened file, or stdout.*
- `int mpi_read_binary` (`mpi *X`, `unsigned char *buf`, `int buflen`)  
*Import X from unsigned binary data, big endian.*
- `int mpi_write_binary` (`mpi *X`, `unsigned char *buf`, `int *buflen`)  
*Export X into unsigned binary data, big endian.*
- `int mpi_msb` (`mpi *X`)  
*Return the total size in bits, without leading 0s.*
- `int mpi_lsb` (`mpi *X`)  
*Return the number of least significant bits.*
- `int mpi_shift_l` (`mpi *X`, `int count`)  
*Left-shift:  $X \ll = \text{count}$ .*
- `int mpi_shift_r` (`mpi *X`, `int count`)  
*Right-shift:  $X \gg = \text{count}$ .*
- `int mpi_cmp_abs` (`mpi *X`, `mpi *Y`)  
*Compare unsigned values.*
- `int mpi_cmp_mpi` (`mpi *X`, `mpi *Y`)  
*Compare signed values.*
- `int mpi_cmp_int` (`mpi *X`, `int z`)  
*Compare signed values.*
- `int mpi_add_abs` (`mpi *X`, `mpi *A`, `mpi *B`)  
*Unsigned addition:  $X = |A| + |B|$ .*
- `static void mpi_sub_hlp` (`int n`, `t_int *s`, `t_int *d`)
- `int mpi_sub_abs` (`mpi *X`, `mpi *A`, `mpi *B`)  
*Unsigned subtraction:  $X = |A| - |B|$ .*
- `int mpi_add_mpi` (`mpi *X`, `mpi *A`, `mpi *B`)  
*Signed addition:  $X = A + B$ .*
- `int mpi_sub_mpi` (`mpi *X`, `mpi *A`, `mpi *B`)  
*Signed subtraction:  $X = A - B$ .*
- `int mpi_add_int` (`mpi *X`, `mpi *A`, `int b`)  
*Signed addition:  $X = A + b$ .*
- `int mpi_sub_int` (`mpi *X`, `mpi *A`, `int b`)  
*Signed subtraction:  $X = A - b$ .*
- `static void mpi_mul_hlp` (`int i`, `t_int *s`, `t_int *d`, `t_int b`)

- `int mpi_mul_mpi (mpi *X, mpi *A, mpi *B)`  
*Baseline multiplication:  $X = A * B$ .*
- `int mpi_mul_int (mpi *X, mpi *A, t_int b)`  
*Baseline multiplication:  $X = A * b$ .*
- `int mpi_div_mpi (mpi *Q, mpi *R, mpi *A, mpi *B)`  
*Division by mpi:  $A = Q * B + R$ .*
- `int mpi_div_int (mpi *Q, mpi *R, mpi *A, int b)`  
*Division by int:  $A = Q * b + R$ .*
- `int mpi_mod_mpi (mpi *R, mpi *A, mpi *B)`  
*Modulo:  $R = A \bmod B$ .*
- `int mpi_mod_int (t_int *r, mpi *A, int b)`  
*Modulo:  $r = A \bmod b$ .*
- `static void mpi_montg_init (t_int *mm, mpi *N)`
- `static void mpi_montmul (mpi *A, mpi *B, mpi *N, t_int mm, mpi *T)`
- `static void mpi_montred (mpi *A, mpi *N, t_int mm, mpi *T)`
- `int mpi_exp_mod (mpi *X, mpi *A, mpi *E, mpi *N, mpi *_RR)`  
*Sliding-window exponentiation:  $X = A^E \bmod N$ .*
- `int mpi_gcd (mpi *G, mpi *A, mpi *B)`  
*Greatest common divisor:  $G = \gcd(A, B)$ .*
- `int mpi_inv_mod (mpi *X, mpi *A, mpi *N)`  
*Modular inverse:  $X = A^{-1} \bmod N$ .*
- `int mpi_is_prime (mpi *X)`  
*Miller-Rabin primality test.*
- `int mpi_gen_prime (mpi *X, int nbits, int dh_flag, int(*rng_f)(void *), void *rng_d)`  
*Prime number generation.*
- `int mpi_self_test (int verbose)`  
*Checkup routine.*

## Variables

- `static const int small_prime [ ]`
- `static const char _bignum_src [ ] = "_bignum_src"`

### 8.110.1 Define Documentation

#### 8.110.1.1 #define \_CRT\_SECURE\_NO\_DEPRECATED 1

Definition at line 31 of file bignum.c.

**8.110.1.2 #define BITS\_TO\_LIMBS(i) (((i) + biL - 1) / biL)**

Definition at line 45 of file bignum.c.

Referenced by mpi\_gen\_prime(), mpi\_read\_string(), and mpi\_shift\_l().

**8.110.1.3 #define CHARS\_TO\_LIMBS(i) (((i) + ciL - 1) / ciL)**

Definition at line 46 of file bignum.c.

Referenced by mpi\_read\_binary().

**8.110.2 Function Documentation****8.110.2.1 int mpi\_add\_abs (mpi \* X, mpi \* A, mpi \* B)**

Unsigned addition:  $X = |A| + |B|$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 644 of file bignum.c.

References CHK, mpi\_copy(), mpi\_grow(), mpi::n, and mpi::p.

Referenced by mpi\_add\_mpi(), and mpi\_sub\_mpi().

**8.110.2.2 int mpi\_add\_int (mpi \* X, mpi \* A, int b)**

Signed addition:  $X = A + b$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 809 of file bignum.c.

References mpi\_add\_mpi(), mpi::n, mpi::p, and mpi::s.

Referenced by mpi\_gen\_prime(), and mpi\_read\_string().

**8.110.2.3 int mpi\_add\_mpi (mpi \* X, mpi \* A, mpi \* B)**

Signed addition:  $X = A + B$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 747 of file bignum.c.

References CHK, mpi\_add\_abs(), mpi\_cmp\_abs(), mpi\_sub\_abs(), and mpi::s.

Referenced by mpi\_add\_int(), mpi\_div\_mpi(), mpi\_inv\_mod(), mpi\_mod\_mpi(), and rsa\_private().

**8.110.2.4 int mpi\_cmp\_abs (mpi \* X, mpi \* Y)**

Compare unsigned values.

**Returns:**

1 if  $|X|$  is greater than  $|Y|$ , -1 if  $|X|$  is lesser than  $|Y|$  or 0 if  $|X|$  is equal to  $|Y|$

Definition at line 565 of file bignum.c.

References `mpi::n`, and `mpi::p`.

Referenced by `mpi_add_mpi()`, `mpi_div_mpi()`, `mpi_montmul()`, `mpi_sub_abs()`, and `mpi_sub_mpi()`.

**8.110.2.5 int mpi\_cmp\_int (mpi \* X, int z)**

Compare signed values.

**Returns:**

1 if X is greater than z, -1 if X is lesser than z or 0 if X is equal to z

Definition at line 628 of file bignum.c.

References `mpi_cmp_mpi()`, `mpi::n`, `mpi::p`, and `mpi::s`.

Referenced by `mpi_div_mpi()`, `mpi_exp_mod()`, `mpi_gcd()`, `mpi_inv_mod()`, `mpi_is_prime()`, `mpi_mod_mpi()`, `mpi_write_hlp()`, `rsa_check_privkey()`, and `rsa_gen_key()`.

**8.110.2.6 int mpi\_cmp\_mpi (mpi \* X, mpi \* Y)**

Compare signed values.

**Returns:**

1 if X is greater than Y, -1 if X is lesser than Y or 0 if X is equal to Y

Definition at line 595 of file bignum.c.

References `mpi::n`, `mpi::p`, and `mpi::s`.

Referenced by `dhm_make_params()`, `dhm_make_public()`, `mpi_cmp_int()`, `mpi_div_mpi()`, `mpi_exp_mod()`, `mpi_gcd()`, `mpi_inv_mod()`, `mpi_is_prime()`, `mpi_mod_mpi()`, `rsa_check_privkey()`, `rsa_gen_key()`, `rsa_private()`, and `rsa_public()`.

**8.110.2.7 int mpi\_copy (mpi \* X, mpi \* Y)**

Copy the contents of Y into X.

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 118 of file bignum.c.

References `CHK`, `ciL`, `mpi_grow()`, `mpi::n`, `mpi::p`, and `mpi::s`.

Referenced by `mpi_add_abs()`, `mpi_div_mpi()`, `mpi_exp_mod()`, `mpi_gcd()`, `mpi_inv_mod()`, `mpi_is_prime()`, `mpi_mul_mpi()`, `mpi_sub_abs()`, and `mpi_write_string()`.



**8.110.2.8 int mpi\_div\_int (mpi \* *Q*, mpi \* *R*, mpi \* *A*, int *b*)**

Division by int:  $A = Q * b + R$ .

**Returns:**

0 if successful, 1 if memory allocation failed, ERR\_MPI\_DIVISION\_BY\_ZERO if  $b == 0$

**Note:**

Either *Q* or *R* can be NULL.

Definition at line 1120 of file bignum.c.

References mpi\_div\_mpi(), mpi::n, mpi::p, and mpi::s.

Referenced by main(), and mpi\_write\_hlp().

**8.110.2.9 int mpi\_div\_mpi (mpi \* *Q*, mpi \* *R*, mpi \* *A*, mpi \* *B*)**

Division by [mpi](#):  $A = Q * B + R$ .

**Returns:**

0 if successful, 1 if memory allocation failed, ERR\_MPI\_DIVISION\_BY\_ZERO if  $B == 0$

**Note:**

Either *Q* or *R* can be NULL.

Definition at line 956 of file bignum.c.

References biH, biL, CHK, ERR\_MPI\_DIVISION\_BY\_ZERO, mpi\_add\_mpi(), mpi\_cmp\_abs(), mpi\_cmp\_int(), mpi\_cmp\_mpi(), mpi\_copy(), mpi\_free(), mpi\_grow(), mpi\_init(), mpi\_lset(), mpi\_msb(), mpi\_mul\_int(), mpi\_shift\_l(), mpi\_shift\_r(), mpi\_sub\_mpi(), mpi::n, mpi::p, and mpi::s.

Referenced by mpi\_div\_int(), and mpi\_mod\_mpi().

**8.110.2.10 int mpi\_exp\_mod (mpi \* *X*, mpi \* *A*, mpi \* *E*, mpi \* *N*, mpi \* *\_RR*)**

Sliding-window exponentiation:  $X = A^E \bmod N$ .

**Returns:**

0 if successful, 1 if memory allocation failed, ERR\_MPI\_INVALID\_PARAMETER if *N* is negative or even

**Note:**

*\_RR* is used to avoid re-computing  $R * R \bmod N$  across multiple calls, which speeds up things a bit. It can be set to NULL if the extra performance is unneeded.

Definition at line 1266 of file bignum.c.

References biL, CHK, ERR\_MPI\_INVALID\_PARAMETER, mpi\_cmp\_int(), mpi\_cmp\_mpi(), mpi\_copy(), mpi\_free(), mpi\_grow(), mpi\_init(), mpi\_lset(), mpi\_mod\_mpi(), mpi\_montg\_init(), mpi\_montmul(), mpi\_montred(), mpi\_msb(), mpi\_shift\_l(), mpi::n, and mpi::p.

Referenced by dhm\_calc\_secret(), dhm\_make\_params(), dhm\_make\_public(), main(), mpi\_is\_prime(), rsa\_private(), and rsa\_public().

**8.110.2.11 void mpi\_free (mpi \* X, ...)**

Unallocate one or more [mpi](#).

Definition at line 69 of file bignum.c.

References [ciL](#), [mpi::n](#), and [mpi::p](#).

Referenced by [dhm\\_free\(\)](#), [main\(\)](#), [mpi\\_div\\_mpi\(\)](#), [mpi\\_exp\\_mod\(\)](#), [mpi\\_gcd\(\)](#), [mpi\\_gen\\_prime\(\)](#), [mpi\\_inv\\_mod\(\)](#), [mpi\\_is\\_prime\(\)](#), [mpi\\_mul\\_mpi\(\)](#), [mpi\\_read\\_string\(\)](#), [mpi\\_sub\\_abs\(\)](#), [mpi\\_write\\_string\(\)](#), [rsa\\_check\\_privkey\(\)](#), [rsa\\_free\(\)](#), [rsa\\_gen\\_key\(\)](#), [rsa\\_private\(\)](#), and [rsa\\_public\(\)](#).

**8.110.2.12 int mpi\_gcd (mpi \* G, mpi \* A, mpi \* B)**

Greatest common divisor:  $G = \gcd(A, B)$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 1443 of file bignum.c.

References [CHK](#), [mpi\\_cmp\\_int\(\)](#), [mpi\\_cmp\\_mpi\(\)](#), [mpi\\_copy\(\)](#), [mpi\\_free\(\)](#), [mpi\\_init\(\)](#), [mpi\\_lsb\(\)](#), [mpi\\_lset\(\)](#), [mpi\\_mul\\_mpi\(\)](#), [mpi\\_shift\\_l\(\)](#), [mpi\\_shift\\_r\(\)](#), [mpi\\_sub\\_abs\(\)](#), [mpi::p](#), and [mpi::s](#).

Referenced by [mpi\\_inv\\_mod\(\)](#), [rsa\\_check\\_privkey\(\)](#), and [rsa\\_gen\\_key\(\)](#).

**8.110.2.13 int mpi\_gen\_prime (mpi \* X, int nbits, int dh\_flag, int(\*) (void \*) rng\_f, void \* rng\_d)**

Prime number generation.

**Parameters:**

*X* destination [mpi](#)

*nbits* required size of X in bits

*dh\_flag* if 1, then (X-1)/2 will be prime too

*rng\_f* points to the RNG function

*rng\_d* points to the RNG data

**Returns:**

0 if successful (probably prime), 1 if memory allocation failed, [ERR\\_MPI\\_INVALID\\_PARAMETER](#) if nbits is < 3

Definition at line 1713 of file bignum.c.

References [BITS\\_TO\\_LIMBS](#), [CHK](#), [ERR\\_MPI\\_INVALID\\_PARAMETER](#), [ERR\\_MPI\\_NOT\\_ACCEPTABLE](#), [mpi\\_add\\_int\(\)](#), [mpi\\_free\(\)](#), [mpi\\_grow\(\)](#), [mpi\\_init\(\)](#), [mpi\\_is\\_prime\(\)](#), [mpi\\_lset\(\)](#), [mpi\\_msb\(\)](#), [mpi\\_shift\\_l\(\)](#), [mpi\\_shift\\_r\(\)](#), [mpi\\_sub\\_int\(\)](#), and [mpi::p](#).

Referenced by [main\(\)](#), and [rsa\\_gen\\_key\(\)](#).

**8.110.2.14 static int mpi\_get\_digit (t\_int \* d, int radix, char c) [static]**

Definition at line 174 of file bignum.c.

References ERR\_MPI\_INVALID\_CHARACTER.

Referenced by mpi\_read\_file(), and mpi\_read\_string().

#### 8.110.2.15 int mpi\_grow (mpi \* X, int nlimbs)

Enlarge X to the specified number of limbs.

##### Returns:

0 if successful, 1 if memory allocation failed

Definition at line 94 of file bignum.c.

References ciL, mpi::n, mpi::p, and mpi::s.

Referenced by dhm\_make\_params(), dhm\_make\_public(), mpi\_add\_abs(), mpi\_copy(), mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_gen\_prime(), mpi\_is\_prime(), mpi\_lset(), mpi\_mul\_mpi(), mpi\_read\_binary(), mpi\_read\_string(), and mpi\_shift\_l().

#### 8.110.2.16 void mpi\_init (mpi \* X, ...)

Initialize one or more [mpi](#).

Definition at line 51 of file bignum.c.

Referenced by main(), mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_gcd(), mpi\_gen\_prime(), mpi\_inv\_mod(), mpi\_is\_prime(), mpi\_mul\_mpi(), mpi\_read\_string(), mpi\_sub\_abs(), mpi\_write\_string(), rsa\_check\_privkey(), rsa\_gen\_key(), rsa\_private(), and rsa\_public().

#### 8.110.2.17 int mpi\_inv\_mod (mpi \* X, mpi \* A, mpi \* N)

Modular inverse:  $X = A^{-1} \bmod N$ .

##### Returns:

0 if successful, 1 if memory allocation failed, ERR\_MPI\_INVALID\_PARAMETER if N is negative or nil ERR\_MPI\_NOT\_ACCEPTABLE if A has no inverse mod N

Definition at line 1491 of file bignum.c.

References CHK, ERR\_MPI\_INVALID\_PARAMETER, ERR\_MPI\_NOT\_ACCEPTABLE, mpi\_add\_mpi(), mpi\_cmp\_int(), mpi\_cmp\_mpi(), mpi\_copy(), mpi\_free(), mpi\_gcd(), mpi\_init(), mpi\_lset(), mpi\_mod\_mpi(), mpi\_shift\_r(), mpi\_sub\_mpi(), and mpi::p.

Referenced by main(), and rsa\_gen\_key().

#### 8.110.2.18 int mpi\_is\_prime (mpi \* X)

Miller-Rabin primality test.

##### Returns:

0 if successful (probably prime), 1 if memory allocation failed, ERR\_MPI\_NOT\_ACCEPTABLE if X is not prime

Definition at line 1617 of file bignum.c.

References CHK, ERR\_MPI\_NOT\_ACCEPTABLE, mpi\_cmp\_int(), mpi\_cmp\_mpi(), mpi\_copy(), mpi\_exp\_mod(), mpi\_free(), mpi\_grow(), mpi\_init(), mpi\_lsb(), mpi\_mod\_int(), mpi\_mod\_mpi(), mpi\_msb(), mpi\_mul\_mpi(), mpi\_shift\_r(), mpi\_sub\_int(), mpi::n, mpi::p, R, mpi::s, and small\_prime.

Referenced by main(), and mpi\_gen\_prime().

#### 8.110.2.19 int mpi\_lsb (mpi \* X)

Return the number of least significant bits.

Definition at line 461 of file bignum.c.

References biL, int, mpi::n, and mpi::p.

Referenced by mpi\_gcd(), and mpi\_is\_prime().

#### 8.110.2.20 int mpi\_lset (mpi \* X, int z)

Set value from integer.

##### Returns:

0 if successful, 1 if memory allocation failed

Definition at line 157 of file bignum.c.

References CHK, ciL, mpi\_grow(), mpi::n, mpi::p, and mpi::s.

Referenced by mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_gcd(), mpi\_gen\_prime(), mpi\_inv\_mod(), mpi\_mul\_mpi(), mpi\_read\_binary(), mpi\_read\_string(), and rsa\_gen\_key().

#### 8.110.2.21 int mpi\_mod\_int (t\_int \* r, mpi \* A, int b)

Modulo:  $r = A \bmod b$ .

##### Returns:

0 if successful, 1 if memory allocation failed, ERR\_MPI\_DIVISION\_BY\_ZERO if  $b == 0$ ,

Definition at line 1156 of file bignum.c.

References biH, ERR\_MPI\_DIVISION\_BY\_ZERO, mpi::n, and mpi::p.

Referenced by mpi\_is\_prime(), and mpi\_write\_hlp().

#### 8.110.2.22 int mpi\_mod\_mpi (mpi \* R, mpi \* A, mpi \* B)

Modulo:  $R = A \bmod B$ .

##### Returns:

0 if successful, 1 if memory allocation failed, ERR\_MPI\_DIVISION\_BY\_ZERO if  $B == 0$

Definition at line 1136 of file bignum.c.

References CHK, mpi\_add\_mpi(), mpi\_cmp\_int(), mpi\_cmp\_mpi(), mpi\_div\_mpi(), and mpi\_sub\_mpi().

Referenced by mpi\_exp\_mod(), mpi\_inv\_mod(), mpi\_is\_prime(), rsa\_gen\_key(), and rsa\_private().

**8.110.2.23 static void mpi\_montg\_init (t\_int \* *mm*, mpi \* *N*)** [static]

Definition at line 1197 of file bignum.c.

References biL, and mpi::p.

Referenced by mpi\_exp\_mod().

**8.110.2.24 static void mpi\_montmul (mpi \* *A*, mpi \* *B*, mpi \* *N*, t\_int *mm*, mpi \* *T*)** [static]

Definition at line 1215 of file bignum.c.

References ciL, mpi\_cmp\_abs(), mpi\_mul\_hlp(), mpi\_sub\_hlp(), mpi::n, and mpi::p.

Referenced by mpi\_exp\_mod(), and mpi\_montred().

**8.110.2.25 static void mpi\_montred (mpi \* *A*, mpi \* *N*, t\_int *mm*, mpi \* *T*)** [static]

Definition at line 1252 of file bignum.c.

References mpi\_montmul(), mpi::n, mpi::p, and mpi::s.

Referenced by mpi\_exp\_mod().

**8.110.2.26 int mpi\_msb (mpi \* *X*)**

Return the total size in bits, without leading 0s.

Definition at line 443 of file bignum.c.

References biL, mpi::n, and mpi::p.

Referenced by dhm\_calc\_secret(), dhm\_make\_params(), dhm\_read\_params(), mpi\_div\_mpi(), mpi\_exp\_mod(), mpi\_gen\_prime(), mpi\_is\_prime(), mpi\_shift\_l(), mpi\_write\_binary(), mpi\_write\_string(), rsa\_check\_pubkey(), rsa\_gen\_key(), rsa\_read\_private(), rsa\_read\_public(), x509\_add\_certs(), and x509\_parse\_key().

**8.110.2.27 static void mpi\_mul\_hlp (int *i*, t\_int \* *s*, t\_int \* *d*, t\_int *b*)** [static]

Definition at line 841 of file bignum.c.

References MULADDC\_CORE, MULADDC\_INIT, and MULADDC\_STOP.

Referenced by mpi\_montmul(), and mpi\_mul\_mpi().

**8.110.2.28 int mpi\_mul\_int (mpi \* *X*, mpi \* *A*, t\_int *b*)**

Baseline multiplication:  $X = A * b$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 940 of file bignum.c.

References mpi\_mul\_mpi(), mpi::n, mpi::p, and mpi::s.

Referenced by mpi\_div\_mpi(), and mpi\_read\_string().

**8.110.2.29 int mpi\_mul\_mpi (mpi \* X, mpi \* A, mpi \* B)**

Baseline multiplication:  $X = A * B$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 905 of file bignum.c.

References CHK, mpi\_copy(), mpi\_free(), mpi\_grow(), mpi\_init(), mpi\_lset(), mpi\_mul\_hlp(), mpi::n, mpi::p, and mpi::s.

Referenced by main(), mpi\_gcd(), mpi\_is\_prime(), mpi\_mul\_int(), rsa\_check\_privkey(), rsa\_gen\_key(), and rsa\_private().

**8.110.2.30 int mpi\_read\_binary (mpi \* X, unsigned char \* buf, int buflen)**

Import X from unsigned binary data, big endian.

**Parameters:**

*X* destination [mpi](#)

*buf* input buffer

*buflen* input buffer size

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 398 of file bignum.c.

References CHARS\_TO\_LIMBS, CHK, ciL, mpi\_grow(), mpi\_lset(), and mpi::p.

Referenced by asn1\_get\_mpi(), dhm\_read\_bignum(), dhm\_read\_public(), rsa\_private(), and rsa\_public().

**8.110.2.31 int mpi\_read\_file (mpi \* X, int radix, FILE \* fin)**

Read X from an opened file.

**Parameters:**

*X* destination [mpi](#)

*radix* input numeric base

*fin* input file handle

**Returns:**

0 if successful, or an ERR\_MPI\_XXX error code

Definition at line 335 of file bignum.c.

References ERR\_MPI\_FILE\_IO\_ERROR, mpi\_get\_digit(), and mpi\_read\_string().

Referenced by main(), rsa\_read\_private(), and rsa\_read\_public().

**8.110.2.32 int mpi\_read\_string (mpi \* X, int radix, char \* s)**

Import X from an ASCII string.

**Parameters:**

*X* destination [mpi](#)  
*radix* input numeric base  
*s* null-terminated string buffer

**Returns:**

0 if successful, or an ERR\_MPI\_XXX error code

Definition at line 191 of file bignum.c.

References [BITS\\_TO\\_LIMBS](#), [CHK](#), [ciL](#), [ERR\\_MPI\\_INVALID\\_PARAMETER](#), [int](#), [mpi\\_add\\_int\(\)](#), [mpi\\_free\(\)](#), [mpi\\_get\\_digit\(\)](#), [mpi\\_grow\(\)](#), [mpi\\_init\(\)](#), [mpi\\_lset\(\)](#), [mpi\\_mul\\_int\(\)](#), [mpi::p](#), and [mpi::s](#).

Referenced by [main\(\)](#), and [mpi\\_read\\_file\(\)](#).

**8.110.2.33 int mpi\_self\_test (int verbose)**

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 1915 of file bignum.c.

Referenced by [main\(\)](#).

**8.110.2.34 int mpi\_shift\_l (mpi \* X, int count)**

Left-shift:  $X \ll= \text{count}$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 476 of file bignum.c.

References [biL](#), [BITS\\_TO\\_LIMBS](#), [CHK](#), [mpi\\_grow\(\)](#), [mpi\\_msb\(\)](#), [mpi::n](#), and [mpi::p](#).

Referenced by [mpi\\_div\\_mpi\(\)](#), [mpi\\_exp\\_mod\(\)](#), [mpi\\_gcd\(\)](#), and [mpi\\_gen\\_prime\(\)](#).

**8.110.2.35 int mpi\_shift\_r (mpi \* X, int count)**

Right-shift:  $X \gg= \text{count}$ .

**Returns:**

0 if successful, 1 if memory allocation failed

Definition at line 525 of file bignum.c.

References `biL`, `mpi::n`, and `mpi::p`.

Referenced by `dhm_make_params()`, `dhm_make_public()`, `mpi_div_mpi()`, `mpi_gcd()`, `mpi_gen_prime()`, `mpi_inv_mod()`, and `mpi_is_prime()`.

#### 8.110.2.36 `int mpi_sub_abs (mpi * X, mpi * A, mpi * B)`

Unsigned subtraction:  $X = |A| - |B|$ .

##### Returns:

0 if successful, `ERR_MPI_NEGATIVE_VALUE` if B is greater than A

Definition at line 711 of file bignum.c.

References `CHK`, `ERR_MPI_NEGATIVE_VALUE`, `mpi_cmp_abs()`, `mpi_copy()`, `mpi_free()`, `mpi_init()`, `mpi_sub_hlp()`, `mpi::n`, and `mpi::p`.

Referenced by `mpi_add_mpi()`, `mpi_gcd()`, and `mpi_sub_mpi()`.

#### 8.110.2.37 `static void mpi_sub_hlp (int n, t_int * s, t_int * d)` `[static]`

Definition at line 690 of file bignum.c.

Referenced by `mpi_montmul()`, and `mpi_sub_abs()`.

#### 8.110.2.38 `int mpi_sub_int (mpi * X, mpi * A, int b)`

Signed subtraction:  $X = A - b$ .

##### Returns:

0 if successful, 1 if memory allocation failed

Definition at line 825 of file bignum.c.

References `mpi_sub_mpi()`, `mpi::n`, `mpi::p`, and `mpi::s`.

Referenced by `main()`, `mpi_gen_prime()`, `mpi_is_prime()`, `rsa_check_privkey()`, and `rsa_gen_key()`.

#### 8.110.2.39 `int mpi_sub_mpi (mpi * X, mpi * A, mpi * B)`

Signed subtraction:  $X = A - B$ .

##### Returns:

0 if successful, 1 if memory allocation failed

Definition at line 778 of file bignum.c.

References `CHK`, `mpi_add_abs()`, `mpi_cmp_abs()`, `mpi_sub_abs()`, and `mpi::s`.

Referenced by `mpi_div_mpi()`, `mpi_inv_mod()`, `mpi_mod_mpi()`, `mpi_sub_int()`, and `rsa_private()`.



**8.110.2.40 void mpi\_swap (mpi \* *X*, mpi \* *Y*)**

Swap the contents of *X* and *Y*.

Definition at line 145 of file bignum.c.

Referenced by rsa\_gen\_key().

**8.110.2.41 int mpi\_write\_binary (mpi \* *X*, unsigned char \* *buf*, int \* *buflen*)**

Export *X* into unsigned binary data, big endian.

**Parameters:**

*X* source [mpi](#)

*buf* output buffer

*buflen* output buffer size

**Returns:**

0 if successful, ERR\_MPI\_BUFFER\_TOO\_SMALL if *buf* isn't large enough

**Note:**

Call this function with \**buflen* = 0 to obtain the minimum required buffer size in \**buflen*.

Definition at line 420 of file bignum.c.

References ciL, ERR\_MPI\_BUFFER\_TOO\_SMALL, mpi\_msb(), and mpi::p.

Referenced by dhm\_calc\_secret(), dhm\_make\_public(), rsa\_private(), and rsa\_public().

**8.110.2.42 int mpi\_write\_file (char \* *p*, mpi \* *X*, int *radix*, FILE \* *fout*)**

Write *X* into an opened file, or stdout.

**Parameters:**

*p* prefix, can be NULL

*X* source [mpi](#)

*radix* output numeric base

*fout* output file handle

**Returns:**

0 if successful, or an ERR\_MPI\_XXX error code

**Note:**

Set *fout* == NULL to print *X* on the console.

Definition at line 360 of file bignum.c.

References CHK, ERR\_MPI\_FILE\_IO\_ERROR, and mpi\_write\_string().

Referenced by main(), rsa\_write\_private(), and rsa\_write\_public().

**8.110.2.43** `static int mpi_write_hlp (mpi * X, int radix, char ** p)` `[static]`

Definition at line 248 of file `bignum.c`.

References `CHK`, `ERR_MPI_INVALID_PARAMETER`, `mpi_cmp_int()`, `mpi_div_int()`, and `mpi_mod_int()`.

Referenced by `mpi_write_string()`.

**8.110.2.44** `int mpi_write_string (mpi * X, int radix, char * s, int * slen)`

Export *X* into an ASCII string.

**Parameters:**

*X* source [mpi](#)

*radix* output numeric base

*s* string buffer

*slen* string buffer size

**Returns:**

0 if successful, or an `ERR_MPI_XXX` error code

**Note:**

Call this function with `*slen = 0` to obtain the minimum required buffer size in `*slen`.

Definition at line 273 of file `bignum.c`.

References `CHK`, `ciL`, `ERR_MPI_BUFFER_TOO_SMALL`, `ERR_MPI_INVALID_PARAMETER`, `mpi_copy()`, `mpi_free()`, `mpi_init()`, `mpi_msb()`, `mpi_write_hlp()`, `mpi::n`, `mpi::p`, and `mpi::s`.

Referenced by `mpi_write_file()`.

**8.110.3 Variable Documentation****8.110.3.1** `const char _bignum_src[] = "_bignum_src"` `[static]`

Definition at line 1781 of file `bignum.c`.

**8.110.3.2** `const int small_prime[]` `[static]`

Definition at line 1582 of file `bignum.c`.

Referenced by `mpi_is_prime()`.

## 8.111 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/certs.c File Reference

### Variables

- char `test_ca_crt` []
- char `test_ca_key` []
- char `test_ca_pwd` [] = "test"
- char `test_srv_crt` []
- char `test_srv_key` []
- char `test_cli_crt` []
- char `test_cli_key` []
- char `xyssl_ca_crt` []
- static const char `_certs_src` [] = "\_certs\_src"

### 8.111.1 Variable Documentation

**8.111.1.1** `const char _certs_src[] = "_certs_src"` [static]

Definition at line 209 of file certs.c.

**8.111.1.2** `char test_ca_crt[]`

**Initial value:**

```
"-----BEGIN CERTIFICATE-----\r\n"
"MIIDpTCCAo2gAwIBAgIBADANBgkqhkiG9w0BAQUFADBFMQswCQYDVQQGEwJGUjEO\r\n"
"MAwGA1UEBxMFUGFyaXNkZjAMBgNVBAoTBVh5U1NMMRYwFAYDVQQDEw1YeVNTTCBU\r\n"
"ZXN0IENBMB4XDTA3MDcwNzA1MDAxOFoXDTE3MDcwNzA1MDAxOFowRTElMAkGA1UE\r\n"
"BhMCRlIxZjAMBgNVBAcTBVBhcm1zMQ4wDAYDVQQKEwVYeVNTTDEWMBQGA1UEAxMN\r\n"
"WH1TU0wgVGZzdCBDQTCASiWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAM+k\r\n"
"gt70fIPiYqmvXr+9uPmWoN405eoSzxdiRLCHqL4V/Ts0E/H+JNfHS4DH1AgxrJu\r\n"
"+ZIadvSJuhKI6eliMkAh5SU1DqaF3jrrFdJCooM6a077M4CRkE1tdAeZDf+BYp0q\r\n"
"BeMU9Y2+j7ibsQaPAizunbXLf4QdteExCwhYRJ8OVXSEaSnt339gJzTD6kOhES3b\r\n"
"lEN3qbx6lqFaJ5MLHTix5uNVc2rvbOizV5oLhJNqm52AKOp1ltv6WTiI8loagvAc\r\n"
"jlhEZRNB9el5SL6Jai/uFcqXKzfXNKW3FYFQHFgobmiMfGt1lUSBj3F2mrqEC7gC\r\n"
"whY/FDvAI64/k5LZAFkCAwEAAoBnzCBnDAMBgNVHRMEBTADAQH/MB0GA1UdDgQW\r\n"
"BBS87h+Y6Porg+SkfV7DdXKTMdKyZzBtBgNVHSMEZjBkgBS87h+Y6Porg+SkfV7D\r\n"
"dXKTMdKyZ6FJpEcwRTElMAkGA1UEBhMCRlIxZjAMBgNVBAcTBVBhcm1zMQ4wDAYD\r\n"
"VQQKEwVYeVNTTDEWMBQGA1UEAxMNWH1TU0wgVGZzdCBDQYIBADANBgkqhkiG9w0B\r\n"
"QAQUFAAOCAQEAIHdohONCg6KAhWdSmfEgSbKUI8/Zr/d56uw42H00sj/uKPQzUco\r\n"
"3Mx2BYElm1itg7q5OhrkB7J4ZB78EtNZM84nV+y6od3YpR0Z9VUxCx7948MozYRy\r\n"
"TKF5x/lKHXlPJkFE04clKdWTFAtWtGhewXrHJQ8C+ENh2Up2wTVh3Z+pEzuZNv3\r\n"
"u/JYu1H+vkt3l1WCy/9mxUnu+anW1DzxPwnjy4lx6Mi0BD2qfKBWLjVS+7v6ALcj\r\n"
"S2oRWWr4LUvXT7z9BBAvw2eJQD+a4uAya6EURG7AsAvr5MnWn/r0wLWmBJ6fB1Yp\r\n"
"FlkOmamOFvstLMf74rLX+LGKeJ/nwui5FQ==\r\n"
"-----END CERTIFICATE-----\r\n"
```

Definition at line 24 of file certs.c.

Referenced by `main()`.

**8.111.1.3** `char test_ca_key[]`

Definition at line 48 of file certs.c.



```
"klMesqK3RQzDGwauDw431LQw0R69onn9fCM3wJw2yEC6wC9sF8I8hsNZbt64yZhZ\r\n"
"4Bi2YRTiHhpEuBqKlhHLDHFHneo3SMYh8PU/PDQQcyWGHUUi9z1RE\r\n"
"-----END RSA PRIVATE KEY-----\r\n"
```

Definition at line 155 of file certs.c.

Referenced by main().

### 8.111.1.7 char test\_srv\_crt[]

**Initial value:**

```
"-----BEGIN CERTIFICATE-----\r\n"
"MIIDPjCCAiaGAwIBAgIBAJANBgkqhkiG9w0BAQUFADBFMQswCQYDVQQGEwJGUjE0\r\n"
"MAwGA1UEBxMFUGFyaXNzA1MDEyOVVh5U1NMMRYwFAYDVQQDEw1YeVNTTCBU\r\n"
"ZXN0IENBMB4XDTA3MDcwNzA1MDEyOVVh5U1NMMRYwFAYDVQQDEw1YeVNTTCBU\r\n"
"BhMCRlIXdJAMBGNVBAOTBVh5U1NMMRIwEAYDVQQDEw1sb2NhbgHvc3QwgGFiMA0G\r\n"
"CSqGSIB3DQEBAAUAA4IBDwAwggEKAoIBAQC40PDcGTgmHkt6noXDfkjVuymjiNYB\r\n"
"gjtiL7uA1Ke3tXStacEecQek/OJxYqYr7ffcWals29LL6HbKpi0xLZKBbD9ACkDh\r\n"
"1Z/SvHlyQPILJdYb9DMw+kzZds5myXUjzn7Aem1YjoxMZUAMyc34i2900X2pL0v2\r\n"
"SfCeJ9Ym4MOnZxYl217+dX9ZbkgIgrT6uY2IYK4boDwxbTcyT8i/NPsvsiMwtWPM\r\n"
"rnQMr+XbgS98sUzcZE70Pe1TlV9Iy8j/8d2OiFo+qTyMu/6UpM2s3gdkQkMzx+Sm\r\n"
"4QitRUjzmEXeUePRUjEgHiv7vz069xuvBzrks36w5BXiVAhLke/OTKVPagMBAAGj\r\n"
"TTBLMAkGA1UdEwQCAAwHQYDVVR0OBBYEFNkOyCTx64SDdPySGWl/tzD7/WMSMB8G\r\n"
"A1UdIwQYMBAAFLzuH5jo+iuD5KR9XsN1cpMx2TJnMA0GCSqGSIB3DQEBBQUAA4IB\r\n"
"AQBelJv5t+suaqy5Lo5bjNeHjNZfgg8EigDQ7Nqaosv1QZAsh2N34Gg5YdkGyVdg\r\n"
"s32I/K5aaywyUbG9qVXQxCM2T95qBqyK56h9yJoZKWQD9H//+zB8kCK/16WvRfv3\r\n"
"VA7eSR19qOFWlHe+1qGh2YhxeDUfyi+fm4D36dGxqc2A34tzjo0QPHKtIeqM0kJy\r\n"
"zzL65T1bJQKkyTurHofFv0jW9ZFG2wkGysVgCY5fjuLi1do/sWUaX2987iNFa+K\r\n"
"FrHsTi6urSfZuG1ZnXDXDHEE7Q2snAvvev+KR7DD9X4DJGcPX9ga4CGJj+9ZzyAA\r\n"
"ZTGpOzk1hIH44RFs21JmZRL\r\n"
"-----END CERTIFICATE-----\r\n"
```

Definition at line 82 of file certs.c.

Referenced by main().

### 8.111.1.8 char test\_srv\_key[]

**Initial value:**

```
"-----BEGIN RSA PRIVATE KEY-----\r\n"
"MIIEowIBAAKCAQEAuNDw3Bk4Jh5Lep6Fw35I1lbsp04jWAYI7Yi+7gNSnt7V0rWnB\r\n"
"HnEHpPzicwKmk+333FmpUtvSy+h2yqYtMS2SgWw/QApA4dWf0rx5ckDyCyXWG/Qz\r\n"
"MPpM2XbOZs11I85+wHptWI6MTGVADmnN+ItvdNF9qS9L9knwniFWJuDDp2cWJdte\r\n"
"/nV/WW5ICIK0+rmNiGcuG6A8MW03Mk/IvzT7FbIjMLVjzK50DK/124EvfLFM3GRO\r\n"
"9D3tU5VfSMvI//HdjohaPqk8jLv+1KTNrN4HZEJDM8fkpuEIrUVI85hf31Hj0ViX\r\n"
"IByL+789OvcblQc65LN+sOQV41QIS5HvzkylTwIDAQABAoIBABeah8h0aBlmMRmd\r\n"
"+vN4Y3D4kf7UcRCMQ21Mz1Oq1Si/QgGLyiBLK0DFE16LzNE7eTZpNRjh/lAQhmt\r\n"
"QcpQGa/x1TomlRbCo8DUVWZkKQWHdYroa0lMDliPtDimzhEepE2M1T5EJmLzY3S+\r\n"
"qVge7UMsJjJfWgJAezyXteANQK+2YSt+CjPIqIHch1KexUnvdN9++1oEx6AbuZ8T\r\n"
"4avhFYZQP15tZNGsk2LfQ1YS/NfbowkCsd0/TVubJBmDGUML/E5MbxjxLz1aNB2M\r\n"
"V59cBNsgsA35CODAUF4xOyoSfZGg1Rb9qQrv1E6Jz56dG8SsKF3HqnDjxiPOVBN\r\n"
"FBnVJ+ECgYEA29MhAsKMM4XqBUPk6pIMFTgm/s1E5vxiG70vqIiL+guvBhhQ7zs1\r\n"
"8UMTNXZoMELNoB/ev9fN0CjclVr46b/x/yDw7wMb96i+vzEN0zu4RHwi3OWpCPbp\r\n"
"qBKEi3hzN8M+BulPX8CDQx3aLRRfxw51J5EuA0NeybngbItgxTi0u6kCgYEA1zr0\r\n"
"6P5YdOhYHtSWDlkeD49MApcVuzahnsHZVAhUqu3Rwi9LRAJLzfr7fQDb9DJBzP\r\n"
"sxTRLG6LSAcSR7mw+m+GvNqGt/9pSqbtW+L/VwVWSyF+YYklxZUD3UAAyrdVcDEC\r\n"
"a5S+ja4dCsi/1Vht5ulWiCkWL1fJvadmSubKNDCcGYA+71xVGPP+1sFgTiytfrC8\r\n"
"5n2rl4MxinJ9+w0I+EbZCKNMYGvTgiU4dJasSMEdiBKs1FMGo7dF8F0BLHF1IsIa\r\n"
"5Ah2tXtXn9154o90iTXMMK6qmRaneM6fhOoeaCwYAhpGxYIpqx/Xr4TOhiag46\r\n"
"-----"
```

```
"jMMaphAeOvw4t1K2RDziOQKBgQCYPCCU0gxuw/o1jda2CxbZy9EmU/erEX09+0n+\r\n"
"TOFQpSEFq/z9WaxAFY9LfsdZ0ZktoeHmalbNdL3i6A3DWAM3YSQzQMRPmzOWnqXx\r\n"
"cgoCBmlvzkzaeLjO5phMoLQHJmmafvuCG6uxov3F8Hi3LyHUF2c8k0nL6ucmJ3vj\r\n"
"uzu4AQKBgBSASMAJS63M9UJB1Eazy2v2NWw04CmzNxUfWrHuKpd/C2ik4QKu0sRO\r\n"
"r9KnkDgxxEhjDm7lXhlW12PU42yORst5I3Eaa1Cfi4KPFn/ozt+iNBYrzd8Tyvnb\r\n"
"qkdEC10+G2Fo/ER4NRCv7a24WNEsOMGzGRqw5cnSJrjbZLYMaIyK\r\n"
"-----END RSA PRIVATE KEY-----\r\n"
```

Definition at line 104 of file certs.c.

Referenced by main().

### 8.111.1.9 char xyssl\_ca\_crt[]

**Initial value:**

```
"-----BEGIN CERTIFICATE-----\r\n"
"MIID4DCCAsigAwIBAgIJAOLw9BMV1jxMMA0GCSqGSIb3DQEBBQUAMFMxCzAJBgNV\r\n"
"BAYTAKZSMQ4wDAYDVQQIEwVQYXJpczEOMAwGA1UEChMFWH1TU0wxJDAiBgNVBAMT\r\n"
"G1h5U1NMIENlcnRpZmljYXRlIEF1dGhvcm10eTAeFw0wNjEwMzEyMjU5MjRaFw0x\r\n"
"NjEwMzEyMjU5MjRaMFEMxCzAJBgNVBAYTAKZSMQ4wDAYDVQQIEwVQYXJpczEOMAwG\r\n"
"A1UEChMFWH1TU0wxJDAiBgNVBAMTG1h5U1NMIENlcnRpZmljYXRlIEF1dGhvcm10\r\n"
"eTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAKnprf1RQ7IYPI3FmI/h\r\n"
"f2EJGfaIP+Jt551VZWFrS3A56Nn4KS57zTAKZUA7YGBLwLAznfaphJ7SvENALeZR\r\n"
"/J0c/n9jwMpFXReQL7RVpgg/zlR+t2DUi3DAwigPZiHHCSJSBC73vpMc6uH0eV2d\r\n"
"itqjjUnJG5F8Zg9/gX4UMRAdlwGWqxvs+jc9i0XFKOEzga8+rONe6WvKyBM4e20I\r\n"
"HcO4BPF92d6sm4qLgyR4oXUkBz6NfDWX8ZdTvXuRaK9qMy1327cCT48sis9F6/eK\r\n"
"QAxx0VHlqGTtxDYjHJPsmLNejuyUDVsaC8TMCSpFTXpMTUvJdmxsF1LZWmIo5lY5\r\n"
"zh0CAwEAaAObtjCBsZAdBgNVHQ4EFgQUb1VzoNJrkkGPKJ8xjuFvVP2E4GswgYMG\r\n"
"A1UdIwR8MHQAFAZVc6DSa5JID5CfMY7hb1T9hOBroVekVTBTMQswCQYDVQQGEwJG\r\n"
"UjEOMAwGA1UEChMFUjEOMAwGA1UEChMFUjEOMAwGA1UEChMFUjEOMAwGA1UEChMF\r\n"
"TCBDZXJ0aWZpY2F0ZSBBDXRob3JpdHmCCQDi8PQTFdY8TDAMBGNVHRMEBTADAQH\r\n"
"MA0GCSqGSIb3DQEBBQUAA4IBAQCgd65b2l5BASfsPv1rcRnLzu/99eWTVAJwJbbD\r\n"
"VhPAQiET0W4U/85EDK7uoFo/SEjyMB/m4T20A8FIDaK7jBPo/1gtbuQjGMR17h+z\r\n"
"F2iGuNhZ6T2d6Uzqc1t3oiFtSvDRoZ/9kqkEy7Lrs7FBzOmvfTvrqvAdf7cLma2D\r\n"
"ri/otDpPr4XoDnwd4C+4bQC/Gr3Uder4VAeTOJtKdGqfYLVpWpSPVBDuVLUybKi\r\n"
"8cMAT6p9IG1e12u6vFqcBT/I67Q0bGU6gzVVz9ZVULXOYZMjjLafVXC1gesUH2WT\r\n"
"gTEAnEBkSRrkfAi+RezoEFAbmEl3fPt09dwSPku3x7cB3zaJ\r\n"
"-----END CERTIFICATE-----\r\n"
```

Definition at line 184 of file certs.c.

Referenced by main().

## 8.112 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/des.c File Reference

```
#include <string.h>
#include "xyssl/des.h"
```

### Defines

- #define [\\_CRT\\_SECURE\\_NO\\_DEPRECATED](#) 1
- #define [GET\\_UINT32\\_BE](#)(n, b, i)
- #define [PUT\\_UINT32\\_BE](#)(n, b, i)
- #define [DES\\_IP](#)(X, Y)
- #define [DES\\_FP](#)(X, Y)
- #define [DES\\_ROUND](#)(X, Y)

### Functions

- static void [des\\_main\\_k](#)s (unsigned long SK[32], unsigned char key[8])
- void [des\\_set\\_key](#) ([des\\_context](#) \*ctx, unsigned char key[8])  
*DES key schedule (56-bit).*
- static void [des\\_crypt](#) (unsigned long SK[32], unsigned char input[8], unsigned char output[8])
- void [des\\_encrypt](#) ([des\\_context](#) \*ctx, unsigned char input[8], unsigned char output[8])  
*DES block encryption (ECB mode).*
- void [des\\_decrypt](#) ([des\\_context](#) \*ctx, unsigned char input[8], unsigned char output[8])  
*DES block decryption (ECB mode).*
- void [des\\_cbc\\_encrypt](#) ([des\\_context](#) \*ctx, unsigned char iv[8], unsigned char \*input, unsigned char \*output, int len)  
*DES-CBC buffer encryption.*
- void [des\\_cbc\\_decrypt](#) ([des\\_context](#) \*ctx, unsigned char iv[8], unsigned char \*input, unsigned char \*output, int len)  
*DES-CBC buffer decryption.*
- void [des3\\_set\\_2keys](#) ([des3\\_context](#) \*ctx, unsigned char key[16])  
*Triple-DES key schedule (112-bit).*
- void [des3\\_set\\_3keys](#) ([des3\\_context](#) \*ctx, unsigned char key[24])  
*Triple-DES key schedule (168-bit).*
- static void [des3\\_crypt](#) (unsigned long SK[96], unsigned char input[8], unsigned char output[8])
- void [des3\\_encrypt](#) ([des3\\_context](#) \*ctx, unsigned char input[8], unsigned char output[8])  
*Triple-DES block encryption (ECB mode).*
- void [des3\\_decrypt](#) ([des3\\_context](#) \*ctx, unsigned char input[8], unsigned char output[8])  
*Triple-DES block decryption (ECB mode).*

- void `des3_cbc_encrypt` (`des3_context` \*ctx, unsigned char iv[8], unsigned char \*input, unsigned char \*output, int len)  
*3DES-CBC buffer encryption*
- void `des3_cbc_decrypt` (`des3_context` \*ctx, unsigned char iv[8], unsigned char \*input, unsigned char \*output, int len)  
*3DES-CBC buffer decryption*
- int `des_self_test` (int verbose)

## Variables

- static const unsigned long `SB1` [64]
- static const unsigned long `SB2` [64]
- static const unsigned long `SB3` [64]
- static const unsigned long `SB4` [64]
- static const unsigned long `SB5` [64]
- static const unsigned long `SB6` [64]
- static const unsigned long `SB7` [64]
- static const unsigned long `SB8` [64]
- static const unsigned long `LHs` [16]
- static const unsigned long `RHs` [16]
- static const char `_des_src` [] = "\_des\_src"

## 8.112.1 Define Documentation

### 8.112.1.1 #define \_CRT\_SECURE\_NO\_DEPRECATED 1

Definition at line 31 of file des.c.

### 8.112.1.2 #define DES\_FP(X, Y)

Value:

```
{
    X = ((X << 31) | (X >> 1)) & 0xFFFFFFFF; \
    T = (X ^ Y) & 0xAAAAAAAA; X ^= T; Y ^= T; \
    Y = ((Y << 31) | (Y >> 1)) & 0xFFFFFFFF; \
    T = ((Y >> 8) ^ X) & 0x00FF00FF; X ^= T; Y ^= (T << 8); \
    T = ((Y >> 2) ^ X) & 0x33333333; X ^= T; Y ^= (T << 2); \
    T = ((X >> 16) ^ Y) & 0x0000FFFF; Y ^= T; X ^= (T << 16); \
    T = ((X >> 4) ^ Y) & 0xF0F0F0F; Y ^= T; X ^= (T << 4); \
}
```

Definition at line 259 of file des.c.

Referenced by `des3_crypt()`, and `des_crypt()`.



**8.112.1.3 #define DES\_IP(X, Y)****Value:**

```

{
    T = ((X >> 4) ^ Y) & 0x0F0F0F0F; Y ^= T; X ^= (T << 4); \
    T = ((X >> 16) ^ Y) & 0x0000FFFF; Y ^= T; X ^= (T << 16); \
    T = ((Y >> 2) ^ X) & 0x33333333; X ^= T; Y ^= (T << 2); \
    T = ((Y >> 8) ^ X) & 0x00FF00FF; X ^= T; Y ^= (T << 8); \
    Y = ((Y << 1) | (Y >> 31)) & 0xFFFFFFFF; \
    T = (X ^ Y) & 0xAAAAAAAA; Y ^= T; X ^= T; \
    X = ((X << 1) | (X >> 31)) & 0xFFFFFFFF; \
}

```

Definition at line 245 of file des.c.

Referenced by des3\_crypt(), and des\_crypt().

**8.112.1.4 #define DES\_ROUND(X, Y)****Value:**

```

{
    T = *SK++ ^ X; \
    Y ^= SB8[ (T >> 8) & 0x3F ] ^ \
    SB6[ (T >> 16) & 0x3F ] ^ \
    SB4[ (T >> 24) & 0x3F ] ^ \
    SB2[ (T >> 31) & 0x3F ]; \
    T = *SK++ ^ ((X << 28) | (X >> 4)); \
    Y ^= SB7[ (T >> 8) & 0x3F ] ^ \
    SB5[ (T >> 16) & 0x3F ] ^ \
    SB3[ (T >> 24) & 0x3F ] ^ \
    SB1[ (T >> 31) & 0x3F ]; \
}

```

Definition at line 273 of file des.c.

Referenced by des3\_crypt(), and des\_crypt().

**8.112.1.5 #define GET\_UINT32\_BE(n, b, i)****Value:**

```

{
    (n) = ( (unsigned long) (b)[(i) << 4] << 24 ) \
    | ( (unsigned long) (b)[(i) << 4 + 1] << 16 ) \
    | ( (unsigned long) (b)[(i) << 4 + 2] << 8 ) \
    | ( (unsigned long) (b)[(i) << 4 + 3] ); \
}

```

Definition at line 42 of file des.c.

**8.112.1.6 #define PUT\_UINT32\_BE(n, b, i)****Value:**

```

{
    (b) [(i)      ] = (unsigned char) ( (n) >> 24 ); \
    (b) [(i) + 1] = (unsigned char) ( (n) >> 16 ); \
    (b) [(i) + 2] = (unsigned char) ( (n) >> 8  ); \
    (b) [(i) + 3] = (unsigned char) ( (n)      ); \
}

```

Definition at line 51 of file des.c.

## 8.112.2 Function Documentation

### 8.112.2.1 void des3\_cbc\_decrypt (des3\_context \* *ctx*, unsigned char *iv*[8], unsigned char \* *input*, unsigned char \* *output*, int *len*)

3DES-CBC buffer decryption

#### Parameters:

*ctx* 3DES context  
*iv* initialization vector (modified after use)  
*input* buffer holding the ciphertext  
*output* buffer holding the plaintext  
*len* length of the data to be decrypted

Definition at line 614 of file des.c.

References des3\_crypt(), and des3\_context::dsk.

Referenced by ssl\_decrypt\_buf(), and x509\_des3\_decrypt().

### 8.112.2.2 void des3\_cbc\_encrypt (des3\_context \* *ctx*, unsigned char *iv*[8], unsigned char \* *input*, unsigned char \* *output*, int *len*)

3DES-CBC buffer encryption

#### Parameters:

*ctx* 3DES context  
*iv* initialization vector (modified after use)  
*input* buffer holding the plaintext  
*output* buffer holding the ciphertext  
*len* length of the data to be encrypted

Definition at line 589 of file des.c.

References des3\_crypt(), and des3\_context::esk.

Referenced by main(), and ssl\_encrypt\_buf().

**8.112.2.3 static void des3\_crypt (unsigned long *SK*[96], unsigned char *input*[8], unsigned char *output*[8])** [static]

Definition at line 522 of file des.c.

References DES\_FP, DES\_IP, DES\_ROUND, GET\_UINT32\_BE, and PUT\_UINT32\_BE.

Referenced by des3\_cbc\_decrypt(), des3\_cbc\_encrypt(), des3\_decrypt(), and des3\_encrypt().

**8.112.2.4 void des3\_decrypt (des3\_context \* *ctx*, unsigned char *input*[8], unsigned char *output*[8])**

Triple-DES block decryption (ECB mode).

**Parameters:**

*ctx* 3DES context

*input* ciphertext block

*output* plaintext block

Definition at line 579 of file des.c.

References des3\_crypt(), and des3\_context::dsk.

**8.112.2.5 void des3\_encrypt (des3\_context \* *ctx*, unsigned char *input*[8], unsigned char *output*[8])**

Triple-DES block encryption (ECB mode).

**Parameters:**

*ctx* 3DES context

*input* plaintext block

*output* ciphertext block

Definition at line 569 of file des.c.

References des3\_crypt(), and des3\_context::esk.

**8.112.2.6 void des3\_set\_2keys (des3\_context \* *ctx*, unsigned char *key*[16])**

Triple-DES key schedule (112-bit).

**Parameters:**

*ctx* 3DES context to be initialized

*key* 16-byte secret key

Definition at line 475 of file des.c.

References des\_main\_ks(), des3\_context::dsk, and des3\_context::esk.

**8.112.2.7 void des3\_set\_3keys (des3\_context \* ctx, unsigned char key[24])**

Triple-DES key schedule (168-bit).

**Parameters:**

*ctx* 3DES context to be initialized

*key* 24-byte secret key

Definition at line 501 of file des.c.

References des\_main\_ks(), des3\_context::dsk, and des3\_context::esk.

Referenced by main(), ssl\_derive\_keys(), and x509\_des3\_decrypt().

**8.112.2.8 void des\_cbc\_decrypt (des\_context \* ctx, unsigned char iv[8], unsigned char \* input, unsigned char \* output, int len)**

DES-CBC buffer decryption.

**Parameters:**

*ctx* DES context

*iv* initialization vector (modified after use)

*input* buffer holding the ciphertext

*output* buffer holding the plaintext

*len* length of the data to be decrypted

Definition at line 447 of file des.c.

References des\_crypt(), and des\_context::dsk.

**8.112.2.9 void des\_cbc\_encrypt (des\_context \* ctx, unsigned char iv[8], unsigned char \* input, unsigned char \* output, int len)**

DES-CBC buffer encryption.

**Parameters:**

*ctx* DES context

*iv* initialization vector (modified after use)

*input* buffer holding the plaintext

*output* buffer holding the ciphertext

*len* length of the data to be encrypted

Definition at line 422 of file des.c.

References des\_crypt(), and des\_context::esk.

Referenced by main().

**8.112.2.10 static void des\_crypt (unsigned long SK[32], unsigned char *input*[8], unsigned char *output*[8])** *[static]*

Definition at line 373 of file des.c.

References DES\_FP, DES\_IP, DES\_ROUND, GET\_UINT32\_BE, and PUT\_UINT32\_BE.

Referenced by des\_cbc\_decrypt(), des\_cbc\_encrypt(), des\_decrypt(), and des\_encrypt().

**8.112.2.11 void des\_decrypt (des\_context \* *ctx*, unsigned char *input*[8], unsigned char *output*[8])**

DES block decryption (ECB mode).

**Parameters:**

*ctx* DES context

*input* ciphertext block

*output* plaintext block

Definition at line 412 of file des.c.

References des\_crypt(), and des\_context::dsk.

**8.112.2.12 void des\_encrypt (des\_context \* *ctx*, unsigned char *input*[8], unsigned char *output*[8])**

DES block encryption (ECB mode).

**Parameters:**

*ctx* DES context

*input* plaintext block

*output* ciphertext block

Definition at line 402 of file des.c.

References des\_crypt(), and des\_context::esk.

**8.112.2.13 static void des\_main\_ks (unsigned long SK[32], unsigned char *key*[8])** *[static]*

Definition at line 288 of file des.c.

References GET\_UINT32\_BE, LHs, and RHs.

Referenced by des3\_set\_2keys(), des3\_set\_3keys(), and des\_set\_key().

**8.112.2.14 int des\_self\_test (int *verbose*)**

Definition at line 736 of file des.c.

Referenced by main().

### 8.112.2.15 void des\_set\_key (des\_context \* ctx, unsigned char key[8])

DES key schedule (56-bit).

#### Parameters:

*ctx* DES context to be initialized  
*key* 8-byte secret key

Definition at line 360 of file des.c.

References des\_main\_ks(), des\_context::dsk, and des\_context::esk.

Referenced by main().

## 8.112.3 Variable Documentation

### 8.112.3.1 const char \_des\_src[ ] = "\_des\_src" [static]

Definition at line 639 of file des.c.

### 8.112.3.2 const unsigned long LHs[16] [static]

#### Initial value:

```
{
    0x00000000, 0x00000001, 0x00000100, 0x00000101,
    0x00010000, 0x00010001, 0x00010100, 0x00010101,
    0x01000000, 0x01000001, 0x01000100, 0x01000101,
    0x01010000, 0x01010001, 0x01010100, 0x01010101
}
```

Definition at line 226 of file des.c.

Referenced by des\_main\_ks().

### 8.112.3.3 const unsigned long RHs[16] [static]

#### Initial value:

```
{
    0x00000000, 0x01000000, 0x00010000, 0x01010000,
    0x00000100, 0x01000100, 0x00010100, 0x01010100,
    0x00000001, 0x01000001, 0x00010001, 0x01010001,
    0x00000101, 0x01000101, 0x00010101, 0x01010101,
}
```

Definition at line 234 of file des.c.

Referenced by des\_main\_ks().

### 8.112.3.4 const unsigned long SB1[64] [static]

#### Initial value:

```

{
    0x01010400, 0x00000000, 0x00010000, 0x01010404,
    0x01010004, 0x00010404, 0x00000004, 0x00010000,
    0x00000400, 0x01010400, 0x01010404, 0x00000400,
    0x01000404, 0x01010004, 0x01000000, 0x00000004,
    0x00000404, 0x01000400, 0x01000400, 0x00010400,
    0x00010400, 0x01010000, 0x01010000, 0x01000404,
    0x00010004, 0x01000004, 0x01000004, 0x00010004,
    0x00000000, 0x00000404, 0x00010404, 0x01000000,
    0x00010000, 0x01010404, 0x00000004, 0x01010000,
    0x01010400, 0x01000000, 0x01000000, 0x00000400,
    0x01010004, 0x00010000, 0x00010400, 0x01000004,
    0x00000400, 0x00000004, 0x01000404, 0x00010404,
    0x01010404, 0x00010004, 0x01010000, 0x01000404,
    0x01000004, 0x00000404, 0x00010404, 0x01010400,
    0x00000404, 0x01000400, 0x01000400, 0x00000000,
    0x00010004, 0x00010400, 0x00000000, 0x01010004
}

```

Definition at line 63 of file des.c.

### 8.112.3.5 const unsigned long SB2[64] [static]

**Initial value:**

```

{
    0x80108020, 0x80008000, 0x00008000, 0x00108020,
    0x00100000, 0x00000020, 0x80100020, 0x80008020,
    0x80000020, 0x80108020, 0x80108000, 0x80000000,
    0x80008000, 0x00100000, 0x00000020, 0x80100020,
    0x00108000, 0x00100020, 0x80008020, 0x00000000,
    0x80000000, 0x00008000, 0x00108020, 0x80100000,
    0x00100020, 0x80000020, 0x00000000, 0x00108000,
    0x00008020, 0x80108000, 0x80100000, 0x00008020,
    0x00000000, 0x00108020, 0x80100020, 0x00100000,
    0x80008020, 0x80100000, 0x80108000, 0x00008000,
    0x80100000, 0x80008000, 0x00000020, 0x80108020,
    0x00108020, 0x00000020, 0x00008000, 0x80000000,
    0x00008020, 0x80108000, 0x00100000, 0x80000020,
    0x00100020, 0x80008020, 0x80000020, 0x00100020,
    0x00108000, 0x00000000, 0x80008000, 0x00008020,
    0x80000000, 0x80100020, 0x80108020, 0x00108000
}

```

Definition at line 83 of file des.c.

### 8.112.3.6 const unsigned long SB3[64] [static]

**Initial value:**

```

{
    0x00000208, 0x08020200, 0x00000000, 0x08020008,
    0x08000200, 0x00000000, 0x00020208, 0x08000200,
    0x00020008, 0x08000008, 0x08000008, 0x00020000,
    0x08020208, 0x00020008, 0x08020000, 0x00000208,
    0x08000000, 0x00000008, 0x08020200, 0x00000200,
    0x00020200, 0x08020000, 0x08020008, 0x00020208,
    0x08000208, 0x00020200, 0x00020000, 0x08000208,
    0x00000008, 0x08020208, 0x00000200, 0x08000000,
    0x08020200, 0x08000000, 0x00020008, 0x00000208,
    0x00020000, 0x08020200, 0x08000200, 0x00000000,
}

```

```

    0x00000200, 0x00020008, 0x08020208, 0x08000200,
    0x08000008, 0x00000200, 0x00000000, 0x08020008,
    0x08000208, 0x00020000, 0x08000000, 0x08020208,
    0x00000008, 0x00020208, 0x00020200, 0x08000008,
    0x08020000, 0x08000208, 0x00000208, 0x08020000,
    0x00020208, 0x00000008, 0x08020008, 0x00020200
}

```

Definition at line 103 of file des.c.

### 8.112.3.7 `const unsigned long SB4[64]` `[static]`

**Initial value:**

```

{
    0x00802001, 0x00002081, 0x00002081, 0x00000080,
    0x00802080, 0x00800081, 0x00800001, 0x00002001,
    0x00000000, 0x00802000, 0x00802000, 0x00802081,
    0x00000081, 0x00000000, 0x00800080, 0x00800001,
    0x00000001, 0x00002000, 0x00800000, 0x00802001,
    0x00000080, 0x00800000, 0x00002001, 0x00002080,
    0x00800081, 0x00000001, 0x00002080, 0x00800080,
    0x00002000, 0x00802080, 0x00802081, 0x00000081,
    0x00800080, 0x00800001, 0x00802000, 0x00802081,
    0x00000081, 0x00000000, 0x00000000, 0x00802000,
    0x00002080, 0x00800080, 0x00800081, 0x00000001,
    0x00802001, 0x00002081, 0x00002081, 0x00000080,
    0x00802081, 0x00000081, 0x00000001, 0x00002000,
    0x00800001, 0x00002001, 0x00802080, 0x00800081,
    0x00002001, 0x00002080, 0x00800000, 0x00802001,
    0x00000080, 0x00800000, 0x00002000, 0x00802080
}

```

Definition at line 123 of file des.c.

### 8.112.3.8 `const unsigned long SB5[64]` `[static]`

**Initial value:**

```

{
    0x00000100, 0x02080100, 0x02080000, 0x42000100,
    0x00080000, 0x00000100, 0x40000000, 0x02080000,
    0x40080100, 0x00080000, 0x02000100, 0x40080100,
    0x42000100, 0x42080000, 0x00080100, 0x40000000,
    0x02000000, 0x40080000, 0x40080000, 0x00000000,
    0x40000100, 0x42080100, 0x42080100, 0x02000100,
    0x42080000, 0x40000100, 0x00000000, 0x42000000,
    0x02080100, 0x02000000, 0x42000000, 0x00080100,
    0x00080000, 0x42000100, 0x00000100, 0x02000000,
    0x40000000, 0x02080000, 0x42000100, 0x40080100,
    0x02000100, 0x40000000, 0x42080000, 0x02080100,
    0x40080100, 0x00000100, 0x02000000, 0x42080000,
    0x42080100, 0x00080100, 0x42000000, 0x42080100,
    0x02080000, 0x00000000, 0x40080000, 0x42000000,
    0x00080100, 0x02000100, 0x40000100, 0x00080000,
    0x00000000, 0x40080000, 0x02080100, 0x40000100
}

```

Definition at line 143 of file des.c.



**8.112.3.9 const unsigned long SB6[64] [static]****Initial value:**

```
{
    0x20000010, 0x20400000, 0x00004000, 0x20404010,
    0x20400000, 0x00000010, 0x20404010, 0x00400000,
    0x20004000, 0x00404010, 0x00400000, 0x20000010,
    0x00400010, 0x20004000, 0x20000000, 0x00004010,
    0x00000000, 0x00400010, 0x20004010, 0x00004000,
    0x00404000, 0x20004010, 0x00000010, 0x20400010,
    0x20400010, 0x00000000, 0x00404010, 0x20404000,
    0x00004010, 0x00404000, 0x20404000, 0x20000000,
    0x20004000, 0x00000010, 0x20400010, 0x00404000,
    0x20404010, 0x00400000, 0x00004010, 0x20000010,
    0x00400000, 0x20004000, 0x20000000, 0x00004010,
    0x20000010, 0x20404010, 0x00404000, 0x20400000,
    0x00404010, 0x20404000, 0x00000000, 0x20400010,
    0x00000010, 0x00004000, 0x20400000, 0x00404010,
    0x00004000, 0x00400010, 0x20004010, 0x00000000,
    0x20404000, 0x20000000, 0x00400010, 0x20004010
}
```

Definition at line 163 of file des.c.

**8.112.3.10 const unsigned long SB7[64] [static]****Initial value:**

```
{
    0x00200000, 0x04200002, 0x04000802, 0x00000000,
    0x00000800, 0x04000802, 0x00200802, 0x04200800,
    0x04200802, 0x00200000, 0x00000000, 0x04000002,
    0x00000002, 0x04000000, 0x04200002, 0x00000802,
    0x04000800, 0x00200802, 0x00200002, 0x04000800,
    0x04000002, 0x04200000, 0x04200800, 0x00200002,
    0x04200000, 0x00000800, 0x00000802, 0x04200802,
    0x00200800, 0x00000002, 0x04000000, 0x00200800,
    0x04000000, 0x00200800, 0x00200000, 0x04000802,
    0x04000802, 0x04200002, 0x04200002, 0x00000002,
    0x00200002, 0x04000000, 0x04000800, 0x00200000,
    0x04200800, 0x00000802, 0x00200802, 0x04200800,
    0x00000802, 0x04000002, 0x04200802, 0x04200000,
    0x00200800, 0x00000000, 0x00000002, 0x04200802,
    0x00000000, 0x00200802, 0x04200000, 0x00000800,
    0x04000002, 0x04000800, 0x00000800, 0x00200002
}
```

Definition at line 183 of file des.c.

**8.112.3.11 const unsigned long SB8[64] [static]****Initial value:**

```
{
    0x10001040, 0x00001000, 0x00040000, 0x10041040,
    0x10000000, 0x10001040, 0x00000040, 0x10000000,
    0x00040040, 0x10040000, 0x10041040, 0x00041000,
    0x10041000, 0x00041040, 0x00001000, 0x00000040,
    0x10040000, 0x10000040, 0x10001000, 0x00001040,
}
```

```
0x00041000, 0x00040040, 0x10040040, 0x10041000,  
0x00001040, 0x00000000, 0x00000000, 0x10040040,  
0x10000040, 0x10001000, 0x00041040, 0x00040000,  
0x00041040, 0x00040000, 0x10041000, 0x00001000,  
0x00000040, 0x10040040, 0x00001000, 0x00041040,  
0x10001000, 0x00000040, 0x10000040, 0x10040000,  
0x10040040, 0x10000000, 0x00040000, 0x10001040,  
0x00000000, 0x10041040, 0x00040040, 0x10000040,  
0x10040000, 0x10001000, 0x10001040, 0x00000000,  
0x10041040, 0x00041000, 0x00041000, 0x00001040,  
0x00001040, 0x00040040, 0x10000000, 0x10041000  
}
```

Definition at line 203 of file des.c.

## 8.113 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/dhm.c File Reference

```
#include <string.h>
#include "xyssl/dhm.h"
```

### Defines

- `#define _CRT_SECURE_NO_DEPRECATED 1`
- `#define DHM_MPI_EXPORT(X, n)`

### Functions

- static `int dhm_read_bignum` (`mpi *X`, unsigned char \*\*p, unsigned char \*end)
- `int dhm_read_params` (`dhm_context *ctx`, unsigned char \*\*p, unsigned char \*end)  
*Parse the ServerKeyExchange parameters.*
- `int dhm_make_params` (`dhm_context *ctx`, `int(*rng_f)(void *)`, void \*rng\_d, unsigned char \*output, `int *olen`)  
*Setup and write the ServerKeyExchange parameters.*
- `int dhm_read_public` (`dhm_context *ctx`, unsigned char \*input, `int ilen`)  
*Import the peer's public value ( $G^Y$ ).*
- `int dhm_make_public` (`dhm_context *ctx`, unsigned char \*output, `int olen`, `int(*rng_f)(void *)`, void \*rng\_d)  
*Create private value X and export  $G^X$ .*
- `int dhm_calc_secret` (`dhm_context *ctx`, unsigned char \*output, `int *olen`)  
*Derive and export the shared secret  $(G^Y)^X \bmod P$ .*
- void `dhm_free` (`dhm_context *ctx`)
- `int dhm_self_test` (`int verbose`)  
*Checkup routine.*

### Variables

- static const char `_dhm_src` [] = "\_dhm\_src"

#### 8.113.1 Define Documentation

##### 8.113.1.1 `#define _CRT_SECURE_NO_DEPRECATED 1`

Definition at line 30 of file dhm.c.

### 8.113.1.2 #define DHM\_MPI\_EXPORT(X, n)

#### Value:

```
CHK( mpi_write_binary( X, p + 2, &n ) );    \
    *p++ = ( n >> 8 ); *p++ = n; p += n;
```

Referenced by `dhm_make_params()`.

## 8.113.2 Function Documentation

### 8.113.2.1 int dhm\_calc\_secret (dhm\_context \* ctx, unsigned char \* output, int \* olen)

Derive and export the shared secret  $(G^Y)^X \bmod P$ .

#### Parameters:

*ctx* DHM context  
*output* destination buffer  
*olen* number of chars written

#### Returns:

0 if successful, or `ERR_DHM_MAKE_PUBLIC_FAILED`

Definition at line 206 of file `dhm.c`.

References `CHK`, `ERR_DHM_BAD_INPUT_DATA`, `ERR_DHM_CALC_SECRET_FAILED`, `dhm_context::GY`, `dhm_context::K`, `mpi_exp_mod()`, `mpi_msb()`, `mpi_write_binary()`, `dhm_context::P`, `dhm_context::RP`, and `dhm_context::X`.

Referenced by `main()`, `ssl_parse_client_key_exchange()`, and `ssl_write_client_key_exchange()`.

### 8.113.2.2 void dhm\_free (dhm\_context \* ctx)

Definition at line 232 of file `dhm.c`.

References `dhm_context::G`, `dhm_context::GX`, `dhm_context::GY`, `dhm_context::K`, `mpi_free()`, `dhm_context::P`, `dhm_context::RP`, and `dhm_context::X`.

Referenced by `main()`, and `ssl_free()`.

### 8.113.2.3 int dhm\_make\_params (dhm\_context \* ctx, int(\*) (void \*) rng\_f, void \* rng\_d, unsigned char \* output, int \* olen)

Setup and write the ServerKeyExchange parameters.

#### Parameters:

*ctx* DHM context  
*rng\_f* points to the RNG function  
*rng\_d* points to the RNG data  
*output* destination buffer

*olen* number of chars written

**Note:**

This function assumes that `ctx->P` and `ctx->G` have already been properly set (for example using `mpi_read_string`).

**Returns:**

0 if successful, or an MPI error code

Definition at line 96 of file `dhm.c`.

References `CHK`, `DHM_MPI_EXPORT`, `ERR_DHM_MAKE_PARAMS_FAILED`, `dhm_context::G`, `dhm_context::GX`, `dhm_context::len`, `mpi_cmp_mpi()`, `mpi_exp_mod()`, `mpi_grow()`, `mpi_msb()`, `mpi_shift_r()`, `dhm_context::P`, `mpi::p`, `dhm_context::RP`, and `dhm_context::X`.

Referenced by `main()`, and `ssl_write_server_key_exchange()`.

#### 8.113.2.4 `int dhm_make_public (dhm_context * ctx, unsigned char * output, int olen, int(*) (void *) rng_f, void * rng_d)`

Create private value  $X$  and export  $G^X$ .

**Parameters:**

*ctx* DHM context

*output* destination buffer

*olen* must be == `ctx->P.len`

*rng\_f* points to the RNG function

*rng\_d* points to the RNG data

**Returns:**

0 if successful, or `ERR_DHM_MAKE_PUBLIC_FAILED`

Definition at line 167 of file `dhm.c`.

References `CHK`, `ERR_DHM_BAD_INPUT_DATA`, `ERR_DHM_MAKE_PUBLIC_FAILED`, `dhm_context::G`, `dhm_context::GX`, `dhm_context::len`, `mpi_cmp_mpi()`, `mpi_exp_mod()`, `mpi_grow()`, `mpi_shift_r()`, `mpi_write_binary()`, `dhm_context::P`, `mpi::p`, `dhm_context::RP`, and `dhm_context::X`.

Referenced by `main()`, and `ssl_write_client_key_exchange()`.

#### 8.113.2.5 `static int dhm_read_bignum (mpi * X, unsigned char ** p, unsigned char * end)` [static]

Definition at line 40 of file `dhm.c`.

References `ERR_DHM_BAD_INPUT_DATA`, `ERR_DHM_READ_PARAMS_FAILED`, and `mpi_read_binary()`.

Referenced by `dhm_read_params()`.

### 8.113.2.6 `int dhm_read_params (dhm_context * ctx, unsigned char ** p, unsigned char * end)`

Parse the ServerKeyExchange parameters.

#### Parameters:

*ctx* DHM context  
*p* &(start of input buffer)  
*end* end of buffer

#### Returns:

0 if successful, or ERR\_DHM\_READ\_PARAMS\_FAILED

Definition at line 66 of file dhm.c.

References `dhm_read_bignum()`, `ERR_DHM_BAD_INPUT_DATA`, `dhm_context::G`, `dhm_context::GY`, `dhm_context::len`, `mpi_msb()`, and `dhm_context::P`.

Referenced by `main()`, and `ssl_parse_server_key_exchange()`.

### 8.113.2.7 `int dhm_read_public (dhm_context * ctx, unsigned char * input, int ilen)`

Import the peer's public value ( $G^Y$ ).

#### Parameters:

*ctx* DHM context  
*input* input buffer  
*ilen* size of buffer

#### Returns:

0 if successful, or ERR\_DHM\_READ\_PUBLIC\_FAILED

Definition at line 150 of file dhm.c.

References `ERR_DHM_BAD_INPUT_DATA`, `ERR_DHM_READ_PUBLIC_FAILED`, `dhm_context::GY`, `dhm_context::len`, and `mpi_read_binary()`.

Referenced by `main()`, and `ssl_parse_client_key_exchange()`.

### 8.113.2.8 `int dhm_self_test (int verbose)`

Checkup routine.

#### Returns:

0 if successful, or 1 if the test failed

Definition at line 246 of file dhm.c.

## 8.113.3 Variable Documentation

### 8.113.3.1 `const char _dhm_src[] = "_dhm_src" [static]`

Definition at line 239 of file dhm.c.

## 8.114 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/dsa.c File Reference

```
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include "xyssl/rsa.h"
```

### Defines

- `#define _CRT_SECURE_NO_DEPRECATED 1`

### Functions

- `int rsa_gen_key (rsa_context *ctx, int nbits, int exponent, int(*rng_f)(void *), void *rng_d)`  
*Generate an RSA keypair.*
- `int rsa_public (rsa_context *ctx, unsigned char *input, int ilen, unsigned char *output, int olen)`  
*Perform an RSA public key operation.*
- `int rsa_private (rsa_context *ctx, unsigned char *input, int ilen, unsigned char *output, int olen)`  
*Perform an RSA private key operation.*
- `int rsa_check_pubkey (rsa_context *ctx)`  
*Return 0 if the public key is valid, or ERR\_RSA\_KEY\_CHECK\_FAILED.*
- `int rsa_check_privkey (rsa_context *ctx)`  
*Return 0 if the private key is valid, or ERR\_RSA\_KEY\_CHECK\_FAILED.*
- `int rsa_pkcs1_encrypt (rsa_context *ctx, unsigned char *input, int ilen, unsigned char *output, int olen)`  
*Add the PKCS#1 v1.5 padding and do a public RSA.*
- `int rsa_pkcs1_decrypt (rsa_context *ctx, unsigned char *input, int ilen, unsigned char *output, int *olen)`  
*Do a private RSA, removes the PKCS#1 v1.5 padding.*
- `int rsa_pkcs1_sign (rsa_context *ctx, int alg_id, unsigned char *hash, int hashlen, unsigned char *sig, int siglen)`  
*Perform a private RSA to sign a message digest.*
- `int rsa_pkcs1_verify (rsa_context *ctx, int alg_id, unsigned char *hash, int hashlen, unsigned char *sig, int siglen)`  
*Perform a public RSA and check the message digest.*
- `void rsa_free (rsa_context *ctx)`  
*Free the components of an RSA key.*

- `int rsa_self_test` (`int verbose`)

*Checkup routine.*

### 8.114.1 Define Documentation

#### 8.114.1.1 `#define _CRT_SECURE_NO_DEPRECATED 1`

Definition at line 31 of file `dsa.c`.

### 8.114.2 Function Documentation

#### 8.114.2.1 `int rsa_check_privkey` (`rsa_context * ctx`)

Return 0 if the private key is valid, or `ERR_RSA_KEY_CHECK_FAILED`.

Definition at line 231 of file `dsa.c`.

References `CHK`, `rsa_context::E`, `ERR_RSA_KEY_CHK_FAILED`, `mpi_cmp_int()`, `mpi_cmp_mpi()`, `mpi_free()`, `mpi_gcd()`, `mpi_init()`, `mpi_mul_mpi()`, `mpi_sub_int()`, `rsa_context::N`, `rsa_context::P`, and `rsa_context::Q`.

#### 8.114.2.2 `int rsa_check_pubkey` (`rsa_context * ctx`)

Return 0 if the public key is valid, or `ERR_RSA_KEY_CHECK_FAILED`.

Definition at line 211 of file `dsa.c`.

References `rsa_context::E`, `ERR_RSA_KEY_CHK_FAILED`, `mpi_msb()`, `rsa_context::N`, and `mpi::p`.

#### 8.114.2.3 `void rsa_free` (`rsa_context * ctx`)

Free the components of an RSA key.

Definition at line 474 of file `dsa.c`.

References `rsa_context::D`, `rsa_context::DP`, `rsa_context::DQ`, `rsa_context::E`, `mpi_free()`, `rsa_context::N`, `rsa_context::P`, `rsa_context::Q`, `rsa_context::QP`, `rsa_context::RN`, `rsa_context::RP`, and `rsa_context::RQ`.

#### 8.114.2.4 `int rsa_gen_key` (`rsa_context * ctx`, `int nbits`, `int exponent`, `int(*) (void *) rng_f`, `void * rng_d`)

Generate an RSA keypair.

#### Parameters:

*ctx* RSA context to be initialized

*nbits* size of the public key in bits

*exponent* public exponent (e.g., 65537)

*rng\_f* points to the RNG function

*rng\_d* points to the RNG data



**Returns:**

0 if successful, or an ERR\_RSA\_XXX error code

Definition at line 44 of file dsa.c.

References CHK, rsa\_context::D, rsa\_context::DP, rsa\_context::DQ, rsa\_context::E, ERR\_RSA\_BAD\_INPUT\_DATA, ERR\_RSA\_KEY\_GEN\_FAILED, rsa\_context::len, mpi\_cmp\_int(), mpi\_cmp\_mpi(), mpi\_free(), mpi\_gcd(), mpi\_gen\_prime(), mpi\_init(), mpi\_inv\_mod(), mpi\_lset(), mpi\_mod\_mpi(), mpi\_msb(), mpi\_mul\_mpi(), mpi\_sub\_int(), mpi\_swap(), rsa\_context::N, rsa\_context::P, rsa\_context::Q, rsa\_context::QP, and rsa\_free().

#### 8.114.2.5 **int rsa\_pkcs1\_decrypt** (rsa\_context \* *ctx*, unsigned char \* *input*, int *ilen*, unsigned char \* *output*, int \* *olen*)

Do a private RSA, removes the PKCS#1 v1.5 padding.

**Parameters:**

*ctx* RSA context

*input* buffer holding the encrypted data

*ilen* must be the same as the modulus size

*output* buffer that will hold the plaintext

*olen* size of output buffer, will be updated to contain the length of the plaintext

**Returns:**

0 if successful, or an ERR\_RSA\_XXX error code

Definition at line 290 of file dsa.c.

References ERR\_RSA\_BAD\_INPUT\_DATA, ERR\_RSA\_INVALID\_PADDING, int, rsa\_context::len, RSA\_CRYPT, and rsa\_private().

#### 8.114.2.6 **int rsa\_pkcs1\_encrypt** (rsa\_context \* *ctx*, unsigned char \* *input*, int *ilen*, unsigned char \* *output*, int \* *olen*)

Add the PKCS#1 v1.5 padding and do a public RSA.

**Parameters:**

*ctx* RSA context

*input* buffer holding the data to be encrypted

*ilen* length of the plaintext; cannot be longer than the modulus, minus 3+8 for padding

*output* buffer that will hold the ciphertext

*olen* must be the same as the modulus size (for example, 128 if RSA-1024 is used)

**Returns:**

0 if successful, or an ERR\_RSA\_XXX error code

Definition at line 260 of file dsa.c.

References ERR\_RSA\_BAD\_INPUT\_DATA, rsa\_context::len, RSA\_CRYPT, and rsa\_public().

**8.114.2.7 int rsa\_pkcs1\_sign (rsa\_context \* ctx, int alg\_id, unsigned char \* hash, int hashlen, unsigned char \* sig, int siglen)**

Perform a private RSA to sign a message digest.

**Parameters:**

*ctx* RSA context

*alg\_id* RSA\_RAW, RSA\_MD2/4/5 or RSA\_SHA1

*hash* buffer holding the message digest

*hashlen* message digest length

*sig* buffer that will hold the ciphertext

*siglen* must be the same as the modulus size (for example, 128 if RSA-1024 is used)

**Returns:**

0 if the signing operation was successful, or an ERR\_RSA\_XXX error code

Definition at line 328 of file dsa.c.

References ASN1\_HASH\_MDX, ASN1\_HASH\_SHA1, ERR\_RSA\_BAD\_INPUT\_DATA, rsa\_context::len, RSA\_MD2, RSA\_MD4, RSA\_MD5, rsa\_private(), RSA\_RAW, RSA\_SHA1, and RSA\_SIGN.

**8.114.2.8 int rsa\_pkcs1\_verify (rsa\_context \* ctx, int alg\_id, unsigned char \* hash, int hashlen, unsigned char \* sig, int siglen)**

Perform a public RSA and check the message digest.

**Parameters:**

*ctx* points to an RSA public key

*alg\_id* RSA\_RAW, RSA\_MD2/4/5 or RSA\_SHA1

*hash* buffer holding the message digest

*hashlen* message digest length

*sig* buffer holding the ciphertext

*siglen* must be the same as the modulus size

**Returns:**

0 if the verify operation was successful, or an ERR\_RSA\_XXX error code

Definition at line 404 of file dsa.c.

References ASN1\_HASH\_MDX, ASN1\_HASH\_SHA1, ERR\_RSA\_BAD\_INPUT\_DATA, ERR\_RSA\_INVALID\_PADDING, ERR\_RSA\_VERIFY\_FAILED, int, rsa\_context::len, RSA\_MD2, RSA\_MD4, RSA\_MD5, rsa\_public(), RSA\_RAW, RSA\_SHA1, and RSA\_SIGN.

**8.114.2.9 int rsa\_private (rsa\_context \* ctx, unsigned char \* input, int ilen, unsigned char \* output, int olen)**

Perform an RSA private key operation.

**Returns:**

0 if successful, or an ERR\_RSA\_XXX error code

**Note:**

This function does not take care of message padding: both ilen and olen must be equal to the modulus size (ctx->len). Also, be sure to set input[0] = 0.

Definition at line 150 of file dsa.c.

References CHK, rsa\_context::D, rsa\_context::DP, rsa\_context::DQ, ERR\_RSA\_BAD\_INPUT\_DATA, ERR\_RSA\_PRIVATE\_FAILED, rsa\_context::len, mpi\_add\_mpi(), mpi\_cmp\_mpi(), mpi\_exp\_mod(), mpi\_free(), mpi\_init(), mpi\_mod\_mpi(), mpi\_mul\_mpi(), mpi\_read\_binary(), mpi\_sub\_mpi(), mpi\_write\_binary(), rsa\_context::N, rsa\_context::P, rsa\_context::Q, rsa\_context::QP, rsa\_context::RN, rsa\_context::RP, and rsa\_context::RQ.

**8.114.2.10 int rsa\_public (rsa\_context \* ctx, unsigned char \* input, int ilen, unsigned char \* output, int olen)**

Perform an RSA public key operation.

**Returns:**

0 if successful, or an ERR\_RSA\_XXX error code

**Note:**

This function does not take care of message padding: both ilen and olen must be equal to the modulus size (ctx->len). Also, be sure to set input[0] = 0.

Definition at line 114 of file dsa.c.

References CHK, rsa\_context::E, ERR\_RSA\_BAD\_INPUT\_DATA, ERR\_RSA\_PUBLIC\_FAILED, rsa\_context::len, mpi\_cmp\_mpi(), mpi\_exp\_mod(), mpi\_free(), mpi\_init(), mpi\_read\_binary(), mpi\_write\_binary(), rsa\_context::N, and rsa\_context::RN.

**8.114.2.11 int rsa\_self\_test (int verbose)**

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 588 of file dsa.c.

## 8.115 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/havege.c File Reference

```
#include <string.h>
#include <time.h>
#include "xyssl/timing.h"
#include "xyssl/havege.h"
```

### Defines

- `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`
- `#define SWAP(X, Y) { int *T = X; X = Y; Y = T; }`
- `#define TST1\_ENTER if( PTEST & 1 ) { PTEST ^= 3; PTEST >>= 1;`
- `#define TST2\_ENTER if( PTEST & 1 ) { PTEST ^= 3; PTEST >>= 1;`
- `#define TST1\_LEAVE U1++; }`
- `#define TST2\_LEAVE U2++; }`
- `#define ONE\_ITERATION`

### Functions

- static void [havege\\_fill](#) ([havege\\_state](#) \*hs)
- void [havege\\_init](#) ([havege\\_state](#) \*hs)  
*HAVEGE initialization.*
- int [havege\\_rand](#) (void \*rng\_d)  
*HAVEGE rand function.*

### Variables

- static const char [\\_havege\\_src](#) [ ] = "\_havege\_src"

#### 8.115.1 Define Documentation

##### 8.115.1.1 `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

Definition at line 32 of file havege.c.

##### 8.115.1.2 `#define ONE\_ITERATION`

Definition at line 63 of file havege.c.

Referenced by [havege\\_fill](#)().

##### 8.115.1.3 `#define SWAP(X, Y) { int *T = X; X = Y; Y = T; }`

Definition at line 55 of file havege.c.

**8.115.1.4** `#define TST1_ENTER if( PTEST & 1 ) { PTEST ^= 3; PTEST >>= 1;`

Definition at line 57 of file havege.c.

**8.115.1.5** `#define TST1_LEAVE U1++; }`

Definition at line 60 of file havege.c.

**8.115.1.6** `#define TST2_ENTER if( PTEST & 1 ) { PTEST ^= 3; PTEST >>= 1;`

Definition at line 58 of file havege.c.

**8.115.1.7** `#define TST2_LEAVE U2++; }`

Definition at line 61 of file havege.c.

## 8.115.2 Function Documentation

**8.115.2.1** `static void havege_fill (havege_state * hs)` `[static]`

Definition at line 157 of file havege.c.

References COLLECT\_SIZE, havege\_state::offset, ONE\_ITERATION, havege\_state::PT1, havege\_state::PT2, and havege\_state::WALK.

Referenced by havege\_init(), and havege\_rand().

**8.115.2.2** `void havege_init (havege_state * hs)`

HAVEGE initialization.

### Parameters:

*hs* HAVEGE state to be initialized

Definition at line 191 of file havege.c.

References havege\_fill().

Referenced by main().

**8.115.2.3** `int havege_rand (void * rng_d)`

HAVEGE rand function.

### Parameters:

*rng\_st* points to an HAVEGE state

### Returns:

A random int

Definition at line 201 of file havege.c.

References COLLECT\_SIZE, havege\_fill(), havege\_state::offset, and havege\_state::pool.

Referenced by main().

### 8.115.3 Variable Documentation

#### 8.115.3.1 `const char _havege_src[] = "_havege_src"` `[static]`

Definition at line 212 of file havege.c.

## 8.116 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/md2.c File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/md2.h"
```

### Defines

- `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

### Functions

- void [md2\\_starts](#) ([md2\\_context](#) \*ctx)  
*MD2 context setup.*
- static void [md2\\_process](#) ([md2\\_context](#) \*ctx)
- void [md2\\_update](#) ([md2\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*MD2 process buffer.*
- void [md2\\_finish](#) ([md2\\_context](#) \*ctx, unsigned char \*output)  
*MD2 final digest.*
- void [md2](#) (unsigned char \*input, [int](#) ilen, unsigned char \*output)  
*Output = MD2( input buffer ).*
- [int](#) [md2\\_file](#) (char \*path, unsigned char \*output)  
*Output = MD2( file contents ).*
- void [md2\\_hmac\\_starts](#) ([md2\\_context](#) \*ctx, unsigned char \*key, [int](#) keylen)  
*MD2 HMAC context setup.*
- void [md2\\_hmac\\_update](#) ([md2\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*MD2 HMAC process buffer.*
- void [md2\\_hmac\\_finish](#) ([md2\\_context](#) \*ctx, unsigned char \*output)  
*MD2 HMAC final digest.*
- void [md2\\_hmac](#) (unsigned char \*key, [int](#) keylen, unsigned char \*input, [int](#) ilen, unsigned char \*output)  
*Output = HMAC-MD2( hmac key, input buffer ).*
- [int](#) [md2\\_self\\_test](#) ([int](#) verbose)  
*Checkup routine.*

## Variables

- static const unsigned char `PI_SUBST` [256]
- static const char `_md2_src` [] = "\_md2\_src"

### 8.116.1 Define Documentation

#### 8.116.1.1 `#define _CRT_SECURE_NO_DEPRECATED 1`

Definition at line 31 of file md2.c.

### 8.116.2 Function Documentation

#### 8.116.2.1 `void md2 (unsigned char * input, int ilen, unsigned char * output)`

Output = MD2( *input* buffer ).

##### Parameters:

*input* buffer holding the data  
*ilen* length of the input data  
*output* MD2 checksum result

Definition at line 154 of file md2.c.

References `md2_finish()`, `md2_starts()`, and `md2_update()`.

Referenced by `x509_hash()`.

#### 8.116.2.2 `int md2_file (char * path, unsigned char * output)`

Output = MD2( *file* contents ).

##### Parameters:

*path* input file name  
*output* MD2 checksum result

##### Returns:

0 if successful, 1 if `fopen` failed, or 2 if `fread` failed

Definition at line 169 of file md2.c.

References `f`, `md2_finish()`, `md2_starts()`, and `md2_update()`.

#### 8.116.2.3 `void md2_finish (md2_context * ctx, unsigned char * output)`

MD2 final digest.

##### Parameters:

*ctx* MD2 context



*output* MD2 checksum result

Definition at line 133 of file md2.c.

References md2\_context::buffer, md2\_context::cksum, md2\_context::left, md2\_process(), and md2\_context::state.

Referenced by md2(), md2\_file(), and md2\_hmac\_finish().

#### 8.116.2.4 void md2\_hmac (unsigned char \* *key*, int *keylen*, unsigned char \* *input*, int *ilen*, unsigned char \* *output*)

Output = HMAC-MD2( hmac key, input buffer ).

##### Parameters:

*key* HMAC secret key

*keylen* length of the HMAC key

*input* buffer holding the data

*ilen* length of the input data

*output* HMAC-MD2 result

Definition at line 249 of file md2.c.

References md2\_hmac\_finish(), md2\_hmac\_starts(), and md2\_hmac\_update().

#### 8.116.2.5 void md2\_hmac\_finish (md2\_context \* *ctx*, unsigned char \* *output*)

MD2 HMAC final digest.

##### Parameters:

*ctx* HMAC context

*output* MD2 HMAC checksum result

Definition at line 233 of file md2.c.

References md2\_finish(), md2\_starts(), md2\_update(), and md2\_context::opad.

Referenced by md2\_hmac().

#### 8.116.2.6 void md2\_hmac\_starts (md2\_context \* *ctx*, unsigned char \* *key*, int *keylen*)

MD2 HMAC context setup.

##### Parameters:

*ctx* HMAC context to be initialized

*key* HMAC secret key

*keylen* length of the HMAC key

Definition at line 201 of file md2.c.

References md2\_context::ipad, md2\_starts(), md2\_update(), and md2\_context::opad.

Referenced by md2\_hmac().

**8.116.2.7 void md2\_hmac\_update (md2\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

MD2 HMAC process buffer.

**Parameters:**

*ctx* HMAC context  
*input* buffer holding the data  
*ilen* length of the input data

Definition at line 224 of file md2.c.

References md2\_update().

Referenced by md2\_hmac().

**8.116.2.8 static void md2\_process (md2\_context \* *ctx*) [static]**

Definition at line 77 of file md2.c.

References md2\_context::buffer, md2\_context::cksum, PI\_SUBST, and md2\_context::state.

Referenced by md2\_finish(), and md2\_update().

**8.116.2.9 int md2\_self\_test (int *verbose*)**

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 330 of file md2.c.

Referenced by main().

**8.116.2.10 void md2\_starts (md2\_context \* *ctx*)**

MD2 context setup.

**Parameters:**

*ctx* context to be initialized

Definition at line 72 of file md2.c.

Referenced by md2(), md2\_file(), md2\_hmac\_finish(), and md2\_hmac\_starts().

**8.116.2.11 void md2\_update (md2\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

MD2 process buffer.

**Parameters:**

*ctx* MD2 context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 105 of file md2.c.

References md2\_context::buffer, md2\_context::left, and md2\_process().

Referenced by md2(), md2\_file(), md2\_hmac\_finish(), md2\_hmac\_starts(), and md2\_hmac\_update().

### 8.116.3 Variable Documentation

#### 8.116.3.1 `const char _md2_src[] = "_md2_src" [static]`

Definition at line 262 of file md2.c.

#### 8.116.3.2 `const unsigned char PI_SUBST[256] [static]`

**Initial value:**

```
{
    0x29, 0x2E, 0x43, 0xC9, 0xA2, 0xD8, 0x7C, 0x01, 0x3D, 0x36,
    0x54, 0xA1, 0xEC, 0xF0, 0x06, 0x13, 0x62, 0xA7, 0x05, 0xF3,
    0xC0, 0xC7, 0x73, 0x8C, 0x98, 0x93, 0x2B, 0xD9, 0xBC, 0x4C,
    0x82, 0xCA, 0x1E, 0x9B, 0x57, 0x3C, 0xFD, 0xD4, 0xE0, 0x16,
    0x67, 0x42, 0x6F, 0x18, 0x8A, 0x17, 0xE5, 0x12, 0xBE, 0x4E,
    0xC4, 0xD6, 0xDA, 0x9E, 0xDE, 0x49, 0xA0, 0xFB, 0xF5, 0x8E,
    0xBB, 0x2F, 0xEE, 0x7A, 0xA9, 0x68, 0x79, 0x91, 0x15, 0xB2,
    0x07, 0x3F, 0x94, 0xC2, 0x10, 0x89, 0x0B, 0x22, 0x5F, 0x21,
    0x80, 0x7F, 0x5D, 0x9A, 0x5A, 0x90, 0x32, 0x27, 0x35, 0x3E,
    0xCC, 0xE7, 0xBF, 0xF7, 0x97, 0x03, 0xFF, 0x19, 0x30, 0xB3,
    0x48, 0xA5, 0xB5, 0xD1, 0xD7, 0x5E, 0x92, 0x2A, 0xAC, 0x56,
    0xAA, 0xC6, 0x4F, 0xB8, 0x38, 0xD2, 0x96, 0xA4, 0x7D, 0xB6,
    0x76, 0xFC, 0x6B, 0xE2, 0x9C, 0x74, 0x04, 0xF1, 0x45, 0x9D,
    0x70, 0x59, 0x64, 0x71, 0x87, 0x20, 0x86, 0x5B, 0xCF, 0x65,
    0xE6, 0x2D, 0xA8, 0x02, 0x1B, 0x60, 0x25, 0xAD, 0xAE, 0xB0,
    0xB9, 0xF6, 0x1C, 0x46, 0x61, 0x69, 0x34, 0x40, 0x7E, 0x0F,
    0x55, 0x47, 0xA3, 0x23, 0xDD, 0x51, 0xAF, 0x3A, 0xC3, 0x5C,
    0xF9, 0xCE, 0xBA, 0xC5, 0xEA, 0x26, 0x2C, 0x53, 0x0D, 0x6E,
    0x85, 0x28, 0x84, 0x09, 0xD3, 0xDF, 0xCD, 0xF4, 0x41, 0x81,
    0x4D, 0x52, 0x6A, 0xDC, 0x37, 0xC8, 0x6C, 0xC1, 0xAB, 0xFA,
    0x24, 0xE1, 0x7B, 0x08, 0x0C, 0xBD, 0xB1, 0x4A, 0x78, 0x88,
    0x95, 0x8B, 0xE3, 0x63, 0xE8, 0x6D, 0xE9, 0xCB, 0xD5, 0xFE,
    0x3B, 0x00, 0x1D, 0x39, 0xF2, 0xEF, 0xB7, 0x0E, 0x66, 0x58,
    0xD0, 0xE4, 0xA6, 0x77, 0x72, 0xF8, 0xEB, 0x75, 0x4B, 0x0A,
    0x31, 0x44, 0x50, 0xB4, 0x8F, 0xED, 0x1F, 0x1A, 0xDB, 0x99,
    0x8D, 0x33, 0x9F, 0x11, 0x83, 0x14
}
```

Definition at line 39 of file md2.c.

Referenced by md2\_process().

## 8.117 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/md4.c File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/md4.h"
```

### Defines

- #define [\\_CRT\\_SECURE\\_NO\\_DEPRECATED](#) 1
- #define [GET\\_UINT32\\_LE](#)(n, b, i)
- #define [PUT\\_UINT32\\_LE](#)(n, b, i)
- #define [S](#)(x, n) ((x << n) | ((x & 0xFFFFFFFF) >> (32 - n)))
- #define [F](#)(x, y, z) ((x & y) | ((~x) & z))
- #define [P](#)(a, b, c, d, x, s) { a += F(b,c,d) + x; a = S(a,s); }
- #define [F](#)(x, y, z) ((x & y) | (x & z) | (y & z))
- #define [P](#)(a, b, c, d, x, s) { a += F(b,c,d) + x + 0x5A827999; a = S(a,s); }
- #define [F](#)(x, y, z) (x ^ y ^ z)
- #define [P](#)(a, b, c, d, x, s) { a += F(b,c,d) + x + 0x6ED9EBA1; a = S(a,s); }

### Functions

- void [md4\\_starts](#) ([md4\\_context](#) \*ctx)  
*MD4 context setup.*
- static void [md4\\_process](#) ([md4\\_context](#) \*ctx, unsigned char data[64])
- void [md4\\_update](#) ([md4\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*MD4 process buffer.*
- void [md4\\_finish](#) ([md4\\_context](#) \*ctx, unsigned char \*output)  
*MD4 final digest.*
- void [md4](#) (unsigned char \*input, [int](#) ilen, unsigned char \*output)  
*Output = MD4( input buffer ).*
- [int](#) [md4\\_file](#) (char \*path, unsigned char \*output)  
*Output = MD4( file contents ).*
- void [md4\\_hmac\\_starts](#) ([md4\\_context](#) \*ctx, unsigned char \*key, [int](#) keylen)  
*MD4 HMAC context setup.*
- void [md4\\_hmac\\_update](#) ([md4\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*MD4 HMAC process buffer.*
- void [md4\\_hmac\\_finish](#) ([md4\\_context](#) \*ctx, unsigned char \*output)  
*MD4 HMAC final digest.*

- void `md4_hmac` (unsigned char \*key, int keylen, unsigned char \*input, int ilen, unsigned char \*output)

*Output = HMAC-MD4( hmac key, input buffer ).*

- int `md4_self_test` (int verbose)

*Checkup routine.*

## Variables

- static const unsigned char `md4_padding` [64]
- static const char `_md4_src` [] = "\_md4\_src"

### 8.117.1 Define Documentation

#### 8.117.1.1 #define \_CRT\_SECURE\_NO\_DEPRECATED 1

Definition at line 31 of file md4.c.

#### 8.117.1.2 #define F(x, y, z) (x ^ y ^ z)

#### 8.117.1.3 #define F(x, y, z) ((x & y) | (x & z) | (y & z))

#### 8.117.1.4 #define F(x, y, z) ((x & y) | ((~x) & z))

Referenced by `sha2_process()`, and `sha4_process()`.

#### 8.117.1.5 #define GET\_UINT32\_LE(n, b, i)

**Value:**

```
{
    (n) = ( (unsigned long) (b)[(i)      ] )
    | ( (unsigned long) (b)[(i) + 1] << 8 )
    | ( (unsigned long) (b)[(i) + 2] << 16 )
    | ( (unsigned long) (b)[(i) + 3] << 24 );
}
```

Definition at line 43 of file md4.c.

Referenced by `md4_process()`, and `md5_process()`.

#### 8.117.1.6 #define P(a, b, c, d, x, s) { a += F(b,c,d) + x + 0x6ED9EBA1; a = S(a,s); }

#### 8.117.1.7 #define P(a, b, c, d, x, s) { a += F(b,c,d) + x + 0x5A827999; a = S(a,s); }

#### 8.117.1.8 #define P(a, b, c, d, x, s) { a += F(b,c,d) + x; a = S(a,s); }

Referenced by `main()`, `md4_process()`, `md5_process()`, `sha1_process()`, `sha2_process()`, and `sha4_process()`.

**8.117.1.9 #define PUT\_UINT32\_LE(n, b, i)****Value:**

```

{
    (b)[(i)] = (unsigned char) ((n) >> 0);
    (b)[(i) + 1] = (unsigned char) ((n) >> 8);
    (b)[(i) + 2] = (unsigned char) ((n) >> 16);
    (b)[(i) + 3] = (unsigned char) ((n) >> 24);
}

```

Definition at line 53 of file md4.c.

Referenced by md4\_finish(), and md5\_finish().

**8.117.1.10 #define S(x, n) ((x << n) | ((x & 0xFFFFFFFF) >> (32 - n)))****8.117.2 Function Documentation****8.117.2.1 void md4 (unsigned char \* *input*, int *ilen*, unsigned char \* *output*)**

Output = MD4( *input* buffer ).

**Parameters:**

*input* buffer holding the data  
*ilen* length of the input data  
*output* MD4 checksum result

Definition at line 262 of file md4.c.

References md4\_finish(), md4\_starts(), and md4\_update().

Referenced by main(), and x509\_hash().

**8.117.2.2 int md4\_file (char \* *path*, unsigned char \* *output*)**

Output = MD4( *file* contents ).

**Parameters:**

*path* input file name  
*output* MD4 checksum result

**Returns:**

0 if successful, 1 if fopen failed, or 2 if fread failed

Definition at line 277 of file md4.c.

References f, md4\_finish(), md4\_starts(), and md4\_update().

**8.117.2.3 void md4\_finish (md4\_context \* *ctx*, unsigned char \* *output*)**

MD4 final digest.

**Parameters:**

*ctx* MD4 context  
*output* MD4 checksum result

Definition at line 234 of file md4.c.

References md4\_padding, md4\_update(), PUT\_UINT32\_LE, md4\_context::state, and md4\_context::total.

Referenced by md4(), md4\_file(), and md4\_hmac\_finish().

**8.117.2.4 void md4\_hmac (unsigned char \* *key*, int *keylen*, unsigned char \* *input*, int *ilen*, unsigned char \* *output*)**

Output = HMAC-MD4( hmac key, input buffer ).

**Parameters:**

*key* HMAC secret key  
*keylen* length of the HMAC key  
*input* buffer holding the data  
*ilen* length of the input data  
*output* HMAC-MD4 result

Definition at line 357 of file md4.c.

References md4\_hmac\_finish(), md4\_hmac\_starts(), and md4\_hmac\_update().

**8.117.2.5 void md4\_hmac\_finish (md4\_context \* *ctx*, unsigned char \* *output*)**

MD4 HMAC final digest.

**Parameters:**

*ctx* HMAC context  
*output* MD4 HMAC checksum result

Definition at line 341 of file md4.c.

References md4\_finish(), md4\_starts(), md4\_update(), and md4\_context::opad.

Referenced by md4\_hmac().

**8.117.2.6 void md4\_hmac\_starts (md4\_context \* *ctx*, unsigned char \* *key*, int *keylen*)**

MD4 HMAC context setup.

**Parameters:**

*ctx* HMAC context to be initialized

*key* HMAC secret key

*keylen* length of the HMAC key

Definition at line 309 of file md4.c.

References md4\_context::ipad, md4\_starts(), md4\_update(), and md4\_context::opad.

Referenced by md4\_hmac().

#### 8.117.2.7 void md4\_hmac\_update (md4\_context \* ctx, unsigned char \* input, int ilen)

MD4 HMAC process buffer.

##### Parameters:

*ctx* HMAC context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 332 of file md4.c.

References md4\_update().

Referenced by md4\_hmac().

#### 8.117.2.8 static void md4\_process (md4\_context \* ctx, unsigned char data[64]) [static]

Definition at line 76 of file md4.c.

References GET\_UINT32\_LE, P, and md4\_context::state.

Referenced by md4\_update().

#### 8.117.2.9 int md4\_self\_test (int verbose)

Checkup routine.

##### Returns:

0 if successful, or 1 if the test failed

Definition at line 438 of file md4.c.

Referenced by main().

#### 8.117.2.10 void md4\_starts (md4\_context \* ctx)

MD4 context setup.

##### Parameters:

*ctx* context to be initialized

Definition at line 65 of file md4.c.

References md4\_context::state, and md4\_context::total.

Referenced by md4(), md4\_file(), md4\_hmac\_finish(), and md4\_hmac\_starts().



**8.117.2.11 void md4\_update (md4\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

MD4 process buffer.

**Parameters:**

*ctx* MD4 context  
*input* buffer holding the data  
*ilen* length of the input data

Definition at line 182 of file md4.c.

References md4\_context::buffer, md4\_process(), and md4\_context::total.

Referenced by md4(), md4\_file(), md4\_finish(), md4\_hmac\_finish(), md4\_hmac\_starts(), and md4\_hmac\_update().

**8.117.3 Variable Documentation****8.117.3.1 const char \_md4\_src[] = "\_md4\_src" [static]**

Definition at line 370 of file md4.c.

**8.117.3.2 const unsigned char md4\_padding[64] [static]****Initial value:**

```
{
    0x80, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
}
```

Definition at line 223 of file md4.c.

Referenced by md4\_finish().

## 8.118 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/md5.c File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/md5.h"
```

### Defines

- #define [\\_CRT\\_SECURE\\_NO\\_DEPRECATED](#) 1
- #define [GET\\_UINT32\\_LE](#)(n, b, i)
- #define [PUT\\_UINT32\\_LE](#)(n, b, i)
- #define [S](#)(x, n) ((x << n) | ((x & 0xFFFFFFFF) >> (32 - n)))
- #define [P](#)(a, b, c, d, k, s, t)
- #define [F](#)(x, y, z) (z ^ (x & (y ^ z)))
- #define [F](#)(x, y, z) (y ^ (z & (x ^ y)))
- #define [F](#)(x, y, z) (x ^ y ^ z)
- #define [F](#)(x, y, z) (y ^ (x | ~z))

### Functions

- void [md5\\_starts](#) ([md5\\_context](#) \*ctx)  
*MD5 context setup.*
- static void [md5\\_process](#) ([md5\\_context](#) \*ctx, unsigned char data[64])
- void [md5\\_update](#) ([md5\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*MD5 process buffer.*
- void [md5\\_finish](#) ([md5\\_context](#) \*ctx, unsigned char \*output)  
*MD5 final digest.*
- void [md5](#) (unsigned char \*input, [int](#) ilen, unsigned char \*output)  
*Output = MD5( input buffer ).*
- [int](#) [md5\\_file](#) (char \*path, unsigned char \*output)  
*Output = MD5( file contents ).*
- void [md5\\_hmac\\_starts](#) ([md5\\_context](#) \*ctx, unsigned char \*key, [int](#) keylen)  
*MD5 HMAC context setup.*
- void [md5\\_hmac\\_update](#) ([md5\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*MD5 HMAC process buffer.*
- void [md5\\_hmac\\_finish](#) ([md5\\_context](#) \*ctx, unsigned char \*output)  
*MD5 HMAC final digest.*
- void [md5\\_hmac](#) (unsigned char \*key, [int](#) keylen, unsigned char \*input, [int](#) ilen, unsigned char \*output)

*Output = HMAC-MD5( hmac key, input buffer ).*

- `int md5_self_test` (`int verbose`)

*Checkup routine.*

## Variables

- static const unsigned char `md5_padding` [64]
- static const char `_md5_src` [] = "\_md5\_src"

## 8.118.1 Define Documentation

### 8.118.1.1 `#define _CRT_SECURE_NO_DEPRECATED 1`

Definition at line 30 of file md5.c.

### 8.118.1.2 `#define F(x, y, z) (y ^ (x | ~z))`

### 8.118.1.3 `#define F(x, y, z) (x ^ y ^ z)`

### 8.118.1.4 `#define F(x, y, z) (y ^ (z & (x ^ y)))`

### 8.118.1.5 `#define F(x, y, z) (z ^ (x & (y ^ z)))`

### 8.118.1.6 `#define GET_UINT32_LE(n, b, i)`

**Value:**

```
{
    (n) = ( (unsigned long) (b) [(i)      ] )
    | ( (unsigned long) (b) [(i) + 1] << 8 )
    | ( (unsigned long) (b) [(i) + 2] << 16 )
    | ( (unsigned long) (b) [(i) + 3] << 24 );
}
```

Definition at line 42 of file md5.c.

### 8.118.1.7 `#define P(a, b, c, d, k, s, t)`

**Value:**

```
{
    a += F(b, c, d) + X[k] + t; a = S(a, s) + b;
}
```

### 8.118.1.8 `#define PUT_UINT32_LE(n, b, i)`

**Value:**

```

{
    (b) [(i)      ] = (unsigned char) ( (n)      ); \
    (b) [(i) + 1] = (unsigned char) ( (n) >> 8 ); \
    (b) [(i) + 2] = (unsigned char) ( (n) >> 16 ); \
    (b) [(i) + 3] = (unsigned char) ( (n) >> 24 ); \
}

```

Definition at line 52 of file md5.c.

**8.118.1.9** `#define S(x, n) ((x << n) | ((x & 0xFFFFFFFF) >> (32 - n)))`

## 8.118.2 Function Documentation

**8.118.2.1** `void md5 (unsigned char * input, int ilen, unsigned char * output)`

Output = MD5( input buffer ).

### Parameters:

*input* buffer holding the data  
*ilen* length of the input data  
*output* MD5 checksum result

Definition at line 281 of file md5.c.

References md5\_finish(), md5\_starts(), and md5\_update().

Referenced by main(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_md5(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_server\_key\_exchange(), and x509\_hash().

**8.118.2.2** `int md5_file (char * path, unsigned char * output)`

Output = MD5( file contents ).

### Parameters:

*path* input file name  
*output* MD5 checksum result

### Returns:

0 if successful, 1 if fopen failed, or 2 if fread failed

Definition at line 296 of file md5.c.

References f, md5\_finish(), md5\_starts(), and md5\_update().

Referenced by md5\_wrapper().

**8.118.2.3** `void md5_finish (md5_context * ctx, unsigned char * output)`

MD5 final digest.

### Parameters:

*ctx* MD5 context

*output* MD5 checksum result

Definition at line 253 of file md5.c.

References md5\_padding(), md5\_update(), PUT\_UINT32\_LE, md5\_context::state, and md5\_context::total.

Referenced by md5(), md5\_file(), md5\_hmac\_finish(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_md5(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_server\_key\_exchange(), and x509\_des3\_decrypt().

#### 8.118.2.4 void md5\_hmac (unsigned char \* *key*, int *keylen*, unsigned char \* *input*, int *ilen*, unsigned char \* *output*)

Output = HMAC-MD5( hmac key, input buffer ).

##### Parameters:

*key* HMAC secret key

*keylen* length of the HMAC key

*input* buffer holding the data

*ilen* length of the input data

*output* HMAC-MD5 result

Definition at line 376 of file md5.c.

References md5\_hmac\_finish(), md5\_hmac\_starts(), and md5\_hmac\_update().

Referenced by ssl\_decrypt\_buf(), ssl\_encrypt\_buf(), and tls1\_prf().

#### 8.118.2.5 void md5\_hmac\_finish (md5\_context \* *ctx*, unsigned char \* *output*)

MD5 HMAC final digest.

##### Parameters:

*ctx* HMAC context

*output* MD5 HMAC checksum result

Definition at line 360 of file md5.c.

References md5\_finish(), md5\_starts(), md5\_update(), and md5\_context::opad.

Referenced by md5\_hmac().

#### 8.118.2.6 void md5\_hmac\_starts (md5\_context \* *ctx*, unsigned char \* *key*, int *keylen*)

MD5 HMAC context setup.

##### Parameters:

*ctx* HMAC context to be initialized

*key* HMAC secret key

*keylen* length of the HMAC key

Definition at line 328 of file md5.c.

References md5\_context::ipad, md5\_starts(), md5\_update(), and md5\_context::opad.

Referenced by md5\_hmac().

#### 8.118.2.7 void md5\_hmac\_update (md5\_context \* ctx, unsigned char \* input, int ilen)

MD5 HMAC process buffer.

##### Parameters:

*ctx* HMAC context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 351 of file md5.c.

References md5\_update().

Referenced by md5\_hmac().

#### 8.118.2.8 static void md5\_process (md5\_context \* ctx, unsigned char data[64]) [static]

Definition at line 75 of file md5.c.

References GET\_UINT32\_LE, P, and md5\_context::state.

Referenced by md5\_update().

#### 8.118.2.9 int md5\_self\_test (int verbose)

Checkup routine.

##### Returns:

0 if successful, or 1 if the test failed

Definition at line 457 of file md5.c.

Referenced by main().

#### 8.118.2.10 void md5\_starts (md5\_context \* ctx)

MD5 context setup.

##### Parameters:

*ctx* context to be initialized

Definition at line 64 of file md5.c.

References md5\_context::state, and md5\_context::total.

Referenced by md5(), md5\_file(), md5\_hmac\_finish(), md5\_hmac\_starts(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_md5(), ssl\_parse\_client\_hello(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_client\_hello(), ssl\_write\_server\_key\_exchange(), and x509\_des3\_decrypt().

**8.118.2.11 void md5\_update (md5\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

MD5 process buffer.

**Parameters:**

*ctx* MD5 context  
*input* buffer holding the data  
*ilen* length of the input data

Definition at line 201 of file md5.c.

References md5\_context::buffer, md5\_process(), and md5\_context::total.

Referenced by md5(), md5\_file(), md5\_finish(), md5\_hmac\_finish(), md5\_hmac\_starts(), md5\_hmac\_update(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_md5(), ssl\_parse\_client\_hello(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_record(), ssl\_write\_server\_key\_exchange(), and x509\_des3\_decrypt().

**8.118.3 Variable Documentation****8.118.3.1 const char \_md5\_src[] = "\_md5\_src" [static]**

Definition at line 389 of file md5.c.

**8.118.3.2 const unsigned char md5\_padding[64] [static]****Initial value:**

```
{
    0x80, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
}
```

Definition at line 242 of file md5.c.

Referenced by md5\_finish().

## 8.119 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/net.c File Reference

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/time.h>
#include <unistd.h>
#include <signal.h>
#include <fcntl.h>
#include <netdb.h>
#include <errno.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include "xyssl/net.h"
```

### Defines

- `#define _CRT_SECURE_NO_DEPRECATED 1`

### Functions

- `int net_connect (int *fd, char *host, unsigned int port)`  
*Initiate a TCP connection with host:port.*
- `int net_bind (int *fd, char *bind_ip, unsigned int port)`  
*Create a listening socket on bind\_ip:port. If bind\_ip == NULL, all interfaces are binded.*
- `static int net_is_blocking (void)`
- `int net_accept (int bind_fd, int *client_fd, unsigned char client_ip[4])`  
*Accept a connection from a remote client.*
- `int net_set_block (int fd)`
- `int net_set_nonblock (int fd)`  
*Set the socket non-blocking.*
- `void net_usleep (unsigned long usec)`  
*Portable usleep helper.*
- `int net_recv (int fd, unsigned char *buf, int *len)`



*Read at most 'len' characters. len is updated to reflect the actual number of characters read.*

- `int net_send (int fd, unsigned char *buf, int *len)`

*Write at most 'len' characters. len is updated to reflect the number of characters \_not\_ written.*

- `void net_close (int fd)`

*Gracefully shutdown the connection.*

## 8.119.1 Define Documentation

### 8.119.1.1 #define \_CRT\_SECURE\_NO\_DEPRECATED 1

Definition at line 25 of file net.c.

## 8.119.2 Function Documentation

### 8.119.2.1 int net\_accept (int bind\_fd, int \* client\_fd, unsigned char client\_ip[4])

Accept a connection from a remote client.

#### Returns:

0 if successful, ERR\_NET\_ACCEPT\_FAILED, or ERR\_NET\_WOULD\_BLOCK is bind\_fd was set to non-blocking and accept() is blocking.

Definition at line 203 of file net.c.

References ERR\_NET\_ACCEPT\_FAILED, ERR\_NET\_WOULD\_BLOCK, int, and net\_is\_blocking().

Referenced by main().

### 8.119.2.2 int net\_bind (int \* fd, char \* bind\_ip, unsigned int port)

Create a listening socket on bind\_ip:port. If bind\_ip == NULL, all interfaces are binded.

#### Returns:

0 if successful, or one of: ERR\_NET\_SOCKET\_FAILED, ERR\_NET\_BIND\_FAILED, ERR\_NET\_LISTEN\_FAILED

Definition at line 115 of file net.c.

References ERR\_NET\_BIND\_FAILED, ERR\_NET\_LISTEN\_FAILED, and ERR\_NET\_SOCKET\_FAILED.

Referenced by main().

### 8.119.2.3 void net\_close (int fd)

Gracefully shutdown the connection.

Definition at line 340 of file net.c.

Referenced by main().

**8.119.2.4 int net\_connect (int \*fd, char \*host, unsigned int port)**

Initiate a TCP connection with host:port.

**Returns:**

0 if successful, or one of: ERR\_NET\_SOCKET\_FAILED, ERR\_NET\_UNKNOWN\_HOST, ERR\_NET\_CONNECT\_FAILED

Definition at line 70 of file net.c.

References ERR\_NET\_CONNECT\_FAILED, ERR\_NET\_SOCKET\_FAILED, and ERR\_NET\_UNKNOWN\_HOST.

Referenced by main().

**8.119.2.5 static int net\_is\_blocking (void) [static]**

Definition at line 181 of file net.c.

Referenced by net\_accept(), net\_recv(), and net\_send().

**8.119.2.6 int net\_recv (int fd, unsigned char \*buf, int \*len)**

Read at most 'len' characters. len is updated to reflect the actual number of characters read.

**Returns:**

0 if successful, ERR\_NET\_CONN\_RESET if the connection was closed from the other side, or ERR\_NET\_WOULD\_BLOCK if read() is blocking.

Definition at line 268 of file net.c.

References ERR\_NET\_CONN\_RESET, ERR\_NET\_RECV\_FAILED, ERR\_NET\_WOULD\_BLOCK, and net\_is\_blocking().

Referenced by main(), and ssl\_parse\_client\_hello().

**8.119.2.7 int net\_send (int fd, unsigned char \*buf, int \*len)**

Write at most 'len' characters. len is updated to reflect the number of characters \_not\_ written.

**Returns:**

0 if successful, ERR\_NET\_CONN\_RESET if the connection was closed from the other side, or ERR\_NET\_WOULD\_BLOCK if write() is blocking.

Definition at line 305 of file net.c.

References ERR\_NET\_CONN\_RESET, ERR\_NET\_SEND\_FAILED, ERR\_NET\_WOULD\_BLOCK, and net\_is\_blocking().

Referenced by main(), ssl\_flush\_output(), and ssl\_write\_record().

**8.119.2.8 int net\_set\_block (int fd)**

Definition at line 234 of file net.c.

#### 8.119.2.9 `int net_set_nonblock (int fd)`

Set the socket non-blocking.

##### **Returns:**

0 if successful, or 1 if the operation failed

Definition at line 244 of file net.c.

#### 8.119.2.10 `void net_usleep (unsigned long usec)`

Portable usleep helper.

##### **Note:**

Real amount of time slept will not be less than select()'s timeout granularity (typically, 10ms).

Definition at line 257 of file net.c.

## 8.120 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/rsa.c File Reference

```
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include "xyssl/rsa.h"
```

### Defines

- `#define _CRT_SECURE_NO_DEPRECATED 1`

### Functions

- `int rsa_gen_key (rsa_context *ctx, int nbits, int exponent, int(*rng_f)(void *), void *rng_d)`  
*Generate an RSA keypair.*
- `int rsa_read_public (rsa_context *ctx, FILE *f)`  
*Read the public key from a file.*
- `int rsa_read_private (rsa_context *ctx, FILE *f)`  
*Read the private key from a file.*
- `int rsa_write_public (rsa_context *ctx, FILE *f)`  
*Write the public key into a file.*
- `int rsa_write_private (rsa_context *ctx, FILE *f)`  
*Write the private key into a file.*
- `int rsa_public (rsa_context *ctx, unsigned char *input, int ilen, unsigned char *output, int olen)`  
*Perform an RSA public key operation.*
- `int rsa_private (rsa_context *ctx, unsigned char *input, int ilen, unsigned char *output, int olen)`  
*Perform an RSA private key operation.*
- `int rsa_check_pubkey (rsa_context *ctx)`  
*Return 0 if the public key is valid, or ERR\_RSA\_KEY\_CHECK\_FAILED.*
- `int rsa_check_privkey (rsa_context *ctx)`  
*Return 0 if the private key is valid, or ERR\_RSA\_KEY\_CHECK\_FAILED.*
- `int rsa_pkcs1_encrypt (rsa_context *ctx, unsigned char *input, int ilen, unsigned char *output, int olen)`  
*Add the PKCS#1 v1.5 padding and do a public RSA.*
- `int rsa_pkcs1_decrypt (rsa_context *ctx, unsigned char *input, int ilen, unsigned char *output, int *olen)`

*Do a private RSA, removes the PKCS#1 v1.5 padding.*

- `int rsa_pkcs1_sign` (`rsa_context` \*ctx, `int` alg\_id, unsigned char \*hash, `int` hashlen, unsigned char \*sig, `int` siglen)

*Perform a private RSA to sign a message digest.*

- `int rsa_pkcs1_verify` (`rsa_context` \*ctx, `int` alg\_id, unsigned char \*hash, `int` hashlen, unsigned char \*sig, `int` siglen)

*Perform a public RSA and check the message digest.*

- `void rsa_free` (`rsa_context` \*ctx)

*Free the components of an RSA key.*

- `int rsa_self_test` (`int` verbose)

*Checkup routine.*

## 8.120.1 Define Documentation

### 8.120.1.1 #define \_CRT\_SECURE\_NO\_DEPRECATED 1

Definition at line 31 of file rsa.c.

## 8.120.2 Function Documentation

### 8.120.2.1 int rsa\_check\_privkey (rsa\_context \* ctx)

Return 0 if the private key is valid, or ERR\_RSA\_KEY\_CHECK\_FAILED.

Definition at line 329 of file rsa.c.

References `CHK`, `rsa_context::E`, `ERR_RSA_KEY_CHK_FAILED`, `mpi_cmp_int()`, `mpi_cmp_mpi()`, `mpi_free()`, `mpi_gcd()`, `mpi_init()`, `mpi_mul_mpi()`, `mpi_sub_int()`, `rsa_context::N`, `rsa_context::P`, and `rsa_context::Q`.

Referenced by `x509_parse_key()`.

### 8.120.2.2 int rsa\_check\_pubkey (rsa\_context \* ctx)

Return 0 if the public key is valid, or ERR\_RSA\_KEY\_CHECK\_FAILED.

Definition at line 309 of file rsa.c.

References `rsa_context::E`, `ERR_RSA_KEY_CHK_FAILED`, `mpi_msb()`, `rsa_context::N`, and `mpi::p`.

Referenced by `x509_add_certs()`.

### 8.120.2.3 void rsa\_free (rsa\_context \* ctx)

Free the components of an RSA key.

Definition at line 572 of file rsa.c.

References `rsa_context::D`, `rsa_context::DP`, `rsa_context::DQ`, `rsa_context::E`, `mpi_free()`, `rsa_context::N`, `rsa_context::P`, `rsa_context::Q`, `rsa_context::QP`, `rsa_context::RN`, `rsa_context::RP`, and `rsa_context::RQ`.

Referenced by `main()`, `rsa_gen_key()`, `rsa_read_private()`, `rsa_read_public()`, `x509_free_cert()`, and `x509_parse_key()`.

#### 8.120.2.4 `int rsa_gen_key (rsa_context * ctx, int nbits, int exponent, int(*) (void *) rng_f, void * rng_d)`

Generate an RSA keypair.

##### Parameters:

*ctx* RSA context to be initialized  
*nbits* size of the public key in bits  
*exponent* public exponent (e.g., 65537)  
*rng\_f* points to the RNG function  
*rng\_d* points to the RNG data

##### Returns:

0 if successful, or an `ERR_RSA_XXX` error code

Definition at line 44 of file `rsa.c`.

References `CHK`, `rsa_context::D`, `rsa_context::DP`, `rsa_context::DQ`, `rsa_context::E`, `ERR_RSA_BAD_INPUT_DATA`, `ERR_RSA_KEY_GEN_FAILED`, `rsa_context::len`, `mpi_cmp_int()`, `mpi_cmp_mpi()`, `mpi_free()`, `mpi_gcd()`, `mpi_gen_prime()`, `mpi_init()`, `mpi_inv_mod()`, `mpi_lset()`, `mpi_mod_mpi()`, `mpi_msb()`, `mpi_mul_mpi()`, `mpi_sub_int()`, `mpi_swap()`, `rsa_context::N`, `rsa_context::P`, `rsa_context::Q`, `rsa_context::QP`, and `rsa_free()`.

Referenced by `main()`.

#### 8.120.2.5 `int rsa_pkcs1_decrypt (rsa_context * ctx, unsigned char * input, int ilen, unsigned char * output, int * olen)`

Do a private RSA, removes the PKCS#1 v1.5 padding.

##### Parameters:

*ctx* RSA context  
*input* buffer holding the encrypted data  
*ilen* must be the same as the modulus size  
*output* buffer that will hold the plaintext  
*olen* size of output buffer, will be updated to contain the length of the plaintext

##### Returns:

0 if successful, or an `ERR_RSA_XXX` error code

Definition at line 388 of file `rsa.c`.

References `ERR_RSA_BAD_INPUT_DATA`, `ERR_RSA_INVALID_PADDING`, `int`, `rsa_context::len`, `RSA_CRYPT`, and `rsa_private()`.

Referenced by `ssl_parse_client_key_exchange()`.

**8.120.2.6 int rsa\_pkcs1\_encrypt (rsa\_context \* ctx, unsigned char \* input, int ilen, unsigned char \* output, int olen)**

Add the PKCS#1 v1.5 padding and do a public RSA.

**Parameters:**

*ctx* RSA context  
*input* buffer holding the data to be encrypted  
*ilen* length of the plaintext; cannot be longer than the modulus, minus 3+8 for padding  
*output* buffer that will hold the ciphertext  
*olen* must be the same as the modulus size (for example, 128 if RSA-1024 is used)

**Returns:**

0 if successful, or an ERR\_RSA\_XXX error code

Definition at line 358 of file rsa.c.

References ERR\_RSA\_BAD\_INPUT\_DATA, rsa\_context::len, RSA\_CRYPT, and rsa\_public().

Referenced by ssl\_write\_client\_key\_exchange().

**8.120.2.7 int rsa\_pkcs1\_sign (rsa\_context \* ctx, int alg\_id, unsigned char \* hash, int hashlen, unsigned char \* sig, int siglen)**

Perform a private RSA to sign a message digest.

**Parameters:**

*ctx* RSA context  
*alg\_id* RSA\_RAW, RSA\_MD2/4/5 or RSA\_SHA1  
*hash* buffer holding the message digest  
*hashlen* message digest length  
*sig* buffer that will hold the ciphertext  
*siglen* must be the same as the modulus size (for example, 128 if RSA-1024 is used)

**Returns:**

0 if the signing operation was successful, or an ERR\_RSA\_XXX error code

Definition at line 426 of file rsa.c.

References ASN1\_HASH\_MDX, ASN1\_HASH\_SHA1, ERR\_RSA\_BAD\_INPUT\_DATA, rsa\_context::len, RSA\_MD2, RSA\_MD4, RSA\_MD5, rsa\_private(), RSA\_RAW, RSA\_SHA1, and RSA\_SIGN.

Referenced by main(), ssl\_write\_certificate\_verify(), and ssl\_write\_server\_key\_exchange().

**8.120.2.8 int rsa\_pkcs1\_verify (rsa\_context \* ctx, int alg\_id, unsigned char \* hash, int hashlen, unsigned char \* sig, int siglen)**

Perform a public RSA and check the message digest.

**Parameters:**

*ctx* points to an RSA public key  
*alg\_id* RSA\_RAW, RSA\_MD2/4/5 or RSA\_SHA1  
*hash* buffer holding the message digest  
*hashlen* message digest length  
*sig* buffer holding the ciphertext  
*siglen* must be the same as the modulus size

**Returns:**

0 if the verify operation was successful, or an ERR\_RSA\_XXX error code

Definition at line 502 of file rsa.c.

References ASN1\_HASH\_MDX, ASN1\_HASH\_SHA1, ERR\_RSA\_BAD\_INPUT\_DATA, ERR\_RSA\_INVALID\_PADDING, ERR\_RSA\_VERIFY\_FAILED, int, rsa\_context::len, RSA\_MD2, RSA\_MD4, RSA\_MD5, rsa\_public(), RSA\_RAW, RSA\_SHA1, and RSA\_SIGN.

Referenced by main(), ssl\_parse\_certificate\_verify(), ssl\_parse\_server\_key\_exchange(), and x509\_verify\_cert().

#### 8.120.2.9 **int rsa\_private (rsa\_context \* ctx, unsigned char \* input, int ilen, unsigned char \* output, int olen)**

Perform an RSA private key operation.

**Returns:**

0 if successful, or an ERR\_RSA\_XXX error code

**Note:**

This function does not take care of message padding: both ilen and olen must be equal to the modulus size (ctx->len). Also, be sure to set input[0] = 0.

Definition at line 248 of file rsa.c.

References CHK, rsa\_context::D, rsa\_context::DP, rsa\_context::DQ, ERR\_RSA\_BAD\_INPUT\_DATA, ERR\_RSA\_PRIVATE\_FAILED, rsa\_context::len, mpi\_add\_mpi(), mpi\_cmp\_mpi(), mpi\_exp\_mod(), mpi\_free(), mpi\_init(), mpi\_mod\_mpi(), mpi\_mul\_mpi(), mpi\_read\_binary(), mpi\_sub\_mpi(), mpi\_write\_binary(), rsa\_context::N, rsa\_context::P, rsa\_context::Q, rsa\_context::QP, rsa\_context::RN, rsa\_context::RP, and rsa\_context::RQ.

Referenced by main(), rsa\_pkcs1\_decrypt(), and rsa\_pkcs1\_sign().

#### 8.120.2.10 **int rsa\_public (rsa\_context \* ctx, unsigned char \* input, int ilen, unsigned char \* output, int olen)**

Perform an RSA public key operation.

**Returns:**

0 if successful, or an ERR\_RSA\_XXX error code



**Note:**

This function does not take care of message padding: both *ilen* and *olen* must be equal to the modulus size (*ctx->len*). Also, be sure to set *input[0] = 0*.

Definition at line 212 of file *rsa.c*.

References *CHK*, *rsa\_context::E*, *ERR\_RSA\_BAD\_INPUT\_DATA*, *ERR\_RSA\_PUBLIC\_FAILED*, *rsa\_context::len*, *mpi\_cmp\_mpi()*, *mpi\_exp\_mod()*, *mpi\_free()*, *mpi\_init()*, *mpi\_read\_binary()*, *mpi\_write\_binary()*, *rsa\_context::N*, and *rsa\_context::RN*.

Referenced by *main()*, *rsa\_pkcs1\_encrypt()*, and *rsa\_pkcs1\_verify()*.

**8.120.2.11 int rsa\_read\_private (rsa\_context \* ctx, FILE \* f)**

Read the private key from a file.

**Parameters:**

*ctx* RSA context to be initialized

*f* Handle of the source file

**Returns:**

0 if successful, or an *ERR\_RSA\_XXX* error code

Definition at line 139 of file *rsa.c*.

References *CHK*, *rsa\_context::D*, *rsa\_context::DP*, *rsa\_context::DQ*, *rsa\_context::E*, *ERR\_RSA\_KEY\_RD\_FAILED*, *rsa\_context::len*, *mpi\_msb()*, *mpi\_read\_file()*, *rsa\_context::N*, *rsa\_context::P*, *rsa\_context::Q*, *rsa\_context::QP*, and *rsa\_free()*.

Referenced by *main()*.

**8.120.2.12 int rsa\_read\_public (rsa\_context \* ctx, FILE \* f)**

Read the public key from a file.

**Parameters:**

*ctx* RSA context to be initialized

*f* Handle of the source file

**Returns:**

0 if successful, or an *ERR\_RSA\_XXX* error code

Definition at line 114 of file *rsa.c*.

References *CHK*, *rsa\_context::E*, *ERR\_RSA\_KEY\_RD\_FAILED*, *rsa\_context::len*, *mpi\_msb()*, *mpi\_read\_file()*, *rsa\_context::N*, and *rsa\_free()*.

Referenced by *main()*.

**8.120.2.13 int rsa\_self\_test (int *verbose*)**

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 686 of file rsa.c.

Referenced by main().

**8.120.2.14 int rsa\_write\_private (rsa\_context \* *ctx*, FILE \* *f*)**

Write the private key into a file.

**Parameters:**

*ctx* RSA context holding the key

*f* Handle of the destination file

**Returns:**

0 if successful, or an ERR\_RSA\_XXX error code

Definition at line 188 of file rsa.c.

References CHK, rsa\_context::D, rsa\_context::DP, rsa\_context::DQ, rsa\_context::E, ERR\_RSA\_KEY\_WR\_FAILED, mpi\_write\_file(), rsa\_context::N, rsa\_context::P, rsa\_context::Q, and rsa\_context::QP.

Referenced by main().

**8.120.2.15 int rsa\_write\_public (rsa\_context \* *ctx*, FILE \* *f*)**

Write the public key into a file.

**Parameters:**

*ctx* RSA context holding the key

*f* Handle of the destination file

**Returns:**

0 if successful, or an ERR\_RSA\_XXX error code

Definition at line 170 of file rsa.c.

References CHK, rsa\_context::E, ERR\_RSA\_KEY\_WR\_FAILED, mpi\_write\_file(), and rsa\_context::N.

Referenced by main().

## 8.121 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/sha1.c File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/sha1.h"
```

### Defines

- #define [\\_CRT\\_SECURE\\_NO\\_DEPRECATED](#) 1
- #define [GET\\_UINT32\\_BE](#)(n, b, i)
- #define [PUT\\_UINT32\\_BE](#)(n, b, i)
- #define [S](#)(x, n) ((x << n) | ((x & 0xFFFFFFFF) >> (32 - n)))
- #define [R](#)(t)
- #define [P](#)(a, b, c, d, e, x)
- #define [F](#)(x, y, z) (z ^ (x & (y ^ z)))
- #define [K](#) 0x5A827999
- #define [F](#)(x, y, z) (x ^ y ^ z)
- #define [K](#) 0x6ED9EBA1
- #define [F](#)(x, y, z) ((x & y) | (z & (x | y)))
- #define [K](#) 0x8F1BBCDC
- #define [F](#)(x, y, z) (x ^ y ^ z)
- #define [K](#) 0xCA62C1D6

### Functions

- void [sha1\\_starts](#) ([sha1\\_context](#) \*ctx)  
*SHA-1 context setup.*
- static void [sha1\\_process](#) ([sha1\\_context](#) \*ctx, unsigned char data[64])
- void [sha1\\_update](#) ([sha1\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*SHA-1 process buffer.*
- void [sha1\\_finish](#) ([sha1\\_context](#) \*ctx, unsigned char \*output)  
*SHA-1 final digest.*
- void [sha1](#) (unsigned char \*input, [int](#) ilen, unsigned char \*output)  
*Output = SHA-1( input buffer ).*
- [int](#) [sha1\\_file](#) (char \*path, unsigned char \*output)  
*Output = SHA-1( file contents ).*
- void [sha1\\_hmac\\_starts](#) ([sha1\\_context](#) \*ctx, unsigned char \*key, [int](#) keylen)  
*SHA-1 HMAC context setup.*
- void [sha1\\_hmac\\_update](#) ([sha1\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*SHA-1 HMAC process buffer.*

- void `sha1_hmac_finish` (`sha1_context` \*ctx, unsigned char \*output)  
*SHA-1 HMAC final digest.*
- void `sha1_hmac` (unsigned char \*key, int keylen, unsigned char \*input, int ilen, unsigned char \*output)  
*Output = HMAC-SHA-1( hmac key, input buffer ).*
- int `sha1_self_test` (int verbose)  
*Checkup routine.*

## Variables

- static const unsigned char `sha1_padding` [64]
- static const char `_sha1_src` [] = "\_sha1\_src"

### 8.121.1 Define Documentation

#### 8.121.1.1 #define \_CRT\_SECURE\_NO\_DEPRECATED 1

Definition at line 30 of file sha1.c.

#### 8.121.1.2 #define F(x, y, z) (x ^ y ^ z)

#### 8.121.1.3 #define F(x, y, z) ((x & y) | (z & (x | y)))

#### 8.121.1.4 #define F(x, y, z) (x ^ y ^ z)

#### 8.121.1.5 #define F(x, y, z) (z ^ (x & (y ^ z)))

#### 8.121.1.6 #define GET\_UINT32\_BE(n, b, i)

Value:

```
{
    (n) = ( (unsigned long) (b) [(i)      ] << 24 ) \
    | ( (unsigned long) (b) [(i) + 1] << 16 ) \
    | ( (unsigned long) (b) [(i) + 2] <<  8 ) \
    | ( (unsigned long) (b) [(i) + 3]      ) ; \
}
```

Definition at line 42 of file sha1.c.

#### 8.121.1.7 #define K 0xCA62C1D6

#### 8.121.1.8 #define K 0x8F1BBCDC

#### 8.121.1.9 #define K 0x6ED9EBA1

#### 8.121.1.10 #define K 0x5A827999

Referenced by `sha4_process()`.

**8.121.1.11 #define P(a, b, c, d, e, x)****Value:**

```
{
    e += S(a, 5) + F(b, c, d) + K + x; b = S(b, 30);
}
```

**8.121.1.12 #define PUT\_UINT32\_BE(n, b, i)****Value:**

```
{
    (b)[(i)] = (unsigned char) ((n) >> 24);
    (b)[(i) + 1] = (unsigned char) ((n) >> 16);
    (b)[(i) + 2] = (unsigned char) ((n) >> 8);
    (b)[(i) + 3] = (unsigned char) (n);
}
```

Definition at line 52 of file sha1.c.

**8.121.1.13 #define R(t)****Value:**

```
(
    temp = W[(t - 3) & 0x0F] ^ W[(t - 8) & 0x0F] ^
           W[(t - 14) & 0x0F] ^ W[t & 0x0F],
    (W[t & 0x0F] = S(temp, 1))
)
```

Referenced by mpi\_is\_prime(), sha1\_process(), and sha2\_process().

**8.121.1.14 #define S(x, n) ((x << n) | ((x & 0xFFFFFFFF) >> (32 - n)))****8.121.2 Function Documentation****8.121.2.1 void sha1 (unsigned char \* *input*, int *ilen*, unsigned char \* *output*)**

Output = SHA-1( input buffer ).

**Parameters:**

- input* buffer holding the data
- ilen* length of the input data
- output* SHA-1 checksum result

Definition at line 316 of file sha1.c.

References sha1\_finish(), sha1\_starts(), and sha1\_update().

Referenced by main(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_get\_session(), ssl\_mac\_sha1(), ssl\_parse\_server\_key\_exchange(), ssl\_set\_session(), ssl\_write\_server\_key\_exchange(), and x509\_hash().

**8.121.2.2 int sha1\_file (char \* *path*, unsigned char \* *output*)**

Output = SHA-1( file contents ).

**Parameters:**

*path* input file name

*output* SHA-1 checksum result

**Returns:**

0 if successful, 1 if fopen failed, or 2 if fread failed

Definition at line 331 of file sha1.c.

References f, sha1\_finish(), sha1\_starts(), and sha1\_update().

Referenced by main(), and sha1\_wrapper().

**8.121.2.3 void sha1\_finish (sha1\_context \* *ctx*, unsigned char \* *output*)**

SHA-1 final digest.

**Parameters:**

*ctx* SHA-1 context

*output* SHA-1 checksum result

Definition at line 287 of file sha1.c.

References PUT\_UINT32\_BE, sha1\_padding, sha1\_update(), sha1\_context::state, and sha1\_context::total.

Referenced by sha1(), sha1\_file(), sha1\_hmac\_finish(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_sha1(), ssl\_parse\_server\_key\_exchange(), and ssl\_write\_server\_key\_exchange().

**8.121.2.4 void sha1\_hmac (unsigned char \* *key*, int *keylen*, unsigned char \* *input*, int *ilen*, unsigned char \* *output*)**

Output = HMAC-SHA-1( hmac key, input buffer ).

**Parameters:**

*key* HMAC secret key

*keylen* length of the HMAC key

*input* buffer holding the data

*ilen* length of the input data

*output* HMAC-SHA-1 result

Definition at line 411 of file sha1.c.

References sha1\_hmac\_finish(), sha1\_hmac\_starts(), and sha1\_hmac\_update().

Referenced by ssl\_decrypt\_buf(), ssl\_encrypt\_buf(), and tls1\_prf().

**8.121.2.5 void sha1\_hmac\_finish (sha1\_context \* *ctx*, unsigned char \* *output*)**

SHA-1 HMAC final digest.

**Parameters:**

*ctx* HMAC context

*output* SHA-1 HMAC checksum result

Definition at line 395 of file sha1.c.

References sha1\_context::opad, sha1\_finish(), sha1\_starts(), and sha1\_update().

Referenced by sha1\_hmac().

**8.121.2.6 void sha1\_hmac\_starts (sha1\_context \* *ctx*, unsigned char \* *key*, int *keylen*)**

SHA-1 HMAC context setup.

**Parameters:**

*ctx* HMAC context to be initialized

*key* HMAC secret key

*keylen* length of the HMAC key

Definition at line 363 of file sha1.c.

References sha1\_context::ipad, sha1\_context::opad, sha1\_starts(), and sha1\_update().

Referenced by sha1\_hmac().

**8.121.2.7 void sha1\_hmac\_update (sha1\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

SHA-1 HMAC process buffer.

**Parameters:**

*ctx* HMAC context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 386 of file sha1.c.

References sha1\_update().

Referenced by sha1\_hmac().

**8.121.2.8 static void sha1\_process (sha1\_context \* *ctx*, unsigned char *data*[64]) [static]**

Definition at line 76 of file sha1.c.

References GET\_UINT32\_BE, P, R, and sha1\_context::state.

Referenced by sha1\_update().

### 8.121.2.9 `int sha1_self_test (int verbose)`

Checksum routine.

#### Returns:

0 if successful, or 1 if the test failed

Definition at line 492 of file sha1.c.

Referenced by main().

### 8.121.2.10 `void sha1_starts (sha1_context * ctx)`

SHA-1 context setup.

#### Parameters:

*ctx* context to be initialized

Definition at line 64 of file sha1.c.

References sha1\_context::state, and sha1\_context::total.

Referenced by sha1(), sha1\_file(), sha1\_hmac\_finish(), sha1\_hmac\_starts(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_sha1(), ssl\_parse\_client\_hello(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_client\_hello(), and ssl\_write\_server\_key\_exchange().

### 8.121.2.11 `void sha1_update (sha1_context * ctx, unsigned char * input, int ilen)`

SHA-1 process buffer.

#### Parameters:

*ctx* SHA-1 context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 235 of file sha1.c.

References sha1\_context::buffer, sha1\_process(), and sha1\_context::total.

Referenced by sha1(), sha1\_file(), sha1\_finish(), sha1\_hmac\_finish(), sha1\_hmac\_starts(), sha1\_hmac\_update(), ssl\_calc\_verify(), ssl\_derive\_keys(), ssl\_mac\_sha1(), ssl\_parse\_client\_hello(), ssl\_parse\_server\_key\_exchange(), ssl\_write\_record(), and ssl\_write\_server\_key\_exchange().

## 8.121.3 Variable Documentation

### 8.121.3.1 `const char _sha1_src[] = "_sha1_src" [static]`

Definition at line 424 of file sha1.c.



**8.121.3.2 const unsigned char sha1\_padding[64] [static]****Initial value:**

```
{  
    0x80, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0  
}
```

Definition at line 276 of file sha1.c.

Referenced by sha1\_finish().

## 8.122 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/sha2.c File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/sha2.h"
```

### Defines

- #define [\\_CRT\\_SECURE\\_NO\\_DEPRECATED](#) 1
- #define [GET\\_UINT32\\_BE](#)(n, b, i)
- #define [PUT\\_UINT32\\_BE](#)(n, b, i)
- #define [SHR](#)(x, n) ((x & 0xFFFFFFFF) >> n)
- #define [ROTR](#)(x, n) (SHR(x,n) | (x << (32 - n)))
- #define [S0](#)(x) (ROTR(x, 7) ^ ROTR(x,18) ^ SHR(x, 3))
- #define [S1](#)(x) (ROTR(x,17) ^ ROTR(x,19) ^ SHR(x,10))
- #define [S2](#)(x) (ROTR(x, 2) ^ ROTR(x,13) ^ ROTR(x,22))
- #define [S3](#)(x) (ROTR(x, 6) ^ ROTR(x,11) ^ ROTR(x,25))
- #define [F0](#)(x, y, z) ((x & y) | (z & (x | y)))
- #define [F1](#)(x, y, z) (z ^ (x & (y ^ z)))
- #define [R](#)(t)
- #define [P](#)(a, b, c, d, e, [f](#), g, h, x, [K](#))

### Functions

- void [sha2\\_starts](#) ([sha2\\_context](#) \*ctx, [int](#) is224)  
*SHA-256 context setup.*
- static void [sha2\\_process](#) ([sha2\\_context](#) \*ctx, unsigned char data[64])
- void [sha2\\_update](#) ([sha2\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*SHA-256 process buffer.*
- void [sha2\\_finish](#) ([sha2\\_context](#) \*ctx, unsigned char \*output)  
*SHA-256 final digest.*
- void [sha2](#) (unsigned char \*input, [int](#) ilen, unsigned char \*output, [int](#) is224)  
*Output = SHA-256( input buffer ).*
- [int](#) [sha2\\_file](#) (char \*path, unsigned char \*output, [int](#) is224)  
*Output = SHA-256( file contents ).*
- void [sha2\\_hmac\\_starts](#) ([sha2\\_context](#) \*ctx, [int](#) is224, unsigned char \*key, [int](#) keylen)  
*SHA-256 HMAC context setup.*
- void [sha2\\_hmac\\_update](#) ([sha2\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*SHA-256 HMAC process buffer.*
- void [sha2\\_hmac\\_finish](#) ([sha2\\_context](#) \*ctx, unsigned char \*output)

*SHA-256 HMAC final digest.*

- void `sha2_hmac` (unsigned char \*key, int keylen, unsigned char \*input, int ilen, unsigned char \*output, int is224)

*Output = HMAC-SHA-256( hmac key, input buffer ).*

- int `sha2_self_test` (int verbose)

*Checkup routine.*

## Variables

- static const unsigned char `sha2_padding` [64]
- static const char `_sha2_src` [] = "\_sha2\_src"

## 8.122.1 Define Documentation

### 8.122.1.1 #define \_CRT\_SECURE\_NO\_DEPRECATED 1

Definition at line 30 of file sha2.c.

### 8.122.1.2 #define F0(x, y, z) ((x & y) | (z & (x | y)))

### 8.122.1.3 #define F1(x, y, z) (z ^ (x & (y ^ z)))

### 8.122.1.4 #define GET\_UINT32\_BE(n, b, i)

**Value:**

```
{
    (n) = ( (unsigned long) (b)[(i)      ] << 24 ) \
    | ( (unsigned long) (b)[(i) + 1] << 16 ) \
    | ( (unsigned long) (b)[(i) + 2] <<  8 ) \
    | ( (unsigned long) (b)[(i) + 3]      ) ; \
}
```

Definition at line 42 of file sha2.c.

### 8.122.1.5 #define P(a, b, c, d, e, f, g, h, x, K)

**Value:**

```
{
    temp1 = h + S3(e) + F1(e, f, g) + K + x; \
    temp2 = S2(a) + F0(a, b, c); \
    d += temp1; h = temp1 + temp2; \
}
```

**8.122.1.6 #define PUT\_UINT32\_BE(n, b, i)****Value:**

```

{
    (b)[(i)] = (unsigned char) ((n) >> 24); \
    (b)[(i) + 1] = (unsigned char) ((n) >> 16); \
    (b)[(i) + 2] = (unsigned char) ((n) >> 8); \
    (b)[(i) + 3] = (unsigned char) ((n)); \
}

```

Definition at line 52 of file sha2.c.

**8.122.1.7 #define R(t)****Value:**

```

(
    W[t] = S1(W[t - 2]) + W[t - 7] + \
    S0(W[t - 15]) + W[t - 16] \
)

```

**8.122.1.8 #define ROTR(x, n) (SHR(x, n) | (x << (32 - n)))****8.122.1.9 #define S0(x) (ROTR(x, 7) ^ ROTR(x, 18) ^ SHR(x, 3))**

Referenced by sha4\_process().

**8.122.1.10 #define S1(x) (ROTR(x, 17) ^ ROTR(x, 19) ^ SHR(x, 10))**

Referenced by sha4\_process(), and tls1\_prf().

**8.122.1.11 #define S2(x) (ROTR(x, 2) ^ ROTR(x, 13) ^ ROTR(x, 22))**

Referenced by tls1\_prf().

**8.122.1.12 #define S3(x) (ROTR(x, 6) ^ ROTR(x, 11) ^ ROTR(x, 25))****8.122.1.13 #define SHR(x, n) ((x & 0xFFFFFFFF) >> n)****8.122.2 Function Documentation****8.122.2.1 void sha2 (unsigned char \* *input*, int *ilen*, unsigned char \* *output*, int *is224*)**

Output = SHA-256( input buffer ).

**Parameters:***input* buffer holding the data*ilen* length of the input data

*output* SHA-224/256 checksum result  
*is224* 0 = use SHA256, 1 = use SHA224

Definition at line 317 of file sha2.c.

References sha2\_finish(), sha2\_starts(), and sha2\_update().

Referenced by main().

#### 8.122.2.2 int sha2\_file (char \* *path*, unsigned char \* *output*, int *is224*)

Output = SHA-256( file contents ).

##### Parameters:

*path* input file name  
*output* SHA-224/256 checksum result  
*is224* 0 = use SHA256, 1 = use SHA224

##### Returns:

0 if successful, 1 if fopen failed, or 2 if fread failed

Definition at line 332 of file sha2.c.

References f, sha2\_finish(), sha2\_starts(), and sha2\_update().

Referenced by sha2\_wrapper().

#### 8.122.2.3 void sha2\_finish (sha2\_context \* *ctx*, unsigned char \* *output*)

SHA-256 final digest.

##### Parameters:

*ctx* SHA-256 context  
*output* SHA-224/256 checksum result

Definition at line 283 of file sha2.c.

References sha2\_context::is224, PUT\_UINT32\_BE, sha2\_padding, sha2\_update(), sha2\_context::state, and sha2\_context::total.

Referenced by main(), sha2(), sha2\_file(), and sha2\_hmac\_finish().

#### 8.122.2.4 void sha2\_hmac (unsigned char \* *key*, int *keylen*, unsigned char \* *input*, int *ilen*, unsigned char \* *output*, int *is224*)

Output = HMAC-SHA-256( hmac key, input buffer ).

##### Parameters:

*key* HMAC secret key  
*keylen* length of the HMAC key

*input* buffer holding the data  
*ilen* length of the input data  
*output* HMAC-SHA-224/256 result  
*is224* 0 = use SHA256, 1 = use SHA224

Definition at line 416 of file sha2.c.

References sha2\_hmac\_finish(), sha2\_hmac\_starts(), and sha2\_hmac\_update().

#### 8.122.2.5 void sha2\_hmac\_finish (sha2\_context \* ctx, unsigned char \* output)

SHA-256 HMAC final digest.

##### Parameters:

*ctx* HMAC context  
*output* SHA-224/256 HMAC checksum result

Definition at line 396 of file sha2.c.

References sha2\_context::is224, sha2\_context::opad, sha2\_finish(), sha2\_starts(), and sha2\_update().

Referenced by main(), and sha2\_hmac().

#### 8.122.2.6 void sha2\_hmac\_starts (sha2\_context \* ctx, int is224, unsigned char \* key, int keylen)

SHA-256 HMAC context setup.

##### Parameters:

*ctx* HMAC context to be initialized  
*is224* 0 = use SHA256, 1 = use SHA224  
*key* HMAC secret key  
*keylen* length of the HMAC key

Definition at line 364 of file sha2.c.

References sha2\_context::ipad, sha2\_context::opad, sha2\_starts(), and sha2\_update().

Referenced by main(), and sha2\_hmac().

#### 8.122.2.7 void sha2\_hmac\_update (sha2\_context \* ctx, unsigned char \* input, int ilen)

SHA-256 HMAC process buffer.

##### Parameters:

*ctx* HMAC context  
*input* buffer holding the data  
*ilen* length of the input data

Definition at line 387 of file sha2.c.

References sha2\_update().

Referenced by main(), and sha2\_hmac().

**8.122.2.8 static void sha2\_process (sha2\_context \* *ctx*, unsigned char *data*[64])** [static]

Definition at line 97 of file sha2.c.

References F, GET\_UINT32\_BE, P, R, and sha2\_context::state.

Referenced by sha2\_update().

**8.122.2.9 int sha2\_self\_test (int *verbose*)**

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 527 of file sha2.c.

Referenced by main().

**8.122.2.10 void sha2\_starts (sha2\_context \* *ctx*, int *is224*)**

SHA-256 context setup.

**Parameters:**

*ctx* context to be initialized

*is224* 0 = use SHA256, 1 = use SHA224

Definition at line 64 of file sha2.c.

References sha2\_context::is224, sha2\_context::state, and sha2\_context::total.

Referenced by main(), sha2(), sha2\_file(), sha2\_hmac\_finish(), and sha2\_hmac\_starts().

**8.122.2.11 void sha2\_update (sha2\_context \* *ctx*, unsigned char \* *input*, int *ilen*)**

SHA-256 process buffer.

**Parameters:**

*ctx* SHA-256 context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 231 of file sha2.c.

References sha2\_context::buffer, sha2\_process(), and sha2\_context::total.

Referenced by main(), sha2(), sha2\_file(), sha2\_finish(), sha2\_hmac\_finish(), sha2\_hmac\_starts(), and sha2\_hmac\_update().

### 8.122.3 Variable Documentation

#### 8.122.3.1 `const char _sha2_src[] = "_sha2_src"` `[static]`

Definition at line 429 of file sha2.c.

#### 8.122.3.2 `const unsigned char sha2_padding[64]` `[static]`

**Initial value:**

```
{  
    0x80, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0  
}
```

Definition at line 272 of file sha2.c.

Referenced by sha2\_finish().



## 8.123 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/sha4.c File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/sha4.h"
```

### Defines

- #define [\\_CRT\\_SECURE\\_NO\\_DEPRECATED](#) 1
- #define [GET\\_UINT64\\_BE](#)(n, b, i)
- #define [PUT\\_UINT64\\_BE](#)(n, b, i)
- #define [SHR](#)(x, n) (x >> n)
- #define [ROTR](#)(x, n) (SHR(x, n) | (x << (64 - n)))
- #define [S0](#)(x) (ROTR(x, 1) ^ ROTR(x, 8) ^ SHR(x, 7))
- #define [S1](#)(x) (ROTR(x, 19) ^ ROTR(x, 61) ^ SHR(x, 6))
- #define [S2](#)(x) (ROTR(x, 28) ^ ROTR(x, 34) ^ ROTR(x, 39))
- #define [S3](#)(x) (ROTR(x, 14) ^ ROTR(x, 18) ^ ROTR(x, 41))
- #define [F0](#)(x, y, z) ((x & y) | (z & (x | y)))
- #define [F1](#)(x, y, z) (z ^ (x & (y ^ z)))
- #define [P](#)(a, b, c, d, e, [f](#), g, h, x, [K](#))

### Functions

- void [sha4\\_starts](#) ([sha4\\_context](#) \*ctx, [int](#) is384)  
*SHA-512 context setup.*
- static void [sha4\\_process](#) ([sha4\\_context](#) \*ctx, unsigned char data[128])
- void [sha4\\_update](#) ([sha4\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*SHA-512 process buffer.*
- void [sha4\\_finish](#) ([sha4\\_context](#) \*ctx, unsigned char \*output)  
*SHA-512 final digest.*
- void [sha4](#) (unsigned char \*input, [int](#) ilen, unsigned char \*output, [int](#) is384)  
*Output = SHA-512( input buffer ).*
- [int](#) [sha4\\_file](#) (char \*path, unsigned char \*output, [int](#) is384)  
*Output = SHA-512( file contents ).*
- void [sha4\\_hmac\\_starts](#) ([sha4\\_context](#) \*ctx, [int](#) is384, unsigned char \*key, [int](#) keylen)  
*SHA-512 HMAC context setup.*
- void [sha4\\_hmac\\_update](#) ([sha4\\_context](#) \*ctx, unsigned char \*input, [int](#) ilen)  
*SHA-512 HMAC process buffer.*
- void [sha4\\_hmac\\_finish](#) ([sha4\\_context](#) \*ctx, unsigned char \*output)  
*SHA-512 HMAC final digest.*

- void `sha4_hmac` (unsigned char \*key, int keylen, unsigned char \*input, int ilen, unsigned char \*output, int is384)  
*Output = HMAC-SHA-512( hmac key, input buffer ).*
- int `sha4_self_test` (int verbose)  
*Checkup routine.*

## Variables

- static const uint64 `K` [80]
- static const unsigned char `sha4_padding` [128]
- static const char `_sha4_src` [] = "\_sha4\_src"

### 8.123.1 Define Documentation

#### 8.123.1.1 #define \_CRT\_SECURE\_NO\_DEPRECATED 1

Definition at line 30 of file sha4.c.

#### 8.123.1.2 #define F0(x, y, z) ((x & y) | (z & (x | y)))

#### 8.123.1.3 #define F1(x, y, z) (z ^ (x & (y ^ z)))

#### 8.123.1.4 #define GET\_UINT64\_BE(n, b, i)

Value:

```
{
    (n) = ( (uint64) (b) [ (i)      ] << 56 )
    | ( (uint64) (b) [ (i) + 1 ] << 48 )
    | ( (uint64) (b) [ (i) + 2 ] << 40 )
    | ( (uint64) (b) [ (i) + 3 ] << 32 )
    | ( (uint64) (b) [ (i) + 4 ] << 24 )
    | ( (uint64) (b) [ (i) + 5 ] << 16 )
    | ( (uint64) (b) [ (i) + 6 ] <<  8 )
    | ( (uint64) (b) [ (i) + 7 ]      );
}
```

Definition at line 42 of file sha4.c.

Referenced by `sha4_process()`.

#### 8.123.1.5 #define P(a, b, c, d, e, f, g, h, x, K)

Value:

```
{
    temp1 = h + S3(e) + F1(e, f, g) + K + x;
    temp2 = S2(a) + F0(a, b, c);
    d += temp1; h = temp1 + temp2;
}
```

**8.123.1.6 #define PUT\_UINT64\_BE(n, b, i)****Value:**

```

{
    (b)[(i)] = (unsigned char) ((n) >> 56); \
    (b)[(i) + 1] = (unsigned char) ((n) >> 48); \
    (b)[(i) + 2] = (unsigned char) ((n) >> 40); \
    (b)[(i) + 3] = (unsigned char) ((n) >> 32); \
    (b)[(i) + 4] = (unsigned char) ((n) >> 24); \
    (b)[(i) + 5] = (unsigned char) ((n) >> 16); \
    (b)[(i) + 6] = (unsigned char) ((n) >> 8); \
    (b)[(i) + 7] = (unsigned char) ((n)); \
}

```

Definition at line 56 of file sha4.c.

Referenced by sha4\_finish().

**8.123.1.7 #define ROTR(x, n) (SHR(x,n) | (x << (64 - n)))****8.123.1.8 #define S0(x) (ROTR(x, 1) ^ ROTR(x, 8) ^ SHR(x, 7))****8.123.1.9 #define S1(x) (ROTR(x,19) ^ ROTR(x,61) ^ SHR(x, 6))****8.123.1.10 #define S2(x) (ROTR(x,28) ^ ROTR(x,34) ^ ROTR(x,39))****8.123.1.11 #define S3(x) (ROTR(x,14) ^ ROTR(x,18) ^ ROTR(x,41))****8.123.1.12 #define SHR(x, n) (x >> n)****8.123.2 Function Documentation****8.123.2.1 void sha4 (unsigned char \* *input*, int *ilen*, unsigned char \* *output*, int *is384*)**

Output = SHA-512( input buffer ).

**Parameters:***input* buffer holding the data*ilen* length of the input data*output* SHA-384/512 checksum result*is384* 0 = use SHA512, 1 = use SHA384

Definition at line 315 of file sha4.c.

References sha4\_finish(), sha4\_starts(), and sha4\_update().

**8.123.2.2 int sha4\_file (char \* *path*, unsigned char \* *output*, int *is384*)**

Output = SHA-512( file contents ).

**Parameters:***path* input file name

*output* SHA-384/512 checksum result  
*is384* 0 = use SHA512, 1 = use SHA384

**Returns:**

0 if successful, 1 if fopen failed, or 2 if fread failed

Definition at line 330 of file sha4.c.

References f, sha4\_finish(), sha4\_starts(), and sha4\_update().

**8.123.2.3 void sha4\_finish (sha4\_context \* ctx, unsigned char \* output)**

SHA-512 final digest.

**Parameters:**

*ctx* SHA-512 context  
*output* SHA-384/512 checksum result

Definition at line 279 of file sha4.c.

References int, sha4\_context::is384, PUT\_UINT64\_BE, sha4\_padding, sha4\_update(), sha4\_context::state, sha4\_context::total, and uint64.

Referenced by sha4(), sha4\_file(), and sha4\_hmac\_finish().

**8.123.2.4 void sha4\_hmac (unsigned char \* key, int keylen, unsigned char \* input, int ilen, unsigned char \* output, int is384)**

Output = HMAC-SHA-512( hmac key, input buffer ).

**Parameters:**

*key* HMAC secret key  
*keylen* length of the HMAC key  
*input* buffer holding the data  
*ilen* length of the input data  
*output* HMAC-SHA-384/512 result  
*is384* 0 = use SHA512, 1 = use SHA384

Definition at line 414 of file sha4.c.

References sha4\_hmac\_finish(), sha4\_hmac\_starts(), and sha4\_hmac\_update().

**8.123.2.5 void sha4\_hmac\_finish (sha4\_context \* ctx, unsigned char \* output)**

SHA-512 HMAC final digest.

**Parameters:**

*ctx* HMAC context

*output* SHA-384/512 HMAC checksum result

Definition at line 394 of file sha4.c.

References sha4\_context::is384, sha4\_context::opad, sha4\_finish(), sha4\_starts(), and sha4\_update().

Referenced by sha4\_hmac().

#### 8.123.2.6 void sha4\_hmac\_starts (sha4\_context \* ctx, int is384, unsigned char \* key, int keylen)

SHA-512 HMAC context setup.

##### Parameters:

*ctx* HMAC context to be initialized

*is384* 0 = use SHA512, 1 = use SHA384

*key* HMAC secret key

*keylen* length of the HMAC key

Definition at line 362 of file sha4.c.

References sha4\_context::ipad, sha4\_context::opad, sha4\_starts(), and sha4\_update().

Referenced by sha4\_hmac().

#### 8.123.2.7 void sha4\_hmac\_update (sha4\_context \* ctx, unsigned char \* input, int ilen)

SHA-512 HMAC process buffer.

##### Parameters:

*ctx* HMAC context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 385 of file sha4.c.

References sha4\_update().

Referenced by sha4\_hmac().

#### 8.123.2.8 static void sha4\_process (sha4\_context \* ctx, unsigned char data[128]) [static]

Definition at line 152 of file sha4.c.

References F, GET\_UINT64\_BE, K, P, S0, S1, sha4\_context::state, and uint64.

Referenced by sha4\_update().

#### 8.123.2.9 int sha4\_self\_test (int verbose)

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 544 of file sha4.c.

**8.123.2.10 void sha4\_starts (sha4\_context \* ctx, int is384)**

SHA-512 context setup.

**Parameters:**

*ctx* context to be initialized

*is384* 0 = use SHA512, 1 = use SHA384

Definition at line 119 of file sha4.c.

References sha4\_context::is384, sha4\_context::state, sha4\_context::total, and UL64.

Referenced by sha4(), sha4\_file(), sha4\_hmac\_finish(), and sha4\_hmac\_starts().

**8.123.2.11 void sha4\_update (sha4\_context \* ctx, unsigned char \* input, int ilen)**

SHA-512 process buffer.

**Parameters:**

*ctx* SHA-512 context

*input* buffer holding the data

*ilen* length of the input data

Definition at line 224 of file sha4.c.

References sha4\_context::buffer, int, sha4\_process(), sha4\_context::total, and uint64.

Referenced by sha4(), sha4\_file(), sha4\_finish(), sha4\_hmac\_finish(), sha4\_hmac\_starts(), and sha4\_hmac\_update().

**8.123.3 Variable Documentation****8.123.3.1 const char \_sha4\_src[] = "\_sha4\_src" [static]**

Definition at line 427 of file sha4.c.

**8.123.3.2 const uint64 K[80] [static]**

Definition at line 72 of file sha4.c.

**8.123.3.3 const unsigned char sha4\_padding[128] [static]****Initial value:**

```
{
  0x80, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
}
```

Definition at line 264 of file sha4.c.

Referenced by sha4\_finish().

## 8.124 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/ssl\_cli.c File Reference

```
#include <string.h>
#include <stdlib.h>
#include <time.h>
#include "xyssl/ssl.h"
```

### Defines

- `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

### Functions

- static `int ssl\_write\_client\_hello (ssl_context *ssl)`
- static `int ssl\_parse\_server\_hello (ssl_context *ssl)`
- static `int ssl\_parse\_server\_key\_exchange (ssl_context *ssl)`
- static `int ssl\_parse\_certificate\_request (ssl_context *ssl)`
- static `int ssl\_parse\_server\_hello\_done (ssl_context *ssl)`
- static `int ssl\_write\_client\_key\_exchange (ssl_context *ssl)`
- static `int ssl\_write\_certificate\_verify (ssl_context *ssl)`
- `int ssl\_client\_start (ssl_context *ssl)`

### Variables

- static const char `\_ssl\_cli\_src [] = "_ssl_cli_src"`

### 8.124.1 Define Documentation

#### 8.124.1.1 `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

Definition at line 25 of file `ssl_cli.c`.

### 8.124.2 Function Documentation

#### 8.124.2.1 `int ssl\_client\_start (ssl_context * ssl)`

Definition at line 441 of file `ssl_cli.c`.

References `SSL_CERTIFICATE_REQUEST`, `SSL_CERTIFICATE_VERIFY`, `SSL_CLIENT_CERTIFICATE`, `SSL_CLIENT_CHANGE_CIPHER_SPEC`, `SSL_CLIENT_FINISHED`, `SSL_CLIENT_HELLO`, `SSL_CLIENT_KEY_EXCHANGE`, `ssl_flush_output()`, `SSL_HELLO_REQUEST`, `ssl_parse_certificate()`, `ssl_parse_certificate_request()`, `ssl_parse_change_cipher_spec()`, `ssl_parse_finished()`, `ssl_parse_server_hello()`, `ssl_parse_server_hello_done()`, `ssl_parse_server_key_exchange()`, `SSL_SERVER_CERTIFICATE`, `SSL_SERVER_CHANGE_CIPHER_SPEC`, `SSL_SERVER_FINISHED`, `SSL_SERVER_HELLO`, `SSL_SERVER_HELLO_DONE`, `SSL_SERVER_KEY_EXCHANGE`, `ssl_write_certificate()`, `ssl_write_certificate_verify()`, `ssl_write_change_cipher_spec()`, `ssl_write_client_hello()`, `ssl_write_client_key_exchange()`, `ssl_write_finished()`, and `ssl_context::state`.



Referenced by `ssl_handshake()`.

#### 8.124.2.2 `static int ssl_parse_certificate_request (ssl_context *ssl)` [static]

Definition at line 267 of file `ssl_cli.c`.

References `ssl_context::client_auth`, `ERR_SSL_CERTIFICATE_REQUIRED`, `ERR_SSL_PRIVATE_KEY_REQUIRED`, `ERR_SSL_UNEXPECTED_MESSAGE`, `ssl_context::in_msg`, `ssl_context::in_msgtype`, `ssl_context::own_cert`, `ssl_context::own_key`, `SSL_HS_CERTIFICATE_REQUEST`, `SSL_MSG_HANDSHAKE`, `ssl_read_record()`, and `ssl_context::state`.

Referenced by `ssl_client_start()`.

#### 8.124.2.3 `static int ssl_parse_server_hello (ssl_context *ssl)` [static]

Definition at line 105 of file `ssl_cli.c`.

References `ssl_context::cipher`, `ssl_context::cipherlist`, `ERR_SSL_BAD_HS_SERVER_HELLO`, `ERR_SSL_NO_CIPHER_CHOSEN`, `ERR_SSL_UNEXPECTED_MESSAGE`, `ssl_context::in_hhlen`, `ssl_context::in_msg`, `ssl_context::in_msgtype`, `int`, `ssl_context::minor_ver`, `ssl_context::randbytes`, `ssl_context::resumed`, `ssl_context::sessid`, `ssl_context::sidlen`, `SSL_COMPRESS_NULL`, `ssl_derive_keys()`, `SSL_HS_SERVER_HELLO`, `SSL_MSG_HANDSHAKE`, `ssl_read_record()`, `SSL_SERVER_CHANGE_CIPHER_SPEC`, `SSLV3_MAJOR_VERSION`, `SSLV3_MINOR_VERSION`, `ssl_context::state`, and `TLS10_MINOR_VERSION`.

Referenced by `ssl_client_start()`.

#### 8.124.2.4 `static int ssl_parse_server_hello_done (ssl_context *ssl)` [static]

Definition at line 310 of file `ssl_cli.c`.

References `ssl_context::client_auth`, `ERR_SSL_BAD_HS_SERVER_HELLO_DONE`, `ERR_SSL_UNEXPECTED_MESSAGE`, `ssl_context::in_hhlen`, `ssl_context::in_msg`, `ssl_context::in_msgtype`, `SSL_HS_SERVER_HELLO_DONE`, `SSL_MSG_HANDSHAKE`, `ssl_read_record()`, and `ssl_context::state`.

Referenced by `ssl_client_start()`.

#### 8.124.2.5 `static int ssl_parse_server_key_exchange (ssl_context *ssl)` [static]

Definition at line 184 of file `ssl_cli.c`.

References `ssl_context::cipher`, `ssl_context::dhm_ctx`, `dhm_read_params()`, `ERR_SSL_BAD_HS_SERVER_KEY_EXCHANGE`, `ERR_SSL_FEATURE_UNAVAILABLE`, `ERR_SSL_UNEXPECTED_MESSAGE`, `ssl_context::in_hhlen`, `ssl_context::in_msg`, `ssl_context::in_msgtype`, `dhm_context::len`, `rsa_context::len`, `md5()`, `md5_finish()`, `md5_starts()`, `md5_update()`, `ssl_context::peer_cert`, `ssl_context::randbytes`, `_x509_cert::rsa`, `rsa_pkcs1_verify()`, `RSA_RAW`, `sha1()`, `sha1_finish()`, `sha1_starts()`, `sha1_update()`, `SSL3_EDH_RSA_DES_168_SHA`, `SSL_HS_SERVER_KEY_EXCHANGE`, `SSL_MSG_HANDSHAKE`, `ssl_read_record()`, `ssl_context::state`, and `TLS1_EDH_RSA_AES_256_SHA`.

Referenced by `ssl_client_start()`.

#### 8.124.2.6 `static int ssl_write_certificate_verify (ssl_context *ssl)` [static]

Definition at line 404 of file `ssl_cli.c`.

References `ssl_context::client_auth`, `rsa_context::len`, `ssl_context::out_msg`, `ssl_context::out_msglen`, `ssl_context::out_msgtype`, `ssl_context::own_key`, `rsa_pkcs1_sign()`, `RSA_RAW`, `ssl_calc_verify()`, `SSL_HS_CERTIFICATE_VERIFY`, `SSL_MSG_HANDSHAKE`, `ssl_write_record()`, and `ssl_context::state`.

Referenced by `ssl_client_start()`.

#### 8.124.2.7 `static int ssl_write_client_hello (ssl_context *ssl) [static]`

Definition at line 34 of file `ssl_cli.c`.

References `ssl_context::cipherlist`, `ssl_context::hs_md5`, `ssl_context::hs_sha1`, `ssl_context::major_ver`, `ssl_context::max_ver`, `md5_starts()`, `ssl_context::minor_ver`, `ssl_context::out_msg`, `ssl_context::out_msglen`, `ssl_context::out_msgtype`, `ssl_context::randbytes`, `ssl_context::rng_d`, `ssl_context::rng_f`, `ssl_context::sessid`, `sha1_starts()`, `ssl_context::sidlen`, `SSL_COMPRESS_NULL`, `SSL_HS_CLIENT_HELLO`, `SSL_MSG_HANDSHAKE`, `ssl_write_record()`, `SSLV3_MAJOR_VERSION`, `SSLV3_MINOR_VERSION`, `ssl_context::state`, and `TLS10_MINOR_VERSION`.

Referenced by `ssl_client_start()`.

#### 8.124.2.8 `static int ssl_write_client_key_exchange (ssl_context *ssl) [static]`

Definition at line 331 of file `ssl_cli.c`.

References `ssl_context::cipher`, `dhm_calc_secret()`, `ssl_context::dhm_ctx`, `dhm_make_public()`, `ERR_SSL_FEATURE_UNAVAILABLE`, `rsa_context::len`, `dhm_context::len`, `ssl_context::max_ver`, `ssl_context::minor_ver`, `ssl_context::out_msg`, `ssl_context::out_msglen`, `ssl_context::out_msgtype`, `ssl_context::peer_cert`, `ssl_context::pmslen`, `ssl_context::premaster`, `ssl_context::rng_d`, `ssl_context::rng_f`, `_x509_cert::rsa`, `rsa_pkcs1_encrypt()`, `SSL3_EDH_RSA_DES_168_SHA`, `ssl_derive_keys()`, `SSL_HS_CLIENT_KEY_EXCHANGE`, `SSL_MSG_HANDSHAKE`, `ssl_write_record()`, `SSLV3_MINOR_VERSION`, `ssl_context::state`, and `TLS1_EDH_RSA_AES_256_SHA`.

Referenced by `ssl_client_start()`.

### 8.124.3 Variable Documentation

#### 8.124.3.1 `const char _ssl_cli_src[] = "_ssl_cli_src" [static]`

Definition at line 436 of file `ssl_cli.c`.

## 8.125 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/ssl\_srv.c File Reference

```
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include "xyssl/net.h"
#include "xyssl/ssl.h"
```

### Defines

- `#define _CRT_SECURE_NO_DEPRECATED 1`

### Functions

- static `int ssl_get_session (ssl_context *ssl)`
- static void `ssl_set_session (ssl_context *ssl)`
- static `int ssl_parse_client_hello (ssl_context *ssl)`
- static `int ssl_write_server_hello (ssl_context *ssl)`
- static `int ssl_write_certificate_request (ssl_context *ssl)`
- static `int ssl_write_server_key_exchange (ssl_context *ssl)`
- static `int ssl_write_server_hello_done (ssl_context *ssl)`
- static `int ssl_parse_client_key_exchange (ssl_context *ssl)`
- static `int ssl_parse_certificate_verify (ssl_context *ssl)`
- `int ssl_server_start (ssl_context *ssl)`

### Variables

- static const char `_ssl_srv_src [] = "_ssl_srv_src"`

## 8.125.1 Define Documentation

### 8.125.1.1 `#define _CRT_SECURE_NO_DEPRECATED 1`

Definition at line 25 of file `ssl_srv.c`.

## 8.125.2 Function Documentation

### 8.125.2.1 `static int ssl_get_session (ssl_context *ssl)` [static]

Definition at line 36 of file `ssl_srv.c`.

References `ssl_context::cipher`, `ERR_SSL_NO_SESSION_FOUND`, `int`, `ssl_context::master`, `ssl_context::sessid`, `sha1()`, `ssl_context::sidlen`, `ssl_context::sidtable`, `SSL_EXPIRATION_TIME`, and `SSL_SESSION_TBL_LEN`.

Referenced by `ssl_write_server_hello()`.

**8.125.2.2 static int ssl\_parse\_certificate\_verify (ssl\_context \* ssl) [static]**

Definition at line 624 of file ssl\_srv.c.

References ssl\_context::authmode, ERR\_SSL\_BAD\_HS\_CERTIFICATE\_VERIFY, ssl\_context::in\_hrlen, ssl\_context::in\_msg, ssl\_context::in\_msgtype, int, rsa\_context::len, ssl\_context::peer\_cert, \_x509\_cert::rsa, rsa\_pkcs1\_verify(), RSA\_RAW, ssl\_calc\_verify(), SSL\_HS\_CERTIFICATE\_VERIFY, SSL\_MSG\_HANDSHAKE, ssl\_read\_record(), SSL\_VERIFY\_REQUIRED, and ssl\_context::state.

Referenced by ssl\_server\_start().

**8.125.2.3 static int ssl\_parse\_client\_hello (ssl\_context \* ssl) [static]**

Definition at line 119 of file ssl\_srv.c.

References ssl\_context::cipher, ssl\_context::cipherlist, ERR\_SSL\_BAD\_HS\_CLIENT\_HELLO, ERR\_SSL\_NO\_CIPHER\_CHOSEN, ssl\_context::hs\_md5, ssl\_context::hs\_sha1, ssl\_context::in\_hdr, ssl\_context::in\_left, ssl\_context::in\_msg, int, ssl\_context::major\_ver, ssl\_context::max\_ver, md5\_starts(), md5\_update(), ssl\_context::minor\_ver, net\_recv(), ssl\_context::randbytes, ssl\_context::read\_fd, ssl\_context::sessid, sha1\_starts(), sha1\_update(), ssl\_context::sidlen, SSL\_HS\_CLIENT\_HELLO, SSL\_MSG\_HANDSHAKE, SSLV3\_MAJOR\_VERSION, ssl\_context::state, and TLS10\_MINOR\_VERSION.

Referenced by ssl\_server\_start().

**8.125.2.4 static int ssl\_parse\_client\_key\_exchange (ssl\_context \* ssl) [static]**

Definition at line 543 of file ssl\_srv.c.

References ssl\_context::cipher, dhm\_calc\_secret(), ssl\_context::dhm\_ctx, dhm\_read\_public(), ERR\_SSL\_BAD\_HS\_CLIENT\_KEY\_EXCHANGE, ERR\_SSL\_FEATURE\_UNAVAILABLE, ssl\_context::in\_hrlen, ssl\_context::in\_msg, ssl\_context::in\_msgtype, rsa\_context::len, dhm\_context::len, ssl\_context::max\_ver, ssl\_context::minor\_ver, ssl\_context::own\_key, ssl\_context::pmslen, ssl\_context::premaster, rsa\_pkcs1\_decrypt(), SSL3\_EDH\_RSA\_DES\_168\_SHA, ssl\_derive\_keys(), SSL\_HS\_CLIENT\_KEY\_EXCHANGE, SSL\_MSG\_HANDSHAKE, ssl\_read\_record(), ssl\_set\_session(), SSLV3\_MINOR\_VERSION, ssl\_context::state, and TLS1\_EDH\_RSA\_AES\_256\_SHA.

Referenced by ssl\_server\_start().

**8.125.2.5 int ssl\_server\_start (ssl\_context \* ssl)**

Definition at line 666 of file ssl\_srv.c.

References SSL\_CERTIFICATE\_REQUEST, SSL\_CERTIFICATE\_VERIFY, SSL\_CLIENT\_CERTIFICATE, SSL\_CLIENT\_CHANGE\_CIPHER\_SPEC, SSL\_CLIENT\_FINISHED, SSL\_CLIENT\_HELLO, SSL\_CLIENT\_KEY\_EXCHANGE, ssl\_flush\_output(), SSL\_HELLO\_REQUEST, ssl\_parse\_certificate(), ssl\_parse\_certificate\_verify(), ssl\_parse\_change\_cipher\_spec(), ssl\_parse\_client\_hello(), ssl\_parse\_client\_key\_exchange(), ssl\_parse\_finished(), SSL\_SERVER\_CERTIFICATE, SSL\_SERVER\_CHANGE\_CIPHER\_SPEC, SSL\_SERVER\_FINISHED, SSL\_SERVER\_HELLO, SSL\_SERVER\_HELLO\_DONE, SSL\_SERVER\_KEY\_EXCHANGE, ssl\_write\_certificate(), ssl\_write\_certificate\_request(), ssl\_write\_change\_cipher\_spec(), ssl\_write\_finished(), ssl\_write\_server\_hello(), ssl\_write\_server\_hello\_done(), ssl\_write\_server\_key\_exchange(), and ssl\_context::state.

Referenced by main(), and ssl\_handshake().

**8.125.2.6 static void ssl\_set\_session (ssl\_context \* ssl) [static]**

Definition at line 81 of file ssl\_srv.c.

References ssl\_context::cipher, int, ssl\_context::master, ssl\_context::sessid, sha1(), ssl\_context::sidlen, ssl\_context::sidtable, and SSL\_SESSION\_TBL\_LEN.

Referenced by ssl\_parse\_client\_key\_exchange().

**8.125.2.7 static int ssl\_write\_certificate\_request (ssl\_context \* ssl) [static]**

Definition at line 409 of file ssl\_srv.c.

References ssl\_context::authmode, ssl\_context::ca\_chain, \_x509\_buf::len, \_x509\_cert::next, ssl\_context::out\_msg, ssl\_context::out\_msglen, ssl\_context::out\_msgtype, \_x509\_buf::p, SSL\_HS\_CERTIFICATE\_REQUEST, SSL\_MSG\_HANDSHAKE, SSL\_VERIFY\_NONE, ssl\_write\_record(), ssl\_context::state, and \_x509\_cert::subject\_raw.

Referenced by ssl\_server\_start().

**8.125.2.8 static int ssl\_write\_server\_hello (ssl\_context \* ssl) [static]**

Definition at line 339 of file ssl\_srv.c.

References ssl\_context::cipher, ssl\_context::major\_ver, ssl\_context::minor\_ver, ssl\_context::out\_msg, ssl\_context::out\_msglen, ssl\_context::out\_msgtype, ssl\_context::randbytes, ssl\_context::resumed, ssl\_context::rng\_d, ssl\_context::rng\_f, ssl\_context::sessid, ssl\_context::sidlen, SSL\_COMPRESS\_NULL, ssl\_derive\_keys(), ssl\_get\_session(), SSL\_HS\_SERVER\_HELLO, SSL\_MSG\_HANDSHAKE, SSL\_SERVER\_CHANGE\_CIPHER\_SPEC, ssl\_write\_record(), and ssl\_context::state.

Referenced by ssl\_server\_start().

**8.125.2.9 static int ssl\_write\_server\_hello\_done (ssl\_context \* ssl) [static]**

Definition at line 533 of file ssl\_srv.c.

References ssl\_context::out\_msg, ssl\_context::out\_msglen, ssl\_context::out\_msgtype, SSL\_HS\_SERVER\_HELLO\_DONE, SSL\_MSG\_HANDSHAKE, ssl\_write\_record(), and ssl\_context::state.

Referenced by ssl\_server\_start().

**8.125.2.10 static int ssl\_write\_server\_key\_exchange (ssl\_context \* ssl) [static]**

Definition at line 463 of file ssl\_srv.c.

References ssl\_context::cipher, ssl\_context::dhm\_ctx, dhm\_make\_params(), ERR\_SSL\_FEATURE\_UNAVAILABLE, rsa\_context::len, md5(), md5\_finish(), md5\_starts(), md5\_update(), ssl\_context::out\_msg, ssl\_context::out\_msglen, ssl\_context::out\_msgtype, ssl\_context::own\_key, ssl\_context::randbytes, ssl\_context::rng\_d, ssl\_context::rng\_f, rsa\_pkcs1\_sign(), RSA\_RAW, sha1(), sha1\_finish(), sha1\_starts(), sha1\_update(), SSL3\_EDH\_RSA\_DES\_168\_SHA, SSL\_HS\_SERVER\_KEY\_EXCHANGE, SSL\_MSG\_HANDSHAKE, ssl\_write\_record(), ssl\_context::state, and TLS1\_EDH\_RSA\_AES\_256\_SHA.

Referenced by ssl\_server\_start().

### 8.125.3 Variable Documentation

#### 8.125.3.1 `const char _ssl_srv_src[] = "_ssl_srv_src"` `[static]`

Definition at line 661 of file `ssl_srv.c`.

## 8.126 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/ssl\_tls.c File Reference

```
#include <string.h>
#include <stdlib.h>
#include <time.h>
#include "xyssl/net.h"
#include "xyssl/ssl.h"
#include "xyssl/aes.h"
#include "xyssl/arc4.h"
#include "xyssl/des.h"
```

### Defines

- `#define _CRT_SECURE_NO_DEPRECATED 1`

### Functions

- static void `tls1_prf` (unsigned char \*secret, int slen, char \*label, unsigned char \*random, int rlen, unsigned char \*dstbuf, int dlen)
- int `ssl_derive_keys` (ssl\_context \*ssl)
- int `ssl_calc_verify` (ssl\_context \*ssl, unsigned char hash[36])
- static void `ssl_mac_md5` (unsigned char \*secret, unsigned char \*buf, int len, unsigned char \*ctr, int type)
- static void `ssl_mac_sha1` (unsigned char \*secret, unsigned char \*buf, int len, unsigned char \*ctr, int type)
- static int `ssl_encrypt_buf` (ssl\_context \*ssl)
- static int `ssl_decrypt_buf` (ssl\_context \*ssl)
- int `ssl_write_record` (ssl\_context \*ssl, int do\_crypt)
- int `ssl_read_record` (ssl\_context \*ssl, int do\_crypt)
- int `ssl_flush_output` (ssl\_context \*ssl)
- int `ssl_write_certificate` (ssl\_context \*ssl)
- int `ssl_parse_certificate` (ssl\_context \*ssl)
- int `ssl_write_change_cipher_spec` (ssl\_context \*ssl)
- int `ssl_parse_change_cipher_spec` (ssl\_context \*ssl)
- static void `ssl_calc_finished` (ssl\_context \*ssl, unsigned char \*buf, int from, md5\_context \*md5, sha1\_context \*sha1)
- int `ssl_write_finished` (ssl\_context \*ssl)
- int `ssl_parse_finished` (ssl\_context \*ssl)
- int `ssl_init` (ssl\_context \*ssl, int client\_resume)
 

*Initialize the SSL context. If client\_resume is not null, the session id and premaster secret are preserved (client-side only).*
- void `ssl_set_endpoint` (ssl\_context \*ssl, int endpoint)
 

*Set the current endpoint type, SSL\_IS\_CLIENT or SSL\_IS\_SERVER.*

- void `ssl_set_authmode` (`ssl_context` \*ssl, int authmode)  
*Set the certificate verification mode.*
- void `ssl_set_rng_func` (`ssl_context` \*ssl, int(\*rng\_f)(void \*), void \*rng\_d)  
*Set the random number generator function.*
- void `ssl_set_io_files` (`ssl_context` \*ssl, int read\_fd, int write\_fd)  
*Set the read and write file descriptors.*
- void `ssl_set_ciphlist` (`ssl_context` \*ssl, int \*ciphers)  
*Set the list of allowed ciphersuites.*
- void `ssl_set_ca_chain` (`ssl_context` \*ssl, x509\_cert \*ca, char \*cn)  
*Set the CA certificate chain used to verify peer cert, and the peer's expected CommonName (or NULL).*
- void `ssl_set_rsa_cert` (`ssl_context` \*ssl, x509\_cert \*own\_cert, rsa\_context \*own\_key)  
*Set own certificate and private RSA key.*
- void `ssl_set_sidtable` (`ssl_context` \*ssl, unsigned char \*sidtable)  
*Set the global session ID table (server-side only).*
- int `ssl_set_dhm_vals` (`ssl_context` \*ssl, char \*dhm\_P, char \*dhm\_G)  
*Set the Diffie-Hellman public P and G values, provided as hexadecimal strings (server-side only).*
- int `ssl_get_verify_result` (`ssl_context` \*ssl)  
*Return the result of the certificate verification.*
- char \* `ssl_get_cipher_name` (`ssl_context` \*ssl)  
*Return the name of the current cipher.*
- int `ssl_handshake` (`ssl_context` \*ssl)  
*Perform the SSL handshake.*
- int `ssl_read` (`ssl_context` \*ssl, unsigned char \*buf, int \*len)  
*Read at most 'len' application data bytes.*
- int `ssl_write` (`ssl_context` \*ssl, unsigned char \*buf, int len)  
*Write 'len' application data bytes.*
- int `ssl_close_notify` (`ssl_context` \*ssl)  
*Notify the peer that the connection is being closed.*
- void `ssl_free` (`ssl_context` \*ssl)  
*Free an SSL context.*

## Variables

- int `ssl_default_ciphers` []
- static const char `_ssl_tls_src` [] = "\_ssl\_tls\_src"



## 8.126.1 Define Documentation

### 8.126.1.1 #define \_CRT\_SECURE\_NO\_DEPRECATED 1

Definition at line 33 of file ssl\_tls.c.

## 8.126.2 Function Documentation

### 8.126.2.1 static void ssl\_calc\_finished (ssl\_context \*ssl, unsigned char \*buf, int from, md5\_context \*md5, sha1\_context \*sha1) [static]

Definition at line 1048 of file ssl\_tls.c.

### 8.126.2.2 int ssl\_calc\_verify (ssl\_context \*ssl, unsigned char hash[36])

Definition at line 329 of file ssl\_tls.c.

References ssl\_context::hs\_md5, ssl\_context::hs\_sha1, ssl\_context::master, md5(), md5\_finish(), md5\_starts(), md5\_update(), ssl\_context::minor\_ver, sha1(), sha1\_finish(), sha1\_starts(), sha1\_update(), and SSLV3\_MINOR\_VERSION.

Referenced by ssl\_parse\_certificate\_verify(), and ssl\_write\_certificate\_verify().

### 8.126.2.3 int ssl\_close\_notify (ssl\_context \*ssl)

Notify the peer that the connection is being closed.

Definition at line 1453 of file ssl\_tls.c.

References ssl\_context::out\_msg, ssl\_context::out\_msglen, ssl\_context::out\_msgtype, SSL\_ALERT\_CLOSE\_NOTIFY, SSL\_ALERT\_WARNING, ssl\_flush\_output(), SSL\_HANDSHAKE\_OVER, SSL\_MSG\_ALERT, ssl\_write\_record(), and ssl\_context::state.

Referenced by main().

### 8.126.2.4 static int ssl\_decrypt\_buf (ssl\_context \*ssl) [static]

Definition at line 519 of file ssl\_tls.c.

References aes\_cbc\_decrypt(), arc4\_crypt(), ssl\_context::ctx\_dec, des3\_cbc\_decrypt(), ERR\_SSL\_FEATURE\_UNAVAILABLE, ERR\_SSL\_INVALID\_MAC, ssl\_context::in\_ctr, ssl\_context::in\_hdr, ssl\_context::in\_msg, ssl\_context::in\_msglen, ssl\_context::in\_msgtype, int, ssl\_context::iv\_dec, ssl\_context::ivlen, ssl\_context::mac\_dec, ssl\_context::maclen, md5\_hmac(), ssl\_context::minlen, ssl\_context::minor\_ver, ssl\_context::nb\_zero, sha1\_hmac(), ssl\_mac\_md5(), ssl\_mac\_sha1(), and SSLV3\_MINOR\_VERSION.

### 8.126.2.5 int ssl\_derive\_keys (ssl\_context \*ssl)

Definition at line 104 of file ssl\_tls.c.

References aes\_set\_key(), arc4\_setup(), ssl\_context::cipher, ssl\_context::ctx\_dec, ssl\_context::ctx\_enc, ssl\_context::ctxlen, des3\_set\_3keys(), ssl\_context::endpoint, ERR\_SSL\_FEATURE\_UNAVAILABLE, ssl\_context::iv\_dec, ssl\_context::iv\_enc, ssl\_context::ivlen, ssl\_context::keylen, ssl\_context::mac\_dec,

ssl\_context::mac\_enc, ssl\_context::maclen, ssl\_context::master, md5(), md5\_finish(), md5\_starts(), md5\_update(), ssl\_context::minlen, ssl\_context::minor\_ver, ssl\_context::pmslen, ssl\_context::premaster, ssl\_context::randbytes, ssl\_context::resumed, sha1(), sha1\_finish(), sha1\_starts(), sha1\_update(), SSL3\_EDH\_RSA\_DES\_168\_SHA, SSL3\_RSA\_DES\_168\_SHA, SSL3\_RSA\_RC4\_128\_MD5, SSL3\_RSA\_RC4\_128\_SHA, SSL\_IS\_CLIENT, SSLV3\_MINOR\_VERSION, TLS1\_EDH\_RSA\_AES\_256\_SHA, tls1\_prf(), and TLS1\_RSA\_AES\_256\_SHA.

Referenced by ssl\_parse\_client\_key\_exchange(), ssl\_parse\_server\_hello(), ssl\_write\_client\_key\_exchange(), and ssl\_write\_server\_hello().

#### 8.126.2.6 static int ssl\_encrypt\_buf (ssl\_context \*ssl) [static]

Definition at line 436 of file ssl\_tls.c.

References aes\_cbc\_encrypt(), arc4\_crypt(), ssl\_context::ctx\_enc, des3\_cbc\_encrypt(), ERR\_SSL\_FEATURE\_UNAVAILABLE, ssl\_context::iv\_enc, ssl\_context::ivlen, ssl\_context::mac\_enc, ssl\_context::maclen, md5\_hmac(), ssl\_context::minor\_ver, ssl\_context::out\_ctr, ssl\_context::out\_msg, ssl\_context::out\_msglen, ssl\_context::out\_msgtype, sha1\_hmac(), ssl\_mac\_md5(), ssl\_mac\_sha1(), and SSLV3\_MINOR\_VERSION.

Referenced by ssl\_write\_record().

#### 8.126.2.7 int ssl\_flush\_output (ssl\_context \*ssl)

Definition at line 826 of file ssl\_tls.c.

References net\_send(), ssl\_context::out\_hdr, ssl\_context::out\_left, ssl\_context::out\_msglen, and ssl\_context::write\_fd.

Referenced by ssl\_client\_start(), ssl\_close\_notify(), and ssl\_server\_start().

#### 8.126.2.8 void ssl\_free (ssl\_context \*ssl)

Free an SSL context.

Definition at line 1476 of file ssl\_tls.c.

References ssl\_context::ctx\_dec, ssl\_context::ctx\_enc, ssl\_context::ctxlen, ssl\_context::dhm\_ctx, dhm\_free(), ssl\_context::in\_ctr, ssl\_context::out\_ctr, ssl\_context::peer\_cert, SSL\_BUFFER\_LEN, and x509\_free\_cert().

Referenced by main().

#### 8.126.2.9 char\* ssl\_get\_cipher\_name (ssl\_context \*ssl)

Return the name of the current cipher.

Definition at line 1311 of file ssl\_tls.c.

Referenced by main().

#### 8.126.2.10 int ssl\_get\_verify\_result (ssl\_context \*ssl)

Return the result of the certificate verification.

Definition at line 1306 of file ssl\_tls.c.

References `ssl_context::verify_result`.

Referenced by `main()`.

#### 8.126.2.11 `int ssl_handshake(ssl_context * ssl)`

Perform the SSL handshake.

##### Returns:

0 if successful, `ERR_NET_WOULD_BLOCK` (only when the socket is set to non-blocking), or a specific SSL error code.

Definition at line 1372 of file `ssl_tls.c`.

References `ssl_context::endpoint`, `ERR_SSL_FEATURE_UNAVAILABLE`, `ssl_client_start()`, `SSL_IS_CLIENT`, `SSL_IS_SERVER`, and `ssl_server_start()`.

Referenced by `main()`, `ssl_read()`, and `ssl_write()`.

#### 8.126.2.12 `int ssl_init(ssl_context * ssl, int client_resume)`

Initialize the SSL context. If `client_resume` is not null, the session id and premaster secret are preserved (client-side only).

##### Returns:

0 if successful, or 1 if memory allocation failed

Definition at line 1202 of file `ssl_tls.c`.

References `ssl_context::in_ctr`, `ssl_context::in_hdr`, `ssl_context::in_msg`, `ssl_context::master`, `ssl_context::out_ctr`, `ssl_context::out_hdr`, `ssl_context::out_msg`, `ssl_context::sessid`, `ssl_context::sidlen`, and `SSL_BUFFER_LEN`.

Referenced by `main()`.

#### 8.126.2.13 `static void ssl_mac_md5(unsigned char * secret, unsigned char * buf, int len, unsigned char * ctr, int type)` [static]

Definition at line 375 of file `ssl_tls.c`.

References `md5()`, `md5_finish()`, `md5_starts()`, and `md5_update()`.

Referenced by `ssl_decrypt_buf()`, and `ssl_encrypt_buf()`.

#### 8.126.2.14 `static void ssl_mac_sha1(unsigned char * secret, unsigned char * buf, int len, unsigned char * ctr, int type)` [static]

Definition at line 404 of file `ssl_tls.c`.

References `sha1()`, `sha1_finish()`, `sha1_starts()`, and `sha1_update()`.

Referenced by `ssl_decrypt_buf()`, and `ssl_encrypt_buf()`.

**8.126.2.15 int ssl\_parse\_certificate (ssl\_context \* ssl)**

Definition at line 917 of file ssl\_tls.c.

Referenced by ssl\_client\_start(), and ssl\_server\_start().

**8.126.2.16 int ssl\_parse\_change\_cipher\_spec (ssl\_context \* ssl)**

Definition at line 1031 of file ssl\_tls.c.

Referenced by ssl\_client\_start(), and ssl\_server\_start().

**8.126.2.17 int ssl\_parse\_finished (ssl\_context \* ssl)**

Definition at line 1157 of file ssl\_tls.c.

Referenced by ssl\_client\_start(), and ssl\_server\_start().

**8.126.2.18 int ssl\_read (ssl\_context \* ssl, unsigned char \* buf, int \* len)**

Read at most 'len' application data bytes.

**Returns:**

0 if successful, ERR\_NET\_WOULD\_BLOCK (only when the socket is set to non-blocking), or a specific SSL error code.

**Note:**

len is updated to reflect the actual number of data bytes read.

Definition at line 1390 of file ssl\_tls.c.

References ERR\_SSL\_UNEXPECTED\_MESSAGE, ssl\_context::in\_msg, ssl\_context::in\_msglen, ssl\_context::in\_msgtype, ssl\_context::in\_offt, ssl\_handshake(), SSL\_MSG\_APPLICATION\_DATA, and ssl\_read\_record().

Referenced by main().

**8.126.2.19 int ssl\_read\_record (ssl\_context \* ssl, int do\_crypt)**

Definition at line 685 of file ssl\_tls.c.

Referenced by ssl\_parse\_certificate\_request(), ssl\_parse\_certificate\_verify(), ssl\_parse\_client\_key\_exchange(), ssl\_parse\_server\_hello(), ssl\_parse\_server\_hello\_done(), ssl\_parse\_server\_key\_exchange(), and ssl\_read().

**8.126.2.20 void ssl\_set\_authmode (ssl\_context \* ssl, int authmode)**

Set the certificate verification mode.

**Parameters:**

*mode* can be:

SSL\_VERIFY\_NONE: peer certificate is not checked (default).

SSL\_VERIFY\_OPTIONAL: peer certificate is checked, however the handshake continues even if verification failed; you may want to check `ssl->verify_result` after.

SSL\_VERIFY\_REQUIRED: peer *must* present a valid certificate, handshake is aborted if verification failed.

Definition at line 1252 of file `ssl_tls.c`.

Referenced by `main()`.

#### 8.126.2.21 void ssl\_set\_ca\_chain (ssl\_context \* ssl, x509\_cert \* ca, char \* cn)

Set the CA certificate chain used to verify peer cert, and the peer's expected CommonName (or NULL).

Definition at line 1276 of file `ssl_tls.c`.

Referenced by `main()`.

#### 8.126.2.22 void ssl\_set\_ciphlist (ssl\_context \* ssl, int \* ciphers)

Set the list of allowed ciphersuites.

Definition at line 1271 of file `ssl_tls.c`.

Referenced by `main()`.

#### 8.126.2.23 int ssl\_set\_dhm\_vals (ssl\_context \* ssl, char \* dhm\_P, char \* dhm\_G)

Set the Diffie-Hellman public P and G values, provided as hexadecimal strings (server-side only).

##### Returns:

0 if successful, or 1 if the values could not be read

Definition at line 1294 of file `ssl_tls.c`.

Referenced by `main()`.

#### 8.126.2.24 void ssl\_set\_endpoint (ssl\_context \* ssl, int endpoint)

Set the current endpoint type, `SSL_IS_CLIENT` or `SSL_IS_SERVER`.

Definition at line 1247 of file `ssl_tls.c`.

References `ssl_context::endpoint`.

Referenced by `main()`.

#### 8.126.2.25 void ssl\_set\_io\_files (ssl\_context \* ssl, int read\_fd, int write\_fd)

Set the read and write file descriptors.

Definition at line 1265 of file `ssl_tls.c`.

Referenced by `main()`.

**8.126.2.26 void ssl\_set\_rng\_func (ssl\_context \* ssl, int(\*) (void \*) rng\_f, void \* rng\_d)**

Set the random number generator function.

Definition at line 1257 of file ssl\_tls.c.

Referenced by main().

**8.126.2.27 void ssl\_set\_rsa\_cert (ssl\_context \* ssl, x509\_cert \* own\_cert, rsa\_context \* own\_key)**

Set own certificate and private RSA key.

Definition at line 1282 of file ssl\_tls.c.

Referenced by main().

**8.126.2.28 void ssl\_set\_sidtable (ssl\_context \* ssl, unsigned char \* sidtable)**

Set the global session ID table (server-side only).

Definition at line 1289 of file ssl\_tls.c.

Referenced by main().

**8.126.2.29 int ssl\_write (ssl\_context \* ssl, unsigned char \* buf, int len)**

Write 'len' application data bytes.

**Returns:**

0 if successful, ERR\_NET\_WOULD\_BLOCK (only when the socket is set to non-blocking), or a specific SSL error code.

**Note:**

When the socket is set to non-blocking and this function returns ERR\_NET\_WOULD\_BLOCK, it should be called again with the *\*same\** arguments until it returns 0.

Definition at line 1425 of file ssl\_tls.c.

References ssl\_context::out\_msg, ssl\_context::out\_msglen, ssl\_context::out\_msgtype, ssl\_context::out\_uoff, ssl\_handshake(), SSL\_MAX\_CONTENT\_LEN, SSL\_MSG\_APPLICATION\_DATA, and ssl\_write\_record().

Referenced by main().

**8.126.2.30 int ssl\_write\_certificate (ssl\_context \* ssl)**

Definition at line 846 of file ssl\_tls.c.

References ssl\_context::client\_auth, ssl\_context::endpoint, ERR\_SSL\_CERTIFICATE\_REQUIRED, ERR\_SSL\_CERTIFICATE\_TOO\_LARGE, \_x509\_buf::len, ssl\_context::minor\_ver, \_x509\_cert::next, ssl\_context::out\_msg, ssl\_context::out\_msglen, ssl\_context::out\_msgtype, ssl\_context::own\_cert, \_x509\_buf::p, \_x509\_cert::raw, SSL\_ALERT\_NO\_CERTIFICATE, SSL\_ALERT\_WARNING, SSL\_HS\_CERTIFICATE, SSL\_IS\_CLIENT, SSL\_MAX\_CONTENT\_LEN, SSL\_MSG\_ALERT, SSL\_MSG\_HANDSHAKE, ssl\_write\_record(), SSLV3\_MINOR\_VERSION, and ssl\_context::state.

Referenced by ssl\_client\_start(), and ssl\_server\_start().

**8.126.2.31 int ssl\_write\_change\_cipher\_spec (ssl\_context \* ssl)**

Definition at line 1022 of file ssl\_tls.c.

Referenced by ssl\_client\_start(), and ssl\_server\_start().

**8.126.2.32 int ssl\_write\_finished (ssl\_context \* ssl)**

Definition at line 1120 of file ssl\_tls.c.

Referenced by ssl\_client\_start(), and ssl\_server\_start().

**8.126.2.33 int ssl\_write\_record (ssl\_context \* ssl, int do\_crypt)**

Definition at line 648 of file ssl\_tls.c.

References ssl\_context::hs\_md5, ssl\_context::hs\_sha1, ssl\_context::major\_ver, md5\_update(), ssl\_context::minor\_ver, net\_send(), ssl\_context::out\_hdr, ssl\_context::out\_left, ssl\_context::out\_msg, ssl\_context::out\_msglen, ssl\_context::out\_msgtype, sha1\_update(), ssl\_encrypt\_buf(), SSL\_MSG\_HANDSHAKE, and ssl\_context::write\_fd.

Referenced by ssl\_close\_notify(), ssl\_write(), ssl\_write\_certificate(), ssl\_write\_certificate\_request(), ssl\_write\_certificate\_verify(), ssl\_write\_client\_hello(), ssl\_write\_client\_key\_exchange(), ssl\_write\_server\_hello(), ssl\_write\_server\_hello\_done(), and ssl\_write\_server\_key\_exchange().

**8.126.2.34 static void tls1\_prf (unsigned char \* secret, int slen, char \* label, unsigned char \* random, int rlen, unsigned char \* dstbuf, int dlen) [static]**

Definition at line 49 of file ssl\_tls.c.

References md5\_hmac(), S1, S2, and sha1\_hmac().

Referenced by ssl\_derive\_keys().

**8.126.3 Variable Documentation****8.126.3.1 const char \_ssl\_tls\_src[] = "\_ssl\_tls\_src" [static]**

Definition at line 1471 of file ssl\_tls.c.

**8.126.3.2 int ssl\_default\_ciphers[]**

**Initial value:**

```
{
    TLS1_EDH_RSA_AES_256_SHA,

    SSL3_EDH_RSA_DES_168_SHA,

    TLS1_RSA_AES_256_SHA,
```

```
    SSL3_RSA_DES_168_SHA,  
  
    SSL3_RSA_RC4_128_SHA,  
    SSL3_RSA_RC4_128_MD5,  
  
    0  
}
```

Definition at line 1346 of file ssl\_tls.c.

Referenced by main().



## 8.127 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/timing.c File Reference

```
#include "xyssl/timing.h"
#include <unistd.h>
#include <signal.h>
#include <sys/time.h>
#include <time.h>
```

### Data Structures

- struct [\\_hr\\_time](#)

### Defines

- [#define \\_CRT\\_SECURE\\_NO\\_DEPRECATED 1](#)

### Functions

- unsigned long [hardclock](#) (void)  
*Return the CPU cycle counter value.*
- unsigned long [set\\_timer](#) (struct [hr\\_time](#) \*val, int reset)  
*Return the elapsed time in milliseconds.*
- static void [sighandler](#) (int signum)
- void [set\\_alarm](#) (int seconds)  
*Setup an alarm clock.*

### Variables

- static int [hardclock\\_init](#) = 0
- static struct timeval [tv\\_init](#)
- static const char [\\_timing\\_src](#) [] = "\_timing\_src"
- int [alarmed](#) = 0

#### 8.127.1 Define Documentation

##### 8.127.1.1 [#define \\_CRT\\_SECURE\\_NO\\_DEPRECATED 1](#)

Definition at line 25 of file timing.c.

## 8.127.2 Function Documentation

### 8.127.2.1 unsigned long hardclock (void)

Return the CPU cycle counter value.

Definition at line 138 of file timing.c.

References tv\_init.

Referenced by main().

### 8.127.2.2 void set\_alarm (int *seconds*)

Setup an alarm clock.

#### Parameters:

*seconds* delay before the "alarmed" flag is set

Definition at line 230 of file timing.c.

References sighandler().

Referenced by main().

### 8.127.2.3 unsigned long set\_timer (struct hr\_time \* *val*, int *reset*)

Return the elapsed time in milliseconds.

#### Parameters:

*val* points to a timer structure

*reset* if set to 1, the timer is restarted

Definition at line 204 of file timing.c.

References \_hr\_time::start.

### 8.127.2.4 static void sighandler (int *signum*) [static]

Definition at line 224 of file timing.c.

Referenced by set\_alarm().

## 8.127.3 Variable Documentation

### 8.127.3.1 const char \_timing\_src[ ] = "\_timing\_src" [static]

Definition at line 161 of file timing.c.

### 8.127.3.2 int alarmed = 0

Definition at line 163 of file timing.c.

Referenced by main().

**8.127.3.3** `int hardclock_init = 0` `[static]`

Definition at line 135 of file timing.c.

**8.127.3.4** `struct timeval tv_init` `[static]`

Definition at line 136 of file timing.c.

Referenced by `hardclock()`.

## 8.128 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/x509read.c File Reference

```
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include "xyssl/x509.h"
#include "xyssl/base64.h"
#include "xyssl/des.h"
#include "xyssl/sha1.h"
#include "xyssl/md5.h"
#include "xyssl/md4.h"
#include "xyssl/md2.h"
```

### Defines

- `#define _CRT_SECURE_NO_DEPRECATED 1`

### Functions

- static `int asn1_get_len` (unsigned char \*\*p, unsigned char \*end, `int` \*len)
- static `int asn1_get_tag` (unsigned char \*\*p, unsigned char \*end, `int` \*len, `int` tag)
- static `int asn1_get_bool` (unsigned char \*\*p, unsigned char \*end, `int` \*val)
- static `int asn1_get_int` (unsigned char \*\*p, unsigned char \*end, `int` \*val)
- static `int asn1_get_mpi` (unsigned char \*\*p, unsigned char \*end, `mpi` \*X)
- static `int x509_get_version` (unsigned char \*\*p, unsigned char \*end, `int` \*ver)
- static `int x509_get_serial` (unsigned char \*\*p, unsigned char \*end, `x509_buf` \*serial)
- static `int x509_get_alg` (unsigned char \*\*p, unsigned char \*end, `x509_buf` \*alg)
- static `int x509_get_name` (unsigned char \*\*p, unsigned char \*end, `x509_name` \*cur)
- static `int x509_get_dates` (unsigned char \*\*p, unsigned char \*end, `x509_time` \*from, `x509_time` \*to)
- static `int x509_get_pubkey` (unsigned char \*\*p, unsigned char \*end, `x509_buf` \*pk\_alg\_oid, `mpi` \*N, `mpi` \*E)
- static `int x509_get_sig` (unsigned char \*\*p, unsigned char \*end, `x509_buf` \*sig)
- static `int x509_get_uid` (unsigned char \*\*p, unsigned char \*end, `x509_buf` \*uid, `int` n)
- static `int x509_get_ext` (unsigned char \*\*p, unsigned char \*end, `x509_buf` \*ext, `int` \*ca\_istrue, `int` \*max\_pathlen)
- `int x509_add_certs` (`x509_cert` \*chain, unsigned char \*buf, `int` buflen)
 

*Parse one or more certificates and add them to the chain.*
- `int x509_read_crtfile` (`x509_cert` \*chain, char \*path)
 

*Load one or more certificates and add them to the chain.*
- static `int x509_des3_getiv` (unsigned char \*s, unsigned char iv[8])

- static void `x509_des3_decrypt` (unsigned char des3\_iv[8], unsigned char \*buf, int buflen, unsigned char \*pwd, int pwrlen)
- int `x509_parse_key` (rsa\_context \*rsa, unsigned char \*buf, int buflen, unsigned char \*pwd, int pwrlen)  
*Parse a private RSA key.*
- int `x509_read_keyfile` (rsa\_context \*rsa, char \*path, char \*pwd)  
*Load and parse a private RSA key.*
- int `x509_dn_gets` (char \*buf, char \*end, x509\_name \*dn)  
*Store the certificate DN in printable form into buf; no more than (end - buf) characters will be written.*
- char \* `x509_cert_info` (x509\_cert \*crt)  
*Return an informational string about the certificate, or NULL if memory allocation failed.*
- int `x509_is_cert_expired` (x509\_cert \*crt)  
*Return 0 if the certificate is still valid, or BADCERT\_HAS\_EXPIRED.*
- static void `x509_hash` (unsigned char \*in, int len, int alg, unsigned char \*out)
- int `x509_verify_cert` (x509\_cert \*crt, x509\_cert \*trust\_ca, char \*cn, int \*flags)  
*Verify the certificate signature.*
- void `x509_free_cert` (x509\_cert \*crt)  
*Unallocate all certificate data.*
- int `x509_self_test` (int verbose)  
*Checkup routine.*

## Variables

- static const char `_x509_read_src` [] = "\_x509read\_src"

### 8.128.1 Define Documentation

#### 8.128.1.1 #define \_CRT\_SECURE\_NO\_DEPRECATED 1

Definition at line 36 of file x509read.c.

### 8.128.2 Function Documentation

#### 8.128.2.1 static int `asn1_get_bool` (unsigned char \*\*p, unsigned char \*end, int \*val) [static]

Definition at line 116 of file x509read.c.

References `ASN1_BOOLEAN`, `asn1_get_tag()`, and `ERR_ASN1_INVALID_LENGTH`.

Referenced by `x509_get_ext()`.

**8.128.2.2 static int asn1\_get\_int (unsigned char \*\**p*, unsigned char \* *end*, int \* *val*) [static]**

Definition at line 134 of file x509read.c.

References `asn1_get_tag()`, `ASN1_INTEGER`, and `ERR_ASN1_INVALID_LENGTH`.

Referenced by `x509_get_ext()`, `x509_get_version()`, and `x509_parse_key()`.

**8.128.2.3 static int asn1\_get\_len (unsigned char \*\**p*, unsigned char \* *end*, int \* *len*) [static]**

Definition at line 60 of file x509read.c.

References `ERR_ASN1_INVALID_LENGTH`, and `ERR_ASN1_OUT_OF_DATA`.

Referenced by `asn1_get_tag()`, `x509_get_name()`, and `x509_get_serial()`.

**8.128.2.4 static int asn1\_get\_mpi (unsigned char \*\**p*, unsigned char \* *end*, mpi \* *X*) [static]**

Definition at line 157 of file x509read.c.

References `asn1_get_tag()`, `ASN1_INTEGER`, and `mpi_read_binary()`.

Referenced by `x509_get_pubkey()`, and `x509_parse_key()`.

**8.128.2.5 static int asn1\_get\_tag (unsigned char \*\**p*, unsigned char \* *end*, int \* *len*, int *tag*) [static]**

Definition at line 101 of file x509read.c.

References `asn1_get_len()`, `ERR_ASN1_OUT_OF_DATA`, and `ERR_ASN1_UNEXPECTED_TAG`.

Referenced by `asn1_get_bool()`, `asn1_get_int()`, `asn1_get_mpi()`, `x509_add_certs()`, `x509_get_alg()`, `x509_get_dates()`, `x509_get_ext()`, `x509_get_name()`, `x509_get_pubkey()`, `x509_get_sig()`, `x509_get_uid()`, `x509_get_version()`, and `x509_parse_key()`.

**8.128.2.6 int x509\_add\_certs (x509\_cert \* *chain*, unsigned char \* *buf*, int *buflen*)**

Parse one or more certificates and add them to the chain.

**Parameters:**

*chain* points to the start of the chain  
*buf* buffer holding the certificate data  
*buflen* size of the buffer

**Returns:**

0 if successful, or a specific X509 error code

Definition at line 646 of file x509read.c.

References `ASN1_CONSTRUCTED`, `asn1_get_tag()`, `ASN1_SEQUENCE`, `base64_decode()`, `_x509_cert::ca_istrue`, `rsa_context::E`, `ERR_ASN1_LENGTH_MISMATCH`, `ERR_BASE64_INVALID_CHARACTER`, `ERR_X509_CERT_INVALID_FORMAT`, `ERR_X509_CERT_INVALID_PEM`, `ERR_X509_CERT_SIG_MISMATCH`, `ERR_X509_CERT_UNKNOWN_SIG_ALG`, `ERR_X509_CERT_UNKNOWN_VERSION`, `_x509_cert::issuer`, `_x509_cert::issuer_id`, `_x509_cert::issuer_raw`,

rsa\_context::len, \_x509\_buf::len, \_x509\_cert::max\_pathlen, mpi\_msb(), rsa\_context::N, \_x509\_cert::next, OID\_PKCS1, \_x509\_buf::p, \_x509\_cert::pk\_oid, \_x509\_cert::raw, \_x509\_cert::rsa, rsa\_check\_pubkey(), \_x509\_cert::serial, \_x509\_cert::sig, \_x509\_cert::sig\_oid1, \_x509\_cert::sig\_oid2, \_x509\_cert::subject, \_x509\_cert::subject\_id, \_x509\_cert::subject\_raw, \_x509\_cert::tbs, \_x509\_cert::v3\_ext, \_x509\_cert::valid\_from, \_x509\_cert::valid\_to, \_x509\_cert::version, x509\_add\_certs(), x509\_free\_cert(), x509\_get\_alg(), x509\_get\_dates(), x509\_get\_ext(), x509\_get\_name(), x509\_get\_pubkey(), x509\_get\_serial(), x509\_get\_sig(), x509\_get\_uid(), and x509\_get\_version().

Referenced by main(), x509\_add\_certs(), and x509\_read\_crtfile().

### 8.128.2.7 char\* x509\_cert\_info (x509\_cert \* crt)

Return an informational string about the certificate, or NULL if memory allocation failed.

Definition at line 1391 of file x509read.c.

References \_x509\_time::day, \_x509\_time::hour, \_x509\_cert::issuer, \_x509\_buf::len, \_x509\_time::min, \_x509\_time::mon, mpi::n, rsa\_context::N, \_x509\_buf::p, \_x509\_cert::rsa, RSA\_MD2, RSA\_MD4, RSA\_MD5, RSA\_SHA1, \_x509\_time::sec, \_x509\_cert::serial, \_x509\_cert::sig\_oid1, \_x509\_cert::subject, \_x509\_cert::valid\_from, \_x509\_cert::valid\_to, \_x509\_cert::version, x509\_dn\_gets(), and \_x509\_time::year.

### 8.128.2.8 static void x509\_des3\_decrypt (unsigned char des3\_iv[8], unsigned char \* buf, int buflen, unsigned char \* pwd, int pwrlen) [static]

Definition at line 1038 of file x509read.c.

References des3\_cbc\_decrypt(), des3\_set\_3keys(), md5\_finish(), md5\_starts(), and md5\_update().

Referenced by x509\_parse\_key().

### 8.128.2.9 static int x509\_des3\_getiv (unsigned char \* s, unsigned char iv[8]) [static]

Definition at line 1013 of file x509read.c.

References ERR\_X509\_KEY\_INVALID\_ENC\_IV.

Referenced by x509\_parse\_key().

### 8.128.2.10 int x509\_dn\_gets (char \* buf, char \* end, x509\_name \* dn)

Store the certificate DN in printable form into buf; no more than (end - buf) characters will be written.

Definition at line 1308 of file x509read.c.

References \_x509\_buf::len, \_x509\_name::next, \_x509\_name::oid, OID\_PKCS9, OID\_X520, \_x509\_buf::p, PKCS9\_EMAIL, \_x509\_name::val, X520\_COMMON\_NAME, X520\_COUNTRY, X520\_LOCALITY, X520\_ORG\_UNIT, X520\_ORGANIZATION, and X520\_STATE.

Referenced by x509\_cert\_info().

### 8.128.2.11 void x509\_free\_cert (x509\_cert \* crt)

Unallocate all certificate data.

Definition at line 1598 of file x509read.c.

References `_x509_cert::issuer`, `_x509_cert::next`, `_x509_name::next`, `_x509_buf::p`, `_x509_cert::raw`, `_x509_cert::rsa`, `rsa_free()`, and `_x509_cert::subject`.

Referenced by `main()`, `ssl_free()`, and `x509_add_certs()`.

**8.128.2.12** `static int x509_get_alg (unsigned char **p, unsigned char *end, x509_buf *alg)`  
[static]

Definition at line 237 of file `x509read.c`.

References `ASN1_CONSTRUCTED`, `asn1_get_tag()`, `ASN1_NULL`, `ASN1_OID`, `ASN1_SEQUENCE`, `ERR_ASN1_LENGTH_MISMATCH`, `ERR_X509_CERT_INVALID_ALG`, `_x509_buf::len`, `_x509_buf::p`, and `_x509_buf::tag`.

Referenced by `x509_add_certs()`, and `x509_get_pubkey()`.

**8.128.2.13** `static int x509_get_dates (unsigned char **p, unsigned char *end, x509_time *from, x509_time *to)` [static]

Definition at line 366 of file `x509read.c`.

References `ASN1_CONSTRUCTED`, `asn1_get_tag()`, `ASN1_SEQUENCE`, `ASN1_UTCTIME`, `_x509_time::day`, `ERR_ASN1_LENGTH_MISMATCH`, `ERR_X509_CERT_INVALID_DATE`, `_x509_time::hour`, `_x509_time::min`, `_x509_time::mon`, `_x509_time::sec`, and `_x509_time::year`.

Referenced by `x509_add_certs()`.

**8.128.2.14** `static int x509_get_ext (unsigned char **p, unsigned char *end, x509_buf *ext, int *ca_istrue, int *max_pathlen)` [static]

Definition at line 538 of file `x509read.c`.

References `ASN1_CONSTRUCTED`, `ASN1_CONTEXT_SPECIFIC`, `asn1_get_bool()`, `asn1_get_int()`, `asn1_get_tag()`, `ASN1_OCTET_STRING`, `ASN1_SEQUENCE`, `ERR_ASN1_LENGTH_MISMATCH`, `ERR_ASN1_UNEXPECTED_TAG`, `ERR_X509_CERT_INVALID_EXTENSIONS`, `_x509_buf::len`, `_x509_buf::p`, and `_x509_buf::tag`.

Referenced by `x509_add_certs()`.

**8.128.2.15** `static int x509_get_name (unsigned char **p, unsigned char *end, x509_name *cur)`  
[static]

Definition at line 284 of file `x509read.c`.

References `ASN1_BMP_STRING`, `ASN1_CONSTRUCTED`, `asn1_get_len()`, `asn1_get_tag()`, `ASN1_IA5_STRING`, `ASN1_OID`, `ASN1_PRINTABLE_STRING`, `ASN1_SEQUENCE`, `ASN1_SET`, `ASN1_T61_STRING`, `ASN1_UNIVERSAL_STRING`, `ASN1_UTF8_STRING`, `ERR_ASN1_LENGTH_MISMATCH`, `ERR_ASN1_OUT_OF_DATA`, `ERR_ASN1_UNEXPECTED_TAG`, `ERR_X509_CERT_INVALID_NAME`, `_x509_buf::len`, `_x509_name::next`, `_x509_name::oid`, `_x509_buf::p`, `_x509_buf::tag`, and `_x509_name::val`.

Referenced by `x509_add_certs()`.



**8.128.2.16** `static int x509_get_pubkey (unsigned char **p, unsigned char * end, x509_buf *  
pk_alg_oid, mpi * N, mpi * E)` [static]

Definition at line 429 of file x509read.c.

References ASN1\_BIT\_STRING, ASN1\_CONSTRUCTED, asn1\_get\_mpi(), asn1\_get\_tag(), ASN1\_SEQUENCE, ERR\_ASN1\_LENGTH\_MISMATCH, ERR\_ASN1\_OUT\_OF\_DATA, ERR\_X509\_CERT\_INVALID\_PUBKEY, ERR\_X509\_CERT\_UNKNOWN\_PK\_ALG, \_x509\_buf::len, OID\_PKCS1\_RSA, \_x509\_buf::p, and x509\_get\_alg().

Referenced by x509\_add\_certs().

**8.128.2.17** `static int x509_get_serial (unsigned char **p, unsigned char * end, x509_buf * serial)`  
[static]

Definition at line 206 of file x509read.c.

References ASN1\_CONTEXT\_SPECIFIC, asn1\_get\_len(), ASN1\_INTEGER, ASN1\_PRIMITIVE, ERR\_ASN1\_OUT\_OF\_DATA, ERR\_ASN1\_UNEXPECTED\_TAG, ERR\_X509\_CERT\_INVALID\_SERIAL, \_x509\_buf::len, \_x509\_buf::p, and \_x509\_buf::tag.

Referenced by x509\_add\_certs().

**8.128.2.18** `static int x509_get_sig (unsigned char **p, unsigned char * end, x509_buf * sig)`  
[static]

Definition at line 484 of file x509read.c.

References ASN1\_BIT\_STRING, asn1\_get\_tag(), ERR\_X509\_CERT\_INVALID\_SIGNATURE, \_x509\_buf::len, \_x509\_buf::p, and \_x509\_buf::tag.

Referenced by x509\_add\_certs().

**8.128.2.19** `static int x509_get_uid (unsigned char **p, unsigned char * end, x509_buf * uid, int n)`  
[static]

Definition at line 509 of file x509read.c.

References ASN1\_CONSTRUCTED, ASN1\_CONTEXT\_SPECIFIC, asn1\_get\_tag(), ERR\_ASN1\_UNEXPECTED\_TAG, \_x509\_buf::len, \_x509\_buf::p, and \_x509\_buf::tag.

Referenced by x509\_add\_certs().

**8.128.2.20** `static int x509_get_version (unsigned char **p, unsigned char * end, int * ver)`  
[static]

Definition at line 176 of file x509read.c.

References ASN1\_CONSTRUCTED, ASN1\_CONTEXT\_SPECIFIC, asn1\_get\_int(), asn1\_get\_tag(), ERR\_ASN1\_LENGTH\_MISMATCH, ERR\_ASN1\_UNEXPECTED\_TAG, and ERR\_X509\_CERT\_INVALID\_VERSION.

Referenced by x509\_add\_certs().

**8.128.2.21** `static void x509_hash (unsigned char * in, int len, int alg, unsigned char * out)`  
`[static]`

Definition at line 1475 of file x509read.c.

References md2(), md4(), md5(), RSA\_MD2, RSA\_MD4, RSA\_MD5, RSA\_SHA1, and sha1().

Referenced by x509\_verify\_cert().

**8.128.2.22** `int x509_is_cert_expired (x509_cert * crt)`

Return 0 if the certificate is still valid, or BADCERT\_HAS\_EXPIRED.

Definition at line 1452 of file x509read.c.

References BADCERT\_HAS\_EXPIRED, \_x509\_time::day, \_x509\_time::mon, \_x509\_cert::valid\_to, and \_x509\_time::year.

Referenced by x509\_verify\_cert().

**8.128.2.23** `int x509_parse_key (rsa_context * rsa, unsigned char * buf, int buflen, unsigned char * pwd, int pwdlen)`

Parse a private RSA key.

#### Parameters:

*rsa* RSA context to be initialized

*buf* input buffer

*buflen* size of the buffer

*pwd* password for decryption (optional)

*pwdlen* size of the password

#### Returns:

0 if successful, or a specific X509 error code

Definition at line 1077 of file x509read.c.

References ASN1\_CONSTRUCTED, asnl\_get\_int(), asnl\_get\_mpi(), asnl\_get\_tag(), ASN1\_SEQUENCE, base64\_decode(), rsa\_context::D, rsa\_context::DP, rsa\_context::DQ, rsa\_context::E, ERR\_ASN1\_LENGTH\_MISMATCH, ERR\_BASE64\_INVALID\_CHARACTER, ERR\_X509\_FEATURE\_UNAVAILABLE, ERR\_X509\_KEY\_INVALID\_ENC\_IV, ERR\_X509\_KEY\_INVALID\_FORMAT, ERR\_X509\_KEY\_INVALID\_PEM, ERR\_X509\_KEY\_INVALID\_VERSION, ERR\_X509\_KEY\_PASSWORD\_MISMATCH, ERR\_X509\_KEY\_PASSWORD\_REQUIRED, ERR\_X509\_KEY\_UNKNOWN\_ENC\_ALG, rsa\_context::len, mpi\_msb(), rsa\_context::N, rsa\_context::P, rsa\_context::Q, rsa\_context::QP, rsa\_check\_privkey(), rsa\_free(), rsa\_context::ver, x509\_des3\_decrypt(), and x509\_des3\_getiv().

Referenced by main(), and x509\_read\_keyfile().

**8.128.2.24** `int x509_read_crtfile (x509_cert * chain, char * path)`

Load one or more certificates and add them to the chain.

**Parameters:**

*chain* points to the start of the chain  
*path* filename to read the certificates from

**Returns:**

0 if successful, or a specific X509 error code

Definition at line 974 of file x509read.c.

References f, and x509\_add\_certs().

**8.128.2.25 int x509\_read\_keyfile (rsa\_context \* *rsa*, char \* *path*, char \* *password*)**

Load and parse a private RSA key.

**Parameters:**

*rsa* RSA context to be initialized  
*path* filename to read the private key from  
*pwd* password to decrypt the file (can be NULL)

**Returns:**

0 if successful, or a specific X509 error code

Definition at line 1261 of file x509read.c.

References f, and x509\_parse\_key().

**8.128.2.26 int x509\_self\_test (int *verbose*)**

Checkup routine.

**Returns:**

0 if successful, or 1 if the test failed

Definition at line 1726 of file x509read.c.

Referenced by main().

**8.128.2.27 int x509\_verify\_cert (x509\_cert \* *crt*, x509\_cert \* *trust\_ca*, char \* *cn*, int \* *flags*)**

Verify the certificate signature.

**Parameters:**

*crt* a certificate to be verified  
*trust\_ca* the trusted CA chain  
*cn* expected Common Name (can be set to NULL if the CN must not be verified)  
*flags* result of the verification

**Returns:**

0 if successful or ERR\_X509\_SIG\_VERIFY\_FAILED, in which case \*flags will have one or more of the following values set: BADCERT\_HAS\_EXPIRED – BADCERT\_CN\_MISMATCH – BADCERT\_NOT\_TRUSTED

Definition at line 1497 of file x509read.c.

References BADCERT\_CN\_MISMATCH, BADCERT\_NOT\_TRUSTED, \_x509\_cert::ca\_istrue, ERR\_X509\_SIG\_VERIFY\_FAILED, \_x509\_cert::issuer\_raw, \_x509\_buf::len, \_x509\_cert::max\_pathlen, \_x509\_cert::next, \_x509\_name::next, \_x509\_name::oid, \_x509\_buf::p, \_x509\_cert::rsa, rsa\_pkcs1\_verify(), \_x509\_cert::sig, \_x509\_cert::sig\_oid1, \_x509\_cert::subject, \_x509\_cert::subject\_raw, \_x509\_cert::tbs, \_x509\_name::val, \_x509\_cert::version, x509\_hash(), and x509\_is\_cert\_expired().

### 8.128.3 Variable Documentation

**8.128.3.1** `const char _x509_read_src[] = "_x509read_src"` [static]

Definition at line 1647 of file x509read.c.

## 8.129 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/aes/aescript2.c File Reference

```
#include <sys/types.h>
#include <unistd.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include "xyssl/aes.h"
#include "xyssl/sha2.h"
```

### Defines

- `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`
- `#define MODE\_ENCRYPT 0`
- `#define MODE\_DECRYPT 1`
- `#define USAGE`

### Functions

- `void scanf\_argv (char *prompt, char **arg)`
- `int main (int argc, char *argv[ ])`

#### 8.129.1 Define Documentation

##### 8.129.1.1 `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

Definition at line 25 of file aescript2.c.

##### 8.129.1.2 `#define MODE\_DECRYPT 1`

Definition at line 44 of file aescript2.c.

Referenced by `main()`.

##### 8.129.1.3 `#define MODE\_ENCRYPT 0`

Definition at line 43 of file aescript2.c.

Referenced by `main()`.

##### 8.129.1.4 `#define USAGE`

### Value:

```
"\n aescrypt2 <mode> <input filename> <output filename> <key>\n" \  
"\n  <mode>: 0 = encrypt, 1 = decrypt\n" \  
"\n  example: aescrypt2 0 file file.aes hex:E76B2413958B00E193\n" \  
"\n"
```

Definition at line 46 of file aescrypt2.c.

Referenced by main().

## 8.129.2 Function Documentation

### 8.129.2.1 int main (int *argc*, char \* *argv*[])

Definition at line 60 of file aescrypt2.c.

References aes\_decrypt(), aes\_encrypt(), aes\_set\_key(), int, MODE\_DECRYPT, MODE\_ENCRYPT, scanf\_argv(), sha2\_finish(), sha2\_hmac\_finish(), sha2\_hmac\_starts(), sha2\_hmac\_update(), sha2\_starts(), sha2\_update(), and USAGE.

### 8.129.2.2 void scanf\_argv (char \* *prompt*, char \*\* *arg*)

Definition at line 52 of file aescrypt2.c.

Referenced by main().

## 8.130 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/hash/hello.c File Reference

```
#include <stdio.h>
#include "xyssl/md5.h"
```

### Defines

- `#define _CRT_SECURE_NO_DEPRECATED 1`

### Functions

- `int main (void)`

#### 8.130.1 Define Documentation

##### 8.130.1.1 `#define _CRT_SECURE_NO_DEPRECATED 1`

Definition at line 25 of file hello.c.

#### 8.130.2 Function Documentation

##### 8.130.2.1 `int main (void)`

Definition at line 32 of file hello.c.

References `md5()`.

## 8.131 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/hash/md5sum.c File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/md5.h"
```

### Defines

- `#define _CRT_SECURE_NO_DEPRECATED 1`

### Functions

- `int md5_wrapper` (char \*filename, unsigned char \*sum)
- `int md5_print` (char \*filename)
- `int md5_check` (char \*filename)
- `int main` (int argc, char \*argv[])

#### 8.131.1 Define Documentation

##### 8.131.1.1 `#define _CRT_SECURE_NO_DEPRECATED 1`

Definition at line 25 of file md5sum.c.

#### 8.131.2 Function Documentation

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

Definition at line 134 of file md5sum.c.

References `md5_check()`, and `md5_print()`.

##### 8.131.2.2 `int md5_check` (char \* *filename*)

Definition at line 61 of file md5sum.c.

References `f`, and `md5_wrapper()`.

Referenced by `main()`.

##### 8.131.2.3 `int md5_print` (char \* *filename*)

Definition at line 46 of file md5sum.c.

References `md5_wrapper()`.

Referenced by `main()`.



**8.131.2.4 int md5\_wrapper (char \**filename*, unsigned char \**sum*)**

Definition at line 33 of file md5sum.c.

References md5\_file().

Referenced by md5\_check(), and md5\_print().

## 8.132 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/hash/sha1sum.c File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/sha1.h"
```

### Defines

- `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

### Functions

- `int sha1\_wrapper (char *filename, unsigned char *sum)`
- `int sha1\_print (char *filename)`
- `int sha1\_check (char *filename)`
- `int main (int argc, char *argv[])`

#### 8.132.1 Define Documentation

##### 8.132.1.1 `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

Definition at line 25 of file sha1sum.c.

#### 8.132.2 Function Documentation

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

Definition at line 134 of file sha1sum.c.

References [sha1\\_check\(\)](#), and [sha1\\_print\(\)](#).

##### 8.132.2.2 `int sha1\_check (char * filename)`

Definition at line 61 of file sha1sum.c.

References [f](#), and [sha1\\_wrapper\(\)](#).

Referenced by [main\(\)](#).

##### 8.132.2.3 `int sha1\_print (char * filename)`

Definition at line 46 of file sha1sum.c.

References [sha1\\_wrapper\(\)](#).

Referenced by [main\(\)](#).

**8.132.2.4 int sha1\_wrapper (char \* *filename*, unsigned char \* *sum*)**

Definition at line 33 of file sha1sum.c.

References sha1\_file().

Referenced by sha1\_check(), and sha1\_print().

## 8.133 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/hash/sha2sum.c File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/sha2.h"
```

### Defines

- `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

### Functions

- `int sha2\_wrapper (char *filename, unsigned char *sum)`
- `int sha2\_print (char *filename)`
- `int sha2\_check (char *filename)`
- `int main (int argc, char *argv[ ])`

#### 8.133.1 Define Documentation

##### 8.133.1.1 `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

Definition at line 25 of file sha2sum.c.

#### 8.133.2 Function Documentation

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

Definition at line 134 of file sha2sum.c.

References [sha2\\_check\(\)](#), and [sha2\\_print\(\)](#).

##### 8.133.2.2 `int sha2\_check (char * filename)`

Definition at line 61 of file sha2sum.c.

References [f](#), and [sha2\\_wrapper\(\)](#).

Referenced by [main\(\)](#).

##### 8.133.2.3 `int sha2\_print (char * filename)`

Definition at line 46 of file sha2sum.c.

References [sha2\\_wrapper\(\)](#).

Referenced by [main\(\)](#).

**8.133.2.4 int sha2\_wrapper (char \**filename*, unsigned char \* *sum*)**

Definition at line 33 of file sha2sum.c.

References sha2\_file().

Referenced by sha2\_check(), and sha2\_print().

## 8.134 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/dh\_client.c File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/net.h"
#include "xyssl/aes.h"
#include "xyssl/dhm.h"
#include "xyssl/rsa.h"
#include "xyssl/sha1.h"
#include "xyssl/havege.h"
```

### Defines

- `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`
- `#define SERVER\_NAME "localhost"`
- `#define SERVER\_PORT 11999`

### Functions

- `int main (void)`

#### 8.134.1 Define Documentation

##### 8.134.1.1 `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

Definition at line 25 of file `dh_client.c`.

##### 8.134.1.2 `#define SERVER\_NAME "localhost"`

Definition at line 38 of file `dh_client.c`.

Referenced by `main()`.

##### 8.134.1.3 `#define SERVER\_PORT 11999`

Definition at line 39 of file `dh_client.c`.

Referenced by `main()`.

#### 8.134.2 Function Documentation

##### 8.134.2.1 `int main (void)`

Definition at line 41 of file `dh_client.c`.

References aes\_decrypt(), aes\_set\_key(), dhm\_calc\_secret(), dhm\_free(), dhm\_make\_public(), dhm\_read\_params(), f, havege\_init(), havege\_rand(), rsa\_context::len, dhm\_context::len, net\_close(), net\_connect(), net\_recv(), net\_send(), rsa\_free(), rsa\_pkcs1\_verify(), rsa\_read\_public(), RSA\_SHA1, server\_fd, SERVER\_NAME, SERVER\_PORT, and sha1().

## 8.135 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/dh\_genprime.c File Reference

```
#include <stdio.h>
#include "xyssl/havege.h"
#include "xyssl/bignum.h"
```

### Defines

- `#define _CRT_SECURE_NO_DEPRECATED` 1
- `#define DH_P_SIZE` 1024
- `#define GENERATOR` "4"

### Functions

- `int main` (void)

#### 8.135.1 Define Documentation

##### 8.135.1.1 `#define _CRT_SECURE_NO_DEPRECATED` 1

Definition at line 25 of file dh\_genprime.c.

##### 8.135.1.2 `#define DH_P_SIZE` 1024

Definition at line 37 of file dh\_genprime.c.

Referenced by main().

##### 8.135.1.3 `#define GENERATOR` "4"

Definition at line 38 of file dh\_genprime.c.

Referenced by main().

#### 8.135.2 Function Documentation

##### 8.135.2.1 `int main` (void)

Definition at line 40 of file dh\_genprime.c.

References DH\_P\_SIZE, GENERATOR, havege\_init(), havege\_rand(), mpi\_div\_int(), mpi\_free(), mpi\_gen\_prime(), mpi\_init(), mpi\_is\_prime(), mpi\_read\_string(), mpi\_sub\_int(), mpi\_write\_file(), and P.



## 8.136 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/dh\_server.c File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/net.h"
#include "xyssl/aes.h"
#include "xyssl/dhm.h"
#include "xyssl/rsa.h"
#include "xyssl/sha1.h"
#include "xyssl/havege.h"
```

### Defines

- `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`
- `#define SERVER\_PORT 11999`
- `#define PLAINTEXT "0123456_89ABCDE_"`

### Functions

- `int main (void)`

#### 8.136.1 Define Documentation

##### 8.136.1.1 `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

Definition at line 25 of file dh\_server.c.

##### 8.136.1.2 `#define PLAINTEXT "0123456_89ABCDE_"`

Definition at line 39 of file dh\_server.c.

Referenced by `main()`.

##### 8.136.1.3 `#define SERVER\_PORT 11999`

Definition at line 38 of file dh\_server.c.

#### 8.136.2 Function Documentation

##### 8.136.2.1 `int main (void)`

Definition at line 41 of file dh\_server.c.

References `aes_encrypt()`, `aes_set_key()`, `client_fd`, `dhm_calc_secret()`, `dhm_free()`, `dhm_make_params()`, `dhm_read_public()`, `f`, `dhm_context::G`, `havege_init()`, `havege_rand()`, `dhm_context::len`, `rsa_context::len`,

mpi\_read\_file(), net\_accept(), net\_bind(), net\_close(), net\_recv(), net\_send(), dhm\_context::P, PLAIN-TEXT, rsa\_free(), rsa\_pkcs1\_sign(), rsa\_read\_private(), RSA\_SHA1, SERVER\_PORT, and sha1().

## 8.137 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/mpi\_demo.c File Reference

```
#include <stdio.h>
#include "xyssl/bignum.h"
```

### Defines

- `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

### Functions

- `int main (void)`

#### 8.137.1 Define Documentation

##### 8.137.1.1 `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

Definition at line 25 of file mpi\_demo.c.

#### 8.137.2 Function Documentation

##### 8.137.2.1 `int main (void)`

Definition at line 32 of file mpi\_demo.c.

References `mpi\_exp\_mod\(\)`, `mpi\_free\(\)`, `mpi\_init\(\)`, `mpi\_inv\_mod\(\)`, `mpi\_mul\_mpi\(\)`, `mpi\_read\_string\(\)`, `mpi\_sub\_int\(\)`, `mpi\_write\_file\(\)`, and `P`.

## 8.138 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/rsa\_genkey.c File Reference

```
#include <stdio.h>
#include "xyssl/havege.h"
#include "xyssl/bignum.h"
#include "xyssl/rsa.h"
```

### Defines

- `#define _CRT_SECURE_NO_DEPRECATED 1`
- `#define KEY_SIZE 1024`
- `#define EXPONENT 65537`

### Functions

- `int main (void)`

#### 8.138.1 Define Documentation

##### 8.138.1.1 `#define _CRT_SECURE_NO_DEPRECATED 1`

Definition at line 25 of file `rsa_genkey.c`.

##### 8.138.1.2 `#define EXPONENT 65537`

Definition at line 35 of file `rsa_genkey.c`.

Referenced by `main()`.

##### 8.138.1.3 `#define KEY_SIZE 1024`

Definition at line 34 of file `rsa_genkey.c`.

Referenced by `main()`.

#### 8.138.2 Function Documentation

##### 8.138.2.1 `int main (void)`

Definition at line 37 of file `rsa_genkey.c`.

References `EXPONENT`, `havege_init()`, `havege_rand()`, `KEY_SIZE`, `rsa_free()`, `rsa_gen_key()`, `rsa_write_private()`, and `rsa_write_public()`.

## 8.139 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/rsa\_sign.c File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/rsa.h"
#include "xyssl/sha1.h"
```

### Defines

- `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

### Functions

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

#### 8.139.1 Define Documentation

##### 8.139.1.1 `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

Definition at line 25 of file `rsa_sign.c`.

#### 8.139.2 Function Documentation

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

Definition at line 34 of file `rsa_sign.c`.

References `f`, `rsa_context::len`, `rsa_pkcs1_sign()`, `rsa_read_private()`, `RSA_SHA1`, and `sha1_file()`.

## 8.140 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/rsa\_verify.c File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/rsa.h"
#include "xyssl/sha1.h"
```

### Defines

- `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

### Functions

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

#### 8.140.1 Define Documentation

##### 8.140.1.1 `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

Definition at line 25 of file `rsa_verify.c`.

#### 8.140.2 Function Documentation

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

Definition at line 34 of file `rsa_verify.c`.

References `f`, `rsa_context::len`, `rsa_pkcs1_verify()`, `rsa_read_public()`, `RSA_SHA1`, and `sha1_file()`.

## 8.141 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/ssl/ssl\_client1.c File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/net.h"
#include "xyssl/ssl.h"
#include "xyssl/havege.h"
```

### Defines

- `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`
- `#define SERVER\_PORT 4433`
- `#define SERVER\_NAME "xyssl.org"`
- `#define GET\_REQUEST`

### Functions

- `int main (void)`

#### 8.141.1 Define Documentation

##### 8.141.1.1 `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

Definition at line 25 of file `ssl_client1.c`.

##### 8.141.1.2 `#define GET\_REQUEST`

###### Value:

```
"GET /hello/ HTTP/1.0\r\n" \
"Host: xyssl.org\r\n\r\n"
```

Definition at line 41 of file `ssl_client1.c`.

Referenced by `main()`.

##### 8.141.1.3 `#define SERVER\_NAME "xyssl.org"`

Definition at line 40 of file `ssl_client1.c`.

##### 8.141.1.4 `#define SERVER\_PORT 4433`

Definition at line 35 of file `ssl_client1.c`.

## 8.141.2 Function Documentation

### 8.141.2.1 `int main (void)`

Definition at line 45 of file `ssl_client1.c`.

References `ERR_NET_WOULD_BLOCK`, `ERR_SSL_PEER_CLOSE_NOTIFY`, `GET_REQUEST`, `havege_init()`, `havege_rand()`, `net_close()`, `net_connect()`, `server_fd`, `SERVER_NAME`, `SERVER_PORT`, `ssl_close_notify()`, `ssl_default_ciphers`, `ssl_free()`, `ssl_init()`, `SSL_IS_CLIENT`, `ssl_read()`, `ssl_set_authmode()`, `ssl_set_ciphlist()`, `ssl_set_endpoint()`, `ssl_set_io_files()`, `ssl_set_rng_func()`, `SSL_VERIFY_NONE`, and `ssl_write()`.



## 8.142 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/ssl/ssl\_client2.c File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/net.h"
#include "xyssl/ssl.h"
#include "xyssl/havege.h"
#include "xyssl/certs.h"
#include "xyssl/x509.h"
```

### Defines

- `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`
- `#define SERVER\_PORT 4433`
- `#define SERVER\_NAME "xyssl.org"`
- `#define GET\_REQUEST`

### Functions

- `int main (void)`

#### 8.142.1 Define Documentation

##### 8.142.1.1 `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

Definition at line 25 of file ssl\_client2.c.

##### 8.142.1.2 `#define GET\_REQUEST`

###### Value:

```
"GET /hello/ HTTP/1.0\r\n" \
"Host: xyssl.org\r\n\r\n"
```

Definition at line 43 of file ssl\_client2.c.

##### 8.142.1.3 `#define SERVER\_NAME "xyssl.org"`

Definition at line 42 of file ssl\_client2.c.

##### 8.142.1.4 `#define SERVER\_PORT 4433`

Definition at line 37 of file ssl\_client2.c.

## 8.142.2 Function Documentation

### 8.142.2.1 `int main (void)`

Definition at line 47 of file `ssl_client2.c`.

References `BADCERT_CN_MISMATCH`, `BADCERT_HAS_EXPIRED`, `BADCERT_NOT_TRUSTED`, `ERR_SSL_PEER_CLOSE_NOTIFY`, `GET_REQUEST`, `havege_init()`, `havege_rand()`, `net_close()`, `net_connect()`, `rsa_free()`, `server_fd`, `SERVER_NAME`, `SERVER_PORT`, `ssl_close_notify()`, `ssl_default_ciphers`, `ssl_free()`, `ssl_get_cipher_name()`, `ssl_get_verify_result()`, `ssl_handshake()`, `ssl_init()`, `SSL_IS_CLIENT`, `ssl_read()`, `ssl_set_authmode()`, `ssl_set_ca_chain()`, `ssl_set_ciphlist()`, `ssl_set_endpoint()`, `ssl_set_io_files()`, `ssl_set_rng_func()`, `ssl_set_rsa_cert()`, `SSL_VERIFY_OPTIONAL`, `ssl_write()`, `test_cli_cert`, `test_cli_key`, `x509_add_certs()`, `x509_free_cert()`, `x509_parse_key()`, and `xyssl_ca_cert`.

## 8.143 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/ssl/ssl\_server.c File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/havege.h"
#include "xyssl/certs.h"
#include "xyssl/x509.h"
#include "xyssl/ssl.h"
#include "xyssl/net.h"
```

### Defines

- `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`
- `#define HTTP\_RESPONSE`

### Functions

- `int main (void)`

### Variables

- `char * dhm\_P`
- `char * dhm\_G = "4"`
- `unsigned char session\_table [SSL_SESSION_TBL_LEN]`
- `int my\_preferred\_ciphers []`

### 8.143.1 Define Documentation

#### 8.143.1.1 `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

Definition at line 25 of file `ssl_server.c`.

#### 8.143.1.2 `#define HTTP\_RESPONSE`

##### Value:

```
"HTTP/1.0 200 OK\r\nContent-Type: text/html\r\n\r\n" \
"<h2><p><center>Successful connection using: %s\r\n"
```

Definition at line 41 of file `ssl_server.c`.

Referenced by `http\_ParseHeader\(\)`, `main\(\)`, and `mtp\_http\_Parse\(\)`.

## 8.143.2 Function Documentation

### 8.143.2.1 `int main (void)`

Definition at line 78 of file `ssl_server.c`.

References `client_fd`, `dhm_G`, `dhm_P`, `ERR_NET_CONN_RESET`, `ERR_SSL_PEER_CLOSE_NOTIFY`, `havege_init()`, `havege_rand()`, `HTTP_RESPONSE`, `my_preferred_ciphers`, `net_accept()`, `net_bind()`, `net_close()`, `_x509_cert::next`, `rsa_free()`, `session_table`, `ssl_close_notify()`, `ssl_free()`, `ssl_get_cipher_name()`, `ssl_init()`, `SSL_IS_SERVER`, `ssl_read()`, `ssl_server_start()`, `ssl_set_authmode()`, `ssl_set_ca_chain()`, `ssl_set_ciphlist()`, `ssl_set_dhm_vals()`, `ssl_set_endpoint()`, `ssl_set_io_files()`, `ssl_set_rng_func()`, `ssl_set_rsa_cert()`, `ssl_set_sidtable()`, `SSL_VERIFY_NONE`, `ssl_write()`, `test_ca_cert`, `test_srv_cert`, `test_srv_key`, `x509_add_certs()`, `x509_free_cert()`, and `x509_parse_key()`.

## 8.143.3 Variable Documentation

### 8.143.3.1 `char* dhm_G = "4"`

Definition at line 60 of file `ssl_server.c`.

Referenced by `main()`.

### 8.143.3.2 `char* dhm_P`

**Initial value:**

```
"E4004C1F94182000103D883A448B3F80"
"2CE4B44A83301270002C20D0321CFD00"
"11CCEF784C26A400F43DFB901BCA7538"
"F2C6B176001CF5A0FD16D2C48B1D0C1C"
"F6AC8E1DA6BCC3B4E1F96B0564965300"
"FFA1D0B601EB2800F489AA512C4B248C"
"01F76949A60BB7F00A40B1EAB64BDD48"
"E8A700D60B7F1200FA8E77B0A979DABF"
```

Definition at line 50 of file `ssl_server.c`.

Referenced by `main()`.

### 8.143.3.3 `int my_preferred_ciphers[]`

**Initial value:**

```
{
    TLS1_EDH_RSA_AES_256_SHA,
    SSL3_EDH_RSA_DES_168_SHA,
    TLS1_RSA_AES_256_SHA,
    SSL3_RSA_DES_168_SHA,
    SSL3_RSA_RC4_128_SHA,
    SSL3_RSA_RC4_128_MD5,
    0
}
```

Definition at line 67 of file `ssl_server.c`.

Referenced by `main()`.

#### 8.143.3.4 unsigned char session\_table[SSL\_SESSION\_TBL\_LEN]

Definition at line 62 of file ssl\_server.c.

Referenced by main().

## 8.144 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/test/benchmark.c File Reference

```
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include "xyssl/md4.h"
#include "xyssl/md5.h"
#include "xyssl/sha1.h"
#include "xyssl/sha2.h"
#include "xyssl/arc4.h"
#include "xyssl/des.h"
#include "xyssl/aes.h"
#include "xyssl/rsa.h"
#include "xyssl/timing.h"
```

### Defines

- `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`
- `#define BUFSIZE 510`

### Functions

- `int myrand (void *rng_state)`
- `int main (void)`

#### 8.144.1 Define Documentation

##### 8.144.1.1 `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

Definition at line 25 of file benchmark.c.

##### 8.144.1.2 `#define BUFSIZE 510`

Definition at line 42 of file benchmark.c.

Referenced by `main()`.

#### 8.144.2 Function Documentation

##### 8.144.2.1 `int main (void)`

Definition at line 50 of file benchmark.c.

References aes\_cbc\_encrypt(), aes\_set\_key(), alarmed, arc4\_crypt(), arc4\_setup(), BUFSIZE, rsa\_context::D, des3\_cbc\_encrypt(), des3\_set\_3keys(), des\_cbc\_encrypt(), des\_set\_key(), rsa\_context::DP, rsa\_context::DQ, rsa\_context::E, hardclock(), KEY\_LEN, rsa\_context::len, md4(), md5(), mpi\_read\_string(), rsa\_context::N, rsa\_context::P, rsa\_context::Q, rsa\_context::QP, RSA\_D, RSA\_DP, RSA\_DQ, RSA\_E, rsa\_free(), RSA\_N, RSA\_P, rsa\_private(), rsa\_public(), RSA\_Q, RSA\_QP, set\_alarm(), sha1(), and sha2().

#### 8.144.2.2 int myrand (void \* *rng\_state*)

Definition at line 44 of file benchmark.c.

## 8.145 /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/test/selftest.c File Reference

```
#include <string.h>
#include <stdio.h>
#include "xyssl/md2.h"
#include "xyssl/md4.h"
#include "xyssl/md5.h"
#include "xyssl/sha1.h"
#include "xyssl/sha2.h"
#include "xyssl/arc4.h"
#include "xyssl/des.h"
#include "xyssl/aes.h"
#include "xyssl/bignum.h"
#include "xyssl/base64.h"
#include "xyssl/rsa.h"
#include "xyssl/x509.h"
```

### Defines

- `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

### Functions

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

#### 8.145.1 Define Documentation

##### 8.145.1.1 `#define \_CRT\_SECURE\_NO\_DEPRECATED 1`

Definition at line 25 of file selftest.c.

#### 8.145.2 Function Documentation

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

Definition at line 44 of file selftest.c.

References `aes_self_test()`, `arc4_self_test()`, `base64_self_test()`, `des_self_test()`, `md2_self_test()`, `md4_self_test()`, `md5_self_test()`, `mpi_self_test()`, `rsa_self_test()`, `sha1_self_test()`, `sha2_self_test()`, and `x509_self_test()`.



## 8.146 /home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAclMe File Reference

### Namespaces

- namespace [LibMC](#)
- namespace [System](#)
- namespace [System::Collections::Generic](#)
- namespace [System::Text](#)

### Data Structures

- class [LibMC::MCAclMessage](#)  
*Encapsulates ACL messages in the Mobile-C library.*

#### 8.146.1 Detailed Description

Defines the MCAclMessage object and its member functions.

Definition in file [MCAclMessage.cs](#).

## 8.147 /home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAgency File Reference

### Namespaces

- namespace [LibMC](#)
- namespace [System::Runtime::InteropServices](#)

### Data Structures

- class [LibMC::MCAgency](#)  
*Wrapper class for [MCAgency\\_t](#) structure.*

#### 8.147.1 Detailed Description

Defines the MCAgency object and its member functions.

Definition in file [MCAgency.cs](#).

## 8.148 /home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAgent.cs File Reference

### Namespaces

- namespace [LibMC](#)

### Data Structures

- class [LibMC::MCAgent](#)  
*Wrapper class for MCAgent\_t structure.*

### 8.148.1 Detailed Description

Defines the MCAgent object and its member functions.

Definition in file [MCAgent.cs](#).

## 8.149 /home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCExport File Reference

### Namespaces

- namespace [LibMC](#)

### Data Structures

- class [LibMC::MCAgency](#)  
*Wrapper class for [MCAgency\\_t](#) structure.*
- struct [LibMC::MCAgency::MCAgency\\_t](#)
- struct [LibMC::MCAgency::MCAgencyOptions\\_t](#)
- struct [LibMC::MCAgency::ChOptions\\_t](#)  
*ChOptions structures.*

#### 8.149.1 Detailed Description

Imports functions, structs, and enums from the Mobile-C library.

Definition in file [MCExports.cs](#).

## 8.150 /home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/Properties File Reference

### Namespaces

- namespace [System::Reflection](#)
- namespace [System::Runtime::CompilerServices](#)

## 8.151 /home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/Properties File Reference

### Namespaces

- namespace [LibMC::Properties](#)

### Data Structures

- class [LibMC::Properties::Settings](#)

## 8.152 /home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/Settings.cs File Reference

### Namespaces

- namespace [LibMC::Properties](#)

### Data Structures

- class [LibMC::Properties::Settings](#)

## 8.153 /home/dko/projects/mobilec/trunk/src/winconfig.h File Reference

### Defines

- #define `PACKAGE_STRING` "MobileC V1.10.0"
- #define `PACKAGE_VERSION` "1.10.0"
- #define `strtok_r`(buf, delim, save\_ptr) strtok( buf, delim )

#### 8.153.1 Define Documentation

##### 8.153.1.1 #define `PACKAGE_STRING` "MobileC V1.10.0"

Definition at line 4 of file winconfig.h.

##### 8.153.1.2 #define `PACKAGE_VERSION` "1.10.0"

Definition at line 5 of file winconfig.h.

Referenced by `mtp_http_ComposeMessage()`, and `mtp_http_CreateMessage()`.

##### 8.153.1.3 #define `strtok_r`(buf, delim, save\_ptr) strtok( buf, delim )

Definition at line 6 of file winconfig.h.

Referenced by `agent_xml_parse__fill_row_data()`, `message_InitializeFromAgent()`, and `message_Send()`.



## 8.154 /home/dko/projects/mobilec/trunk/src/xml\_compose.c File Reference

```
#include <mxml.h>
#include "include/agent.h"
#include "include/xml_compose.h"
#include "include/xml_helper.h"
```

### Functions

- [mxml\\_node\\_t \\* agent\\_xml\\_compose \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_gaf\\_message \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_message \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_mobile\\_agent \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_agent\\_data \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_name \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_owner \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_home \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_tasks \(agent\\_p agent\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_task \(agent\\_p agent, int index\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_data \(agent\\_p agent, int index, interpreter\\_variable\\_data\\_t \\*interp\\_variable\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_agent\\_code \(agent\\_p agent, int index\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_row \(interpreter\\_variable\\_data\\_t \\*interp\\_variable, int index\)](#)
- [mxml\\_node\\_t \\* agent\\_xml\\_compose\\_\\_create\\_row\\_nodes \(void \\*data, int index, int \\*extent, ChType\\_t type, int dim, int extent\\_index\)](#)

### 8.154.1 Function Documentation

#### 8.154.1.1 mxml\_node\_t\* agent\_xml\_compose (agent\_p agent)

Definition at line 41 of file `xml_compose.c`.

References `agent_xml_compose__gaf_message()`, `MXML_ADD_AFTER`, `MXML_ADD_TO_PARENT`, `MXML_NO_CALLBACK`, `mxmlAdd()`, `mxmlLoadString()`, and `node`.

Referenced by `message_InitializeFromAgent()`.

#### 8.154.1.2 mxml\_node\_t\* agent\_xml\_compose\_\_agent\_code (agent\_p agent, int index)

Definition at line 501 of file `xml_compose.c`.

References `agent_datastate_s::agent_code_ids`, `agent_datastate_s::agent_codes`, `agent_s::datastate`, `MXML_NO_PARENT`, `mxmlElementSetAttr()`, `mxmlNewElement()`, `node`, and `xml_new_cdata()`.

Referenced by `agent_xml_compose__tasks()`.

**8.154.1.3 mxml\_node\_t\* agent\_xml\_compose\_\_agent\_data (agent\_p agent)**

Definition at line 145 of file xml\_compose.c.

References agent\_xml\_compose\_\_home(), agent\_xml\_compose\_\_name(), agent\_xml\_compose\_\_owner(), agent\_xml\_compose\_\_tasks(), MXML\_ADD\_AFTER, mxmlAdd(), mxmlNewElement(), and node.

Referenced by agent\_xml\_compose\_\_mobile\_agent().

**8.154.1.4 mxml\_node\_t\* agent\_xml\_compose\_\_create\_row\_nodes (void \* data, int index, int \* extent, ChType\_t type, int dim, int extent\_index)**

Definition at line 552 of file xml\_compose.c.

References agent\_xml\_compose\_\_create\_row\_nodes(), CH\_DATATYPE\_SIZE, CH\_DATATYPE\_VALUE\_STRING, MXML\_ADD\_AFTER, MXML\_ADD\_TO\_PARENT, MXML\_NO\_PARENT, mxmlAdd(), mxmlElementSetAttr(), mxmlNewElement(), mxmlNewText(), node, and size.

Referenced by agent\_xml\_compose\_\_create\_row\_nodes(), and agent\_xml\_compose\_\_row().

**8.154.1.5 mxml\_node\_t\* agent\_xml\_compose\_\_data (agent\_p agent, int index, interpreter\_variable\_data\_t \* interp\_variable)**

Definition at line 427 of file xml\_compose.c.

References agent\_xml\_compose\_\_row(), interpreter\_variable\_data\_s::array\_dim, CH\_DATATYPE\_STRING, CH\_DATATYPE\_VALUE\_STRING, interpreter\_variable\_data\_s::data, interpreter\_variable\_data\_s::data\_type, MXML\_ADD\_AFTER, mxmlAdd(), mxmlElementSetAttr(), mxmlNewElement(), interpreter\_variable\_data\_s::name, and node.

Referenced by agent\_xml\_compose\_\_task().

**8.154.1.6 mxml\_node\_t\* agent\_xml\_compose\_\_gaf\_message (agent\_p agent)**

Definition at line 61 of file xml\_compose.c.

References agent\_xml\_compose\_\_message(), MXML\_ADD\_AFTER, mxmlAdd(), mxmlNewElement(), and node.

Referenced by agent\_xml\_compose().

**8.154.1.7 mxml\_node\_t\* agent\_xml\_compose\_\_home (agent\_p agent)**

Definition at line 237 of file xml\_compose.c.

References agent\_s::home, mxmlNewElement(), mxmlNewText(), and node.

Referenced by agent\_xml\_compose\_\_agent\_data().

**8.154.1.8 mxml\_node\_t\* agent\_xml\_compose\_\_message (agent\_p agent)**

Definition at line 80 of file xml\_compose.c.

References `agent_s::agent_type`, `agent_xml_compose__mobile_agent()`, `MC_LOCAL_AGENT`, `MC_REMOTE_AGENT`, `MC_RETURN_AGENT`, `MXML_ADD_AFTER`, `mxmlAdd()`, `mxmlElementSetAttr()`, `mxmlNewElement()`, and `node`.

Referenced by `agent_xml_compose__gaf_message()`.

#### 8.154.1.9 `mxml_node_t* agent_xml_compose__mobile_agent (agent_p agent)`

Definition at line 124 of file `xml_compose.c`.

References `agent_xml_compose__agent_data()`, `MXML_ADD_AFTER`, `mxmlAdd()`, `mxmlNewElement()`, and `node`.

Referenced by `agent_xml_compose__message()`.

#### 8.154.1.10 `mxml_node_t* agent_xml_compose__name (agent_p agent)`

Definition at line 205 of file `xml_compose.c`.

References `mxmlNewElement()`, `mxmlNewText()`, `agent_s::name`, and `node`.

Referenced by `agent_xml_compose__agent_data()`.

#### 8.154.1.11 `mxml_node_t* agent_xml_compose__owner (agent_p agent)`

Definition at line 221 of file `xml_compose.c`.

References `mxmlNewElement()`, `mxmlNewText()`, `node`, and `agent_s::owner`.

Referenced by `agent_xml_compose__agent_data()`.

#### 8.154.1.12 `mxml_node_t* agent_xml_compose__row (interpreter_variable_data_t * interp_variable, int index)`

Definition at line 529 of file `xml_compose.c`.

References `agent_xml_compose__create_row_nodes()`, `interpreter_variable_data_s::array_dim`, `interpreter_variable_data_s::array_extent`, `interpreter_variable_data_s::data`, `interpreter_variable_data_s::data_type`, and `node`.

Referenced by `agent_xml_compose__data()`.

#### 8.154.1.13 `mxml_node_t* agent_xml_compose__task (agent_p agent, int index)`

Definition at line 310 of file `xml_compose.c`.

References `agent_task_s::agent_return_data`, `agent_task_s::agent_variable_list`, `agent_xml_compose__data()`, `agent_task_s::code_id`, `agent_s::datastate`, `MXML_ADD_AFTER`, `mxmlAdd()`, `mxmlElementSetAttr()`, `mxmlNewElement()`, `node`, `agent_task_s::persistent`, `agent_datastate_s::persistent`, `agent_task_s::server_name`, `agent_task_s::task_completed`, `agent_datastate_s::tasks`, and `agent_task_s::var_name`.

Referenced by `agent_xml_compose__tasks()`.

**8.154.1.14 mxml\_node\_t\* agent\_xml\_compose\_\_tasks (agent\_p *agent*)**

Definition at line 253 of file xml\_compose.c.

References agent\_xml\_compose\_\_agent\_code(), agent\_xml\_compose\_\_task(), agent\_s::datastate, MXML\_ADD\_AFTER, mxmlAdd(), mxmlElementSetAttr(), mxmlNewElement(), node, agent\_datastate\_s::number\_of\_tasks, and agent\_datastate\_s::task\_progress.

Referenced by agent\_xml\_compose\_\_agent\_data().

## 8.155 /home/dko/projects/mobilec/trunk/src/xml\_helper.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <mxml.h>
#include "include/xml_helper.h"
#include "include/macros.h"
```

### Functions

- `mxml_node_t * xml_find_sibling` (const `mxml_node_t *node`, const char \*`sibling_name`)
- char \* `xml_get_cdata` (const `mxml_node_t *node`)
- `mxml_node_t * xml_get_child` (const `mxml_node_t *node`, const char \*`child_name`, int `descend`)
- `mxml_node_t * xml_get_deep_child` (const `mxml_node_t *parent`, const char \*\*`child_path`)
- `mxml_node_t * xml_get_next_element` (const `mxml_node_t *node`)
- char \* `xml_get_text` (const `mxml_node_t *node`)
- const char \* `xml_get_element_name` (const `mxml_node_t *node`)
- `mxml_node_t * xml_new_cdata` (`mxml_node_t *parent`, const char \*`text`)
- const char \* `whitespace_cb` (`mxml_node_t *node`, int `where`)

### 8.155.1 Function Documentation

#### 8.155.1.1 const char\* whitespace\_cb (mxml\_node\_t \* node, int where)

Definition at line 249 of file `xml_helper.c`.

References `mxml_node_s::child`, `mxml_value_u::element`, `MXML_WS_AFTER_CLOSE`, `MXML_WS_AFTER_OPEN`, `MXML_WS_BEFORE_CLOSE`, `MXML_WS_BEFORE_OPEN`, `mxml_value_s::name`, `mxml_node_s::parent`, and `mxml_node_s::value`.

Referenced by `main()`.

#### 8.155.1.2 mxml\_node\_t\* xml\_find\_sibling (const mxml\_node\_t \* node, const char \* sibling\_name)

Definition at line 49 of file `xml_helper.c`.

References `MXML_NO_DESCEND`, `mxmlFindElement()`, `mxml_node_s::parent`, and `xml_get_element_name()`.

#### 8.155.1.3 char\* xml\_get\_cdata (const mxml\_node\_t \* node)

Definition at line 71 of file `xml_helper.c`.

References `CHECK_NULL`, `MXML_ELEMENT`, `mxml_node_s::type`, and `xml_get_element_name()`.

Referenced by `xml_get_text()`.

**8.155.1.4 mxml\_node\_t\* xml\_get\_child (const mxml\_node\_t \* *node*, const char \* *child\_name*, int *descend*)**

Definition at line 104 of file xml\_helper.c.

References mxmlFindElement().

Referenced by agent\_return\_xml\_parse(), agent\_xml\_parse\_\_agent\_data(), agent\_xml\_parse\_\_data(), agent\_xml\_parse\_\_mobile\_agent(), and xml\_get\_deep\_child().

**8.155.1.5 mxml\_node\_t\* xml\_get\_deep\_child (const mxml\_node\_t \* *parent*, const char \*\* *child\_path*)**

Definition at line 123 of file xml\_helper.c.

References MXML\_NO\_DESCEND, node, and xml\_get\_child().

**8.155.1.6 const char\* xml\_get\_element\_name (const mxml\_node\_t \* *node*)**

Definition at line 217 of file xml\_helper.c.

References mxml\_value\_u::element, MXML\_ELEMENT, mxml\_value\_s::name, mxml\_node\_s::type, and mxml\_node\_s::value.

Referenced by agent\_xml\_parse\_\_data(), agent\_xml\_parse\_\_mobile\_agent(), agent\_xml\_parse\_\_row(), xml\_find\_sibling(), and xml\_get\_cdata().

**8.155.1.7 mxml\_node\_t\* xml\_get\_next\_element (const mxml\_node\_t \* *node*)**

Definition at line 137 of file xml\_helper.c.

References MXML\_ELEMENT, mxml\_node\_s::next, and mxml\_node\_s::type.

**8.155.1.8 char\* xml\_get\_text (const mxml\_node\_t \* *node*)**

Definition at line 155 of file xml\_helper.c.

References CHECK\_NULL, mxml\_node\_s::child, mxml\_value\_u::element, MXML\_ELEMENT, MXML\_TEXT, mxml\_value\_s::name, mxml\_node\_s::next, mxml\_text\_s::string, mxml\_value\_u::text, mxml\_node\_s::type, mxml\_node\_s::value, and xml\_get\_cdata().

Referenced by agent\_xml\_parse\_\_agent\_code(), agent\_xml\_parse\_\_home(), agent\_xml\_parse\_\_name(), and agent\_xml\_parse\_\_owner().

**8.155.1.9 mxml\_node\_t\* xml\_new\_cdata (mxml\_node\_t \* *parent*, const char \* *text*)**

Definition at line 230 of file xml\_helper.c.

References CHECK\_NULL, mxmlNewElement(), and node.

Referenced by agent\_xml\_compose\_\_agent\_code().

## 8.156 /home/dko/projects/mobilec/trunk/src/xml\_parser.c File Reference

```
#include <mxml.h>
#include <string.h>
#include <stdlib.h>
#include "config.h"
#include "include/interpreter_variable_data.h"
#include "include/message.h"
#include "include/xml_parser.h"
#include "include/xml_helper.h"
```

### Defines

- `#define \_XOPEN\_SOURCE 600`

### Functions

- `error\_code\_t agent\_xml\_parse (agent\_p agent)`
- `error\_code\_t agent\_xml\_parse\_\_mobile\_agent (agent\_p agent, xml\_parser\_p xml_parser)`
- `error\_code\_t agent\_xml\_parse\_\_agent\_data (agent\_p agent, xml\_parser\_p xml_parser)`
- `error\_code\_t agent\_xml\_parse\_\_name (agent\_p agent, xml\_parser\_p xml_parser)`
- `error\_code\_t agent\_xml\_parse\_\_owner (agent\_p agent, xml\_parser\_p xml_parser)`
- `error\_code\_t agent\_xml\_parse\_\_home (agent\_p agent, xml\_parser\_p xml_parser)`
- `error\_code\_t agent\_xml\_parse\_\_tasks (agent\_p agent, xml\_parser\_p xml_parser)`
- `error\_code\_t agent\_xml\_parse\_\_task (agent\_p agent, xml\_parser\_p xml_parser, int index)`
- `error\_code\_t agent\_xml\_parse\_\_data (agent\_p agent, xml\_parser\_p xml_parser, int index)`
- `error\_code\_t agent\_xml\_parse\_\_row (interpreter\_variable\_data\_t *interp_variable, xml\_parser\_p xml_parser, int index)`
- `void agent\_xml\_parse\_\_fill\_row\_data (void *data, ChType\_t type, int *extent, const mxml\_node\_t *node, int *index)`
- `error\_code\_t agent\_xml\_parse\_\_agent\_code (agent\_p agent, int index, xml\_parser\_p xml_parser)`
- `error\_code\_t agent\_return\_xml\_parse (agent\_p agent)`
- `error\_code\_t message\_xml\_parse (message\_p message)`
- `error\_code\_t message\_xml\_parse\_\_message (message\_p message, xml\_parser\_p xml_parser)`

### 8.156.1 Define Documentation

#### 8.156.1.1 `#define \_XOPEN\_SOURCE 600`

Definition at line 38 of file `xml_parser.c`.

## 8.156.2 Function Documentation

### 8.156.2.1 `error_code_t agent_return_xml_parse (agent_p agent)`

Definition at line 823 of file `xml_parser.c`.

References `agent_xml_parse__home()`, `agent_xml_parse__name()`, `agent_xml_parse__owner()`, `agent_xml_parse__tasks()`, `agent_s::datastate`, `MC_SUCCESS`, `xml_get_child()`, and `agent_datastate_s::xml_root`.

### 8.156.2.2 `error_code_t agent_xml_parse (agent_p agent)`

Definition at line 52 of file `xml_parser.c`.

References `agent_xml_parse__mobile_agent()`, `agent_s::datastate`, `MC_SUCCESS`, and `agent_datastate_s::xml_agent_root`.

Referenced by `agent_initialize()`.

### 8.156.2.3 `error_code_t agent_xml_parse__agent_code (agent_p agent, int index, xml_parser_p xml_parser)`

Definition at line 782 of file `xml_parser.c`.

References `agent_datastate_s::agent_code`, `agent_datastate_s::agent_code_ids`, `agent_datastate_s::agent_codes`, `agent_task_s::code_id`, `agent_s::datastate`, `MC_SUCCESS`, `mxmlElementGetAttr()`, `agent_datastate_s::number_of_tasks`, `agent_datastate_s::task_progress`, `agent_datastate_s::tasks`, and `xml_get_text()`.

Referenced by `agent_xml_parse__tasks()`.

### 8.156.2.4 `error_code_t agent_xml_parse__agent_data (agent_p agent, xml_parser_p xml_parser)`

Definition at line 93 of file `xml_parser.c`.

References `agent_xml_parse__home()`, `agent_xml_parse__name()`, `agent_xml_parse__owner()`, `agent_xml_parse__tasks()`, `MC_ERR_PARSE`, `MC_SUCCESS`, and `xml_get_child()`.

Referenced by `agent_xml_parse__mobile_agent()`.

### 8.156.2.5 `error_code_t agent_xml_parse__data (agent_p agent, xml_parser_p xml_parser, int index)`

Definition at line 457 of file `xml_parser.c`.

References `agent_task_s::agent_return_data`, `agent_task_s::agent_variable_list`, `agent_xml_parse__row()`, `interpreter_variable_data_s::array_dim`, `CH_DATATYPE_SIZE`, `CH_DATATYPE_STR_TO_VAL`, `CH_STRING_DATATYPE`, `interpreter_variable_data_s::data`, `interpreter_variable_data_s::data_type`, `agent_s::datastate`, `interpreter_variable_data_New()`, `MC_ERR_PARSE`, `MC_SUCCESS`, `mxmlElementGetAttr()`, `interpreter_variable_data_s::name`, `mxml_node_s::parent`, `agent_task_s::persistent`, `agent_datastate_s::tasks`, `xml_get_child()`, and `xml_get_element_name()`.

Referenced by `agent_xml_parse__task()`.



**8.156.2.6 void agent\_xml\_parse\_\_fill\_row\_data (void \* *data*, ChType\_t *type*, int \* *extent*, const mxml\_node\_t \* *node*, int \* *index*)**

Definition at line 666 of file xml\_parser.c.

References agent\_xml\_parse\_\_fill\_row\_data(), CH\_DATATYPE\_SIZE, mxml\_node\_s::child, MXML\_DESCEND\_FIRST, MXML\_ELEMENT, MXML\_TEXT, mxmlFindElement(), mxml\_text\_s::string, strtok\_r, mxml\_value\_u::text, mxml\_node\_s::type, and mxml\_node\_s::value.

Referenced by agent\_xml\_parse\_\_fill\_row\_data(), and agent\_xml\_parse\_\_row().

**8.156.2.7 error\_code\_t agent\_xml\_parse\_\_home (agent\_p *agent*, xml\_parser\_p *xml\_parser*)**

Definition at line 201 of file xml\_parser.c.

References CHECK\_NULL, agent\_s::home, MC\_SUCCESS, and xml\_get\_text().

Referenced by agent\_return\_xml\_parse(), and agent\_xml\_parse\_\_agent\_data().

**8.156.2.8 error\_code\_t agent\_xml\_parse\_\_mobile\_agent (agent\_p *agent*, xml\_parser\_p *xml\_parser*)**

Definition at line 65 of file xml\_parser.c.

References agent\_xml\_parse\_\_agent\_data(), MC\_ERR\_PARSE, xml\_get\_child(), and xml\_get\_element\_name().

Referenced by agent\_xml\_parse().

**8.156.2.9 error\_code\_t agent\_xml\_parse\_\_name (agent\_p *agent*, xml\_parser\_p *xml\_parser*)**

Definition at line 148 of file xml\_parser.c.

References CHECK\_NULL, MC\_ERR\_PARSE, MC\_SUCCESS, agent\_s::name, and xml\_get\_text().

Referenced by agent\_return\_xml\_parse(), and agent\_xml\_parse\_\_agent\_data().

**8.156.2.10 error\_code\_t agent\_xml\_parse\_\_owner (agent\_p *agent*, xml\_parser\_p *xml\_parser*)**

Definition at line 174 of file xml\_parser.c.

References CHECK\_NULL, MC\_SUCCESS, agent\_s::owner, and xml\_get\_text().

Referenced by agent\_return\_xml\_parse(), and agent\_xml\_parse\_\_agent\_data().

**8.156.2.11 error\_code\_t agent\_xml\_parse\_\_row (interpreter\_variable\_data\_t \* *interp\_variable*, xml\_parser\_p *xml\_parser*, int *index*)**

Definition at line 602 of file xml\_parser.c.

References agent\_xml\_parse\_\_fill\_row\_data(), interpreter\_variable\_data\_s::array\_dim, interpreter\_variable\_data\_s::array\_extent, CH\_DATATYPE\_SIZE, interpreter\_variable\_data\_s::data, interpreter\_variable\_data\_s::data\_type, MC\_SUCCESS, and xml\_get\_element\_name().

Referenced by agent\_xml\_parse\_\_data().

**8.156.2.12 error\_code\_t agent\_xml\_parse\_\_task (agent\_p agent, xml\_parser\_p xml\_parser, int index)**

Definition at line 365 of file xml\_parser.c.

References agent\_xml\_parse\_\_data(), CHECK\_NULL, agent\_task\_s::code\_id, agent\_s::datastate, MC\_ERR\_PARSE, MC\_SUCCESS, MXML\_DESCEND\_FIRST, MXML\_NO\_DESCEND, mxmlElementGetAttr(), mxmlFindElement(), agent\_task\_s::server\_name, agent\_task\_s::task\_completed, agent\_datastate\_s::tasks, and agent\_task\_s::var\_name.

Referenced by agent\_xml\_parse\_\_tasks().

**8.156.2.13 error\_code\_t agent\_xml\_parse\_\_tasks (agent\_p agent, xml\_parser\_p xml\_parser)**

Definition at line 227 of file xml\_parser.c.

References agent\_datastate\_s::agent\_code, agent\_datastate\_s::agent\_code\_ids, agent\_datastate\_s::agent\_codes, agent\_task\_New(), agent\_xml\_parse\_\_agent\_code(), agent\_xml\_parse\_\_task(), agent\_s::datastate, MC\_ERR\_PARSE, MXML\_DESCEND, MXML\_DESCEND\_FIRST, MXML\_NO\_DESCEND, mxmlElementGetAttr(), mxmlFindElement(), agent\_datastate\_s::number\_of\_tasks, agent\_datastate\_s::task\_progress, and agent\_datastate\_s::tasks.

Referenced by agent\_return\_xml\_parse(), and agent\_xml\_parse\_\_agent\_data().

**8.156.2.14 error\_code\_t message\_xml\_parse (message\_p message)**

Definition at line 858 of file xml\_parser.c.

Referenced by acc\_Thread(), and message\_InitializeFromConnection().

**8.156.2.15 error\_code\_t message\_xml\_parse\_\_message (message\_p message, xml\_parser\_p xml\_parser)**

Definition at line 913 of file xml\_parser.c.

## **Chapter 9**

# **Mobile-C Example Documentation**

### **9.1 LibMCConsole/Program.cs**

Basic Mobile-C console demo program

## 9.2 LibMCCppEx/LibMCCppEx.cpp

Demonstrates using LibMC.NET from a VC++ program.

## 9.3 LibMCFipaTest/Program.cs

Mobile-C FIPA ACL message demo program.

## 9.4 LibMCGui/Form1.cs

Basic Mobile-C Windows Forms demo program

## 9.5 LibMCMiscTest/Program.cs

Demonstrates miscellaneous Mobile-C functions.

## 9.6 LibMCVbEx/Form1.vb

Demonstrates using LibMC.NET from a VB program.



## Chapter 10

# Mobile-C Page Documentation

### 10.1 MCAclMessage

Examples of commonly used MCAclMessage operations:

Create a new, blank ACL message:

```
MCAclMessage tmp = new MCAclMessage();  
tmp.New();
```

Set the performative field:

```
tmp.SetPerformative(MCAclMessage.MC_FipaPerformative_e.FIPA_INFORM);
```

Set the sender:

```
tmp.SetSender("agency", "http://" + host + ":" +  
    localport.ToString() + "/acc");
```

Add an alternate reply-to field:

```
tmp.AddReplyTo("mobagent2", "http://" + host + ":" +  
    localport.ToString() + "/acc");
```

Add a receiver to the message:

```
tmp.AddReceiver("mobagent1", "http://" + host + ":" +  
    localport.ToString() + "/acc");
```

Set the content of the message:

```
tmp.SetContent("This is content. Yay!");
```

Finally, send and destroy the message:

```
Agency.AclSend(tmp);  
tmp.Destroy();
```

Note that messages contain a pointer to allocated unmanaged memory and need to be disposed of after they are used. The agency creates a copy of the message when it is sent, and therefore the MCAclMessage object is no longer needed.

## 10.2 Installing LibMC.NET

Installing LibMC.NET is straightforward but involves several steps.

### 10.2.1 Requirements

In order to use LibMC.NET you will need the following:

- Ch version 6.0.0 or greater from <http://www.softintegration.com/>.
- Embedded Ch version 6.0.0 or greater, also from <http://www.softintegration.com/>.
- Mobile-C 1.10.0 or greater. See Section [Downloading Mobile-C](#) for instructions on how to obtain Mobile-C.
- Visual Studio 2005 or later. Express versions of Visual Studio can be found at <http://www.microsoft.com/express/>.

### 10.2.2 Downloading Mobile-C

First, you must obtain a version of the Mobile-C source code. If you are reading this, chances are you have already completed this step. If you have not already downloaded the source code, it can be done in one of three ways:

- Download a supported release of Mobile-C. Visit [the Mobile-C website](#) for more information on supported releases.
- Download the latest source (unsupported) from [Sourceforge](#). This will give you the most current version of Mobile-C, but not necessarily the most stable version.
- Check out the latest source code from the SVN repository. This requires that you have a subversion client installed. More information can be found at [this location](#).

### 10.2.3 Building the Mobile-C Libraries

Once you have obtained the Mobile-C source, please see the Mobile-C User's Guide for information on compiling Mobile-C under Windows. Currently, only the Visual Studio .NET 2005 project is supported for LibMC.NET. Section 2.3 of the User's Guide describes how to compile Mobile-C into a static library. For LibMC.NET, at least one of two configurations are required: the "Debug\_DLL" or "Release\_DLL" versions. To build either one, select the appropriate configuration (this replaces step 3 in the User's Guide, Section 2.3.1) and build the solution (step 4). Alternatively, you may select "Batch Build" from the "Build" menu and build all four possible configurations.

### 10.2.4 Install the Mobile-C Libraries

After building the Mobile-C DLL files, the project will automatically copy the files to the system directory. By default, the files are copied to C:/Windows/System32/. If your system is configured differently or you wish to change the installation directory, right-click on the mc\_lib\_win32 project in the Solution Explorer and select "Properties." In the mc\_lib\_win32 Property Pages treeview, select "Configuration Properties," then "Build Events," and finally "Post-Build Event." You can then change the "Command Line" field to copy the files to the directory of your choice. If you change the installation directory, be sure that your chosen directory is in the system path and that you remove any other versions of the files. You will also need to execute a "Rebuild" on the project to ensure the files are copied to the new location.

### 10.2.5 Build LibMC.NET

Once you have built the Mobile-C DLL files, you can build LibMC.NET. Open the LibMC.NET solution file located in the directory you installed or checked out Mobile-C to at src/win32/LibMC.NET/LibMC.Net.sln. From the "Build" menu, select "Rebuild Solution." You may want to build both the "Debug" and "Release" versions, or perform a batch build as described previously.

## 10.3 Getting Started

LibMC.NET is very easy to use. The demo programs provided with the download are a good place to start. Please see the Examples section for more information.

### 10.3.1 Build the Demo Programs

The LibMC.NET demo program solution is located in the directory you installed or checked out Mobile-C to at `demos/win32/LibMC.NET/LibMCDemos.sln`. As before, select "Rebuild Solution" from the "Build" menu. Note that the demo program solution contains the LibMC.NET project as well. You may also build LibMC.NET from within the demo program solution.

By default, the LibMCGui demo is selected in the demo program solution. You may run this program by selecting the "Debug" menu then "Start Debugging" or by pressing F5. Other demo programs can be started by right-clicking the project in the Solution Explorer and selecting "Debug" then "Start new instance." The demo programs have their own documentation as well. See the README file in `src/win32/LibMC.NET` or `demos/win32/LibMC.NET` for information on how to build the demo program documentation.

## 10.4 Using LibMC.NET

This section explains how to use the LibMC.NET class library in your .NET project. Currently, it only describes the process for using the library in a C# console or GUI application. Other languages, such as VB and managed C++, will require similar actions.

### 10.4.1 Create a Project

First, create the type of project you would like to use from the Visual Studio "Start Page" or the "File" menu. Select the name and location of the project as you would any other project. Second, add a reference to the configuration of LibMC.NET you would like to use. For debugging purposes, the "Debug" configuration is probably best. To add the reference, right click the "References" item in the Solution Explorer for the project you just created. Select the "Browse" tab and navigate to the output directory of the LibMC.NET project. The directory is located at src/win32/LibMC.NET/bin/Configuration/ in the Mobile-C source directory, where Configuration is either "Debug" or "Release." Select the DLL file and click "Ok." The References item in the Solution Explorer should now list "LibMC." Be sure to save the solution at this point.

### 10.4.2 Using LibMC.NET Classes and Functions

As with any other namespace, you must add the declaration "using LibMC;" to any file you want to have access to the class libraries. Once you have added the using statement, you can declare objects from the library as you normally would declare any other objects. See the example programs for more details.

### 10.4.3 Other Options

You may want to enable one or more features in your project that can help you use LibMC.NET or debug problems. If you add any XML files to your project, you probably will want to set their properties in the project to copy the files to the output directory. This is done by selecting the file in the Solution Explorer, opening its properties, and setting two fields:

- Set the "Build Action" field to "Content" if it is not already set. This will make the file part of the project should you decide to publish or package it.
- Set the "Copy to output directory" to "Copy if newer" or "Copy always." This will copy the file when you build the project.

There is also one important note regarding XML files in Visual Studio. *Do not create XML files from within Visual Studio.* The Visual Studio XML file template contains a few leading characters that specify the encoding of the file. They are hidden and you will not be able to change them. These characters are not currently supported by Mobile-C and will crash a receiving agency.

To open the project properties, right-click the project in the Solution Explorer and select "Properties." In the "Debug" pane, you may wish to set an alternate working directory for the project if you want easy access to XML files outside of the project. This is useful for debugging, but may result in errors finding files if you package the project or create an installer. In general, it is best to specify all files with full paths because the Mobile-C library loads from a different location than the project. If you would like to be able to debug the Mobile-C library, you should select the "Enable unmanaged code debugging" check box. This will allow you to more easily see any errors that may occur in the unmanaged library, though hopefully none will.

## 10.5 Common Operations

This section contains examples of commonly used operations for three main [LibMC](#) classes:

- [MCAgency](#) The mobile agent agency.
- [MCAgent](#) Mobile agents.
- [MCAclMessage](#) Agent communication language messages.

For complete programs and more detailed examples, see the Examples section.

## 10.6 MCAgency

Examples of commonly used MCAgency operations:

Declare an agency as a member of a class:

```
public static MCAgency Agency = new MCAgency();
```

Set the agency's port:

```
int temp = 5051;
Agency.Port = temp;
```

Start an agency:

```
int temp;
temp = Agency.Initialize();
if (temp != 0)
    Console.WriteLine("Initialize: " + temp.ToString());
```

Pause and resume an agency:

```
Agency.HaltAgency();
Agency.ResumeAgency();
```

Turn off the command prompt thread:

```
temp = Agency.SetThreadOff(MCAgency.MC_ThreadIndex_e.MC_THREAD_CP);
if (temp != 0)
    Console.WriteLine("SetThreadOff: " + temp.ToString());
```

Load an agent into a local agency:

```
String filename = "agent.xml";
try
{
    Agency.LoadAgentMigrationMessageFile(filename);
}
catch (Exception ex)
{
    Console.WriteLine("Error loading file: " + ex.Message);
}
```

Ideally, the file name should be specified absolutely.

Send an agent to a remote agency:

```
String filename = "agent.xml";
String ip = "192.168.23.93";
int port = 5051;
try
{
    Agency.SendAgentMigrationMessageFile(filename, ip, port);
}
catch (Exception ex)
{
    Console.WriteLine("Error sending file: " + ex.Message);
}
```

Find an agent by name:

```
MCAgent agent;
try
{
    agent = Agency.FindAgentByName("persistent1");
}
catch (Exception e)
{
    Console.WriteLine("Exception: " + e.Message);
}
```

Wait for an agent to arrive:

```
MCAgent agent;
Agency.ResetSignal();
try
{
    agent = Agency.WaitRetrieveAgent();
}
catch (Exception e)
{
    Console.WriteLine("Exception: " + e.Message);
}
```

Wait indefinitely while an agency runs:

```
Agency.MainLoop();
```



## 10.7 MCAgent

Examples of commonly used MCAgent operations: Find an agent by name (assumes an MCAgency named Agency):

```
MCAgent agent;
try
{
    agent = Agency.FindAgentByName("persistent1");
}
catch (Exception e)
{
    Console.WriteLine("Exception: " + e.Message);
}
```

Terminate an agent

```
int temp;
try
{
    temp = agent.TerminateAgent();
    Console.WriteLine("TerminateAgent() returned " +
        temp.ToString() + ".");
}
catch (Exception e)
{
    Console.WriteLine("Exception: " + e.Message);
}
```

Print information about an agent:

```
Console.WriteLine(agent.ToString());
Console.WriteLine(agent.GetAgentXMLString());
Console.WriteLine(agent.RetrieveAgentCode());
```

## 10.8 Todo List

Global **LibMC::MCAgency::SteerControl()** Test MC\_SteerControl, MC\_Steer.

Global **LibMC::MCAgency::Steer(IntPtr funcptr, IntPtr arg)** Implement MC\_Steer

Global **LibMC::MCAgency::RegisterService(MCAgent agent, int agentID, String agentName, String[] serviceNames, int agentType)** Test MC\_RegisterService and MC\_SearchForService.

Global **LibMC::MCAgency::SearchForService(String searchString, IntPtr agentNames, IntPtr serviceNames, IntPtr agentType)** Implement SearchForService

Global **LibMC::MCAgent::CallAgentFunc(String funcName, IntPtr retval, IntPtr varg)** Find a better way to handle retval and varg

Global **LibMC::MCAgent::GetAgentExecEngine()** Wrap MC\_GetAgentExecEngine with an object for the void\* pointer return type (Ch interpreter).

Global **LibMC::MCAgent::GetAgentReturnData(int task\_num, IntPtr data, IntPtr dim, IntPtr extent)** Implement GetAgentReturnData

## 10.9 Bug List

Global **LibMC::MCAgency::Steer**(IntPtr funcptr, IntPtr arg) MC\_Steer is not yet implemented.

Global **LibMC::MCAgency::SearchForService**(String searchString, IntPtr agentNames, IntPtr serviceNames, IntPtr agentNames) MC\_SearchForService is not yet implemented.

# Index

~foo\_c  
  foo\_c, 78  
~foo\_s  
  foo\_s, 81  
/home/dko/projects/mobilec/trunk/src/acc.c, 199  
/home/dko/projects/mobilec/trunk/src/agent.c, 202  
/home/dko/projects/mobilec/trunk/src/agent\_  
  datastate.c, 205  
/home/dko/projects/mobilec/trunk/src/agent\_  
  mailbox.c, 206  
/home/dko/projects/mobilec/trunk/src/agent\_  
  return\_data.c, 208  
/home/dko/projects/mobilec/trunk/src/agent\_task.c,  
  210  
/home/dko/projects/mobilec/trunk/src/ams.c, 211  
/home/dko/projects/mobilec/trunk/src/barrier.c, 213  
/home/dko/projects/mobilec/trunk/src/cmd\_  
  prompt.c, 215  
/home/dko/projects/mobilec/trunk/src/connection.c,  
  218  
/home/dko/projects/mobilec/trunk/src/data\_  
  structures.c, 219  
/home/dko/projects/mobilec/trunk/src/df.c, 221  
/home/dko/projects/mobilec/trunk/src/dynstring.c,  
  225  
/home/dko/projects/mobilec/trunk/src/fipa\_acl.c,  
  226  
/home/dko/projects/mobilec/trunk/src/fipa\_  
  envelope.c, 238  
/home/dko/projects/mobilec/trunk/src/include/acc.h,  
  246  
/home/dko/projects/mobilec/trunk/src/include/agent.h,  
  249  
/home/dko/projects/mobilec/trunk/src/include/agent\_  
  datastate.h, 252  
/home/dko/projects/mobilec/trunk/src/include/agent\_  
  lib.h, 254  
/home/dko/projects/mobilec/trunk/src/include/agent\_  
  mailbox.h, 263  
/home/dko/projects/mobilec/trunk/src/include/agent\_  
  task.h, 265  
/home/dko/projects/mobilec/trunk/src/include/ams.h,  
  267  
/home/dko/projects/mobilec/trunk/src/include/ap\_  
  queue\_template.h, 269  
/home/dko/projects/mobilec/trunk/src/include/barrier.h,  
  272  
/home/dko/projects/mobilec/trunk/src/include/cmd\_  
  prompt.h, 275  
/home/dko/projects/mobilec/trunk/src/include/commands.h,  
  277  
/home/dko/projects/mobilec/trunk/src/include/commands.x.h,  
  279  
/home/dko/projects/mobilec/trunk/src/include/connection.h,  
  281  
/home/dko/projects/mobilec/trunk/src/include/data\_  
  structures.h, 283  
/home/dko/projects/mobilec/trunk/src/include/df.h,  
  286  
/home/dko/projects/mobilec/trunk/src/include/df\_  
  request.x.h, 292  
/home/dko/projects/mobilec/trunk/src/include/dynstring.h,  
  293  
/home/dko/projects/mobilec/trunk/src/include/fipa\_  
  acl.h, 295  
/home/dko/projects/mobilec/trunk/src/include/fipa\_  
  acl\_envelope.h, 310  
/home/dko/projects/mobilec/trunk/src/include/host\_  
  id.h, 318  
/home/dko/projects/mobilec/trunk/src/include/interpreter\_  
  variable\_data.h, 319  
/home/dko/projects/mobilec/trunk/src/include/libmc.h,  
  321  
/home/dko/projects/mobilec/trunk/src/include/macros.h,  
  356  
/home/dko/projects/mobilec/trunk/src/include/mc\_  
  error.h, 367  
/home/dko/projects/mobilec/trunk/src/include/mc\_  
  platform.h, 369  
/home/dko/projects/mobilec/trunk/src/include/mc\_  
  rlock.h, 371  
/home/dko/projects/mobilec/trunk/src/include/message.h,  
  373  
/home/dko/projects/mobilec/trunk/src/include/mtp\_  
  http.h, 376  
/home/dko/projects/mobilec/trunk/src/include/xml\_  
  compose.h, 382  
/home/dko/projects/mobilec/trunk/src/include/xml\_  
  helper.h, 385

- /home/dko/projects/mobilec/trunk/src/include/xml\_-  
parser.h, 387
- /home/dko/projects/mobilec/trunk/src/libmc.c, 391
- /home/dko/projects/mobilec/trunk/src/mc\_list/list.c,  
431
- /home/dko/projects/mobilec/trunk/src/mc\_-  
list/list.h, 433
- /home/dko/projects/mobilec/trunk/src/mc\_-  
platform.c, 436
- /home/dko/projects/mobilec/trunk/src/mc\_-  
rlock.c, 438
- /home/dko/projects/mobilec/trunk/src/mc\_-  
sync/sync\_list.c, 440
- /home/dko/projects/mobilec/trunk/src/mc\_-  
sync/sync\_list.h, 442
- /home/dko/projects/mobilec/trunk/src/message.c,  
445
- /home/dko/projects/mobilec/trunk/src/message\_-  
queue.c, 448
- /home/dko/projects/mobilec/trunk/src/mtp\_http.c,  
449
- /home/dko/projects/mobilec/trunk/src/mxml-  
2.2.2/config.h, 453
- /home/dko/projects/mobilec/trunk/src/mxml-  
2.2.2/mxml-attr.c, 457
- /home/dko/projects/mobilec/trunk/src/mxml-  
2.2.2/mxml-entity.c, 458
- /home/dko/projects/mobilec/trunk/src/mxml-  
2.2.2/mxml-file.c, 460
- /home/dko/projects/mobilec/trunk/src/mxml-  
2.2.2/mxml-index.c, 467
- /home/dko/projects/mobilec/trunk/src/mxml-  
2.2.2/mxml-node.c, 469
- /home/dko/projects/mobilec/trunk/src/mxml-  
2.2.2/mxml-private.c, 472
- /home/dko/projects/mobilec/trunk/src/mxml-  
2.2.2/mxml-search.c, 474
- /home/dko/projects/mobilec/trunk/src/mxml-  
2.2.2/mxml-set.c, 475
- /home/dko/projects/mobilec/trunk/src/mxml-  
2.2.2/mxml-string.c, 477
- /home/dko/projects/mobilec/trunk/src/mxml-  
2.2.2/mxml.h, 478
- /home/dko/projects/mobilec/trunk/src/mxml-  
2.2.2/mxmldoc.c, 490
- /home/dko/projects/mobilec/trunk/src/mxml-  
2.2.2/test/class.cxx, 494
- /home/dko/projects/mobilec/trunk/src/mxml-  
2.2.2/test/enum.cxx, 495
- /home/dko/projects/mobilec/trunk/src/mxml-  
2.2.2/test/function.cxx, 496
- /home/dko/projects/mobilec/trunk/src/mxml-  
2.2.2/test/struct.cxx, 497
- /home/dko/projects/mobilec/trunk/src/mxml-  
2.2.2/testmxml.c, 498
- /home/dko/projects/mobilec/trunk/src/mxml-  
2.2.2/vcnet/config.h, 454
- /home/dko/projects/mobilec/trunk/src/mxml-  
2.2.2/vcnet2005/config.h, 455
- /home/dko/projects/mobilec/trunk/src/security/asm.c,  
500
- /home/dko/projects/mobilec/trunk/src/security/asm.h,  
501
- /home/dko/projects/mobilec/trunk/src/security/asm\_-  
message\_composer.c, 502
- /home/dko/projects/mobilec/trunk/src/security/asm\_-  
message\_composer.h, 503
- /home/dko/projects/mobilec/trunk/src/security/asm\_-  
message\_parser.c, 504
- /home/dko/projects/mobilec/trunk/src/security/asm\_-  
message\_parser.h, 505
- /home/dko/projects/mobilec/trunk/src/security/asm\_-  
node.c, 506
- /home/dko/projects/mobilec/trunk/src/security/asm\_-  
node.h, 507
- /home/dko/projects/mobilec/trunk/src/security/mc\_-  
dh.c, 508
- /home/dko/projects/mobilec/trunk/src/security/mc\_-  
dh.h, 509
- /home/dko/projects/mobilec/trunk/src/security/xyssl-  
0.7/include/xyssl/aes.h, 510
- /home/dko/projects/mobilec/trunk/src/security/xyssl-  
0.7/include/xyssl/arc4.h, 513
- /home/dko/projects/mobilec/trunk/src/security/xyssl-  
0.7/include/xyssl/base64.h, 515
- /home/dko/projects/mobilec/trunk/src/security/xyssl-  
0.7/include/xyssl/bignum.h, 517
- /home/dko/projects/mobilec/trunk/src/security/xyssl-  
0.7/include/xyssl/bn\_asm.h, 533
- /home/dko/projects/mobilec/trunk/src/security/xyssl-  
0.7/include/xyssl/certs.h, 535
- /home/dko/projects/mobilec/trunk/src/security/xyssl-  
0.7/include/xyssl/des.h, 537
- /home/dko/projects/mobilec/trunk/src/security/xyssl-  
0.7/include/xyssl/dhm.h, 542
- /home/dko/projects/mobilec/trunk/src/security/xyssl-  
0.7/include/xyssl/havege.h, 547
- /home/dko/projects/mobilec/trunk/src/security/xyssl-  
0.7/include/xyssl/md2.h, 549
- /home/dko/projects/mobilec/trunk/src/security/xyssl-  
0.7/include/xyssl/md4.h, 553
- /home/dko/projects/mobilec/trunk/src/security/xyssl-  
0.7/include/xyssl/md5.h, 557
- /home/dko/projects/mobilec/trunk/src/security/xyssl-  
0.7/include/xyssl/net.h, 562
- /home/dko/projects/mobilec/trunk/src/security/xyssl-  
0.7/include/xyssl/rsa.h, 567

- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/sha1.h, [579](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/sha2.h, [584](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/sha4.h, [589](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/ssl.h, [594](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/ssl\_conf.h, [612](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/timing.h, [613](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/include/xyssl/x509.h, [615](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/aes.c, [628](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/arc4.c, [635](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/base64.c, [637](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/bignum.c, [640](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/certs.c, [655](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/des.c, [659](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/dhm.c, [671](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/dsa.c, [675](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/havege.c, [680](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/md2.c, [683](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/md4.c, [688](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/md5.c, [694](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/net.c, [700](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/rsa.c, [704](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/sha1.c, [711](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/sha2.c, [718](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/sha4.c, [725](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/ssl\_cli.c, [732](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/ssl\_srv.c, [735](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/ssl\_tls.c, [739](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/timing.c, [749](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/library/x509read.c, [752](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/aes/aesencrypt2.c, [761](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/hash/hello.c, [763](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/hash/md5sum.c, [764](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/hash/sha1sum.c, [766](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/hash/sha2sum.c, [768](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/dh\_client.c, [770](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/dh\_genprime.c, [772](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/dh\_server.c, [773](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/mpi\_demo.c, [775](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/rsa\_genkey.c, [776](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/rsa\_sign.c, [777](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/pkey/rsa\_verify.c, [778](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/ssl/ssl\_client1.c, [779](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/ssl/ssl\_client2.c, [781](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/ssl/ssl\_server.c, [783](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/test/benchmark.c, [786](#)
- /home/dko/projects/mobilec/trunk/src/security/xyssl-0.7/programs/test/selftest.c, [788](#)
- /home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAcl, [789](#)
- /home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAg, [790](#)
- /home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCAg, [791](#)
- /home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/MCExp, [792](#)
- /home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/Propert, [793](#)
- /home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/Propert, [794](#)
- /home/dko/projects/mobilec/trunk/src/win32/LibMC.net/LibMC/Setting, [795](#)
- /home/dko/projects/mobilec/trunk/src/winconfig.h, [796](#)

- /home/dko/projects/mobilec/trunk/src/xml\_-  
compose.c, [797](#)
- /home/dko/projects/mobilec/trunk/src/xml\_-  
helper.c, [801](#)
- /home/dko/projects/mobilec/trunk/src/xml\_parser.c,  
[803](#)
- \_CRT\_SECURE\_NO\_DEPRECATED
  - aes.c, [629](#)
  - aesencrypt2.c, [761](#)
  - arc4.c, [635](#)
  - base64.c, [637](#)
  - benchmark.c, [786](#)
  - bignum.c, [642](#)
  - des.c, [660](#)
  - dh\_client.c, [770](#)
  - dh\_genprime.c, [772](#)
  - dh\_server.c, [773](#)
  - dhm.c, [671](#)
  - dsa.c, [676](#)
  - havege.c, [680](#)
  - hello.c, [763](#)
  - md2.c, [684](#)
  - md4.c, [689](#)
  - md5.c, [695](#)
  - md5sum.c, [764](#)
  - mpi\_demo.c, [775](#)
  - net.c, [701](#)
  - rsa.c, [705](#)
  - rsa\_genkey.c, [776](#)
  - rsa\_sign.c, [777](#)
  - rsa\_verify.c, [778](#)
  - selftest.c, [788](#)
  - sha1.c, [712](#)
  - sha1sum.c, [766](#)
  - sha2.c, [719](#)
  - sha2sum.c, [768](#)
  - sha4.c, [726](#)
  - ssl\_cli.c, [732](#)
  - ssl\_client1.c, [779](#)
  - ssl\_client2.c, [781](#)
  - ssl\_server.c, [783](#)
  - ssl\_srv.c, [735](#)
  - ssl\_tls.c, [741](#)
  - timing.c, [749](#)
  - x509read.c, [753](#)
- \_MC\_AclAddReceiver
  - LibMC::MCAgency, [129](#)
- \_MC\_AclAddReplyTo
  - LibMC::MCAgency, [129](#)
- \_MC\_AclDestroy
  - LibMC::MCAgency, [129](#)
- \_MC\_AclNew
  - LibMC::MCAgency, [129](#)
- \_MC\_AclPost
  - LibMC::MCAgency, [129](#)
- \_MC\_AclReply
  - LibMC::MCAgency, [129](#)
- \_MC\_AclRetrieve
  - LibMC::MCAgency, [129](#)
- \_MC\_AclSend
  - LibMC::MCAgency, [129](#)
- \_MC\_AclSetContent
  - LibMC::MCAgency, [129](#)
- \_MC\_AclSetPerformative
  - LibMC::MCAgency, [129](#)
- \_MC\_AclSetSender
  - LibMC::MCAgency, [129](#)
- \_MC\_AclWaitRetrieve
  - LibMC::MCAgency, [129](#)
- \_MC\_AddAgent
  - LibMC::MCAgency, [127](#)
- \_MC\_BarrierDelete
  - LibMC::MCAgency, [126](#)
- \_MC\_BarrierInit
  - LibMC::MCAgency, [126](#)
- \_MC\_CallAgentFunc
  - LibMC::MCAgency, [129](#)
- \_MC\_ChInitializeOptions
  - LibMC::MCAgency, [124](#)
- \_MC\_CondBroadcast
  - LibMC::MCAgency, [125](#)
- \_MC\_CondReset
  - LibMC::MCAgency, [125](#)
- \_MC\_CondSignal
  - LibMC::MCAgency, [125](#)
- \_MC\_CondWait
  - LibMC::MCAgency, [125](#)
- \_MC\_DeleteAgent
  - LibMC::MCAgency, [127](#)
- \_MC\_End
  - LibMC::MCAgency, [124](#)
- \_MC\_FindAgentByID
  - LibMC::MCAgency, [127](#)
- \_MC\_FindAgentByName
  - LibMC::MCAgency, [127](#)
- \_MC\_GetAgentExecEngine
  - LibMC::MCAgency, [129](#)
- \_MC\_GetAgentID
  - LibMC::MCAgency, [127](#)
- \_MC\_GetAgentName
  - LibMC::MCAgency, [129](#)
- \_MC\_GetAgentNumTasks
  - LibMC::MCAgency, [129](#)
- \_MC\_GetAgentReturnData
  - LibMC::MCAgency, [129](#)
- \_MC\_GetAgentStatus
  - LibMC::MCAgency, [129](#)
- \_MC\_GetAgentType

- LibMC::MCAgency, 129
- \_MC\_GetAgentXMLString
  - LibMC::MCAgency, 129
- \_MC\_HaltAgency
  - LibMC::MCAgency, 124
- \_MC\_Initialize
  - LibMC::MCAgency, 124
- \_MC\_InitializeAgencyOptions
  - LibMC::MCAgency, 124
- \_MC\_MainLoop
  - LibMC::MCAgency, 125
- \_MC\_MutexLock
  - LibMC::MCAgency, 125
- \_MC\_MutexUnlock
  - LibMC::MCAgency, 125
- \_MC\_PrintAgentCode
  - LibMC::MCAgency, 129
- \_MC\_RegisterService
  - LibMC::MCAgency, 127
- \_MC\_ResetSignal
  - LibMC::MCAgency, 126
- \_MC\_ResumeAgency
  - LibMC::MCAgency, 124
- \_MC\_RetrieveAgent
  - LibMC::MCAgency, 127
- \_MC\_RetrieveAgentCode
  - LibMC::MCAgency, 129
- \_MC\_SearchForService
  - LibMC::MCAgency, 127
- \_MC\_SemaphorePost
  - LibMC::MCAgency, 126
- \_MC\_SemaphoreWait
  - LibMC::MCAgency, 126
- \_MC\_SendAgentMigrationMessage
  - LibMC::MCAgency, 125
- \_MC\_SendAgentMigrationMessageFile
  - LibMC::MCAgency, 125
- \_MC\_SetAgentStatus
  - LibMC::MCAgency, 129
- \_MC\_SetDefaultAgentStatus
  - LibMC::MCAgency, 124
- \_MC\_SetThreadOff
  - LibMC::MCAgency, 124
- \_MC\_SetThreadOn
  - LibMC::MCAgency, 124
- \_MC\_SetThreadsAllOff
  - LibMC::MCAgency, 124
- \_MC\_SetThreadsAllOn
  - LibMC::MCAgency, 124
- \_MC\_Steer
  - LibMC::MCAgency, 126
- \_MC\_SteerControl
  - LibMC::MCAgency, 126
- \_MC\_SyncDelete
  - LibMC::MCAgency, 126
- \_MC\_SyncInit
  - LibMC::MCAgency, 126
- \_MC\_TerminateAgent
  - LibMC::MCAgency, 129
- \_MC\_WaitAgent
  - LibMC::MCAgency, 125
- \_MC\_WaitRetrieveAgent
  - LibMC::MCAgency, 125
- \_MC\_WaitSignal
  - LibMC::MCAgency, 126
- \_XOPEN\_SOURCE
  - xml\_parser.c, 803
- \_aes\_src
  - aes.c, 633
- \_arc4\_src
  - arc4.c, 636
- \_base64\_src
  - base64.c, 638
- \_bignum\_src
  - bignum.c, 654
- \_certs\_src
  - certs.c, 655
- \_des\_src
  - des.c, 666
- \_dhm\_src
  - dhm.c, 674
- \_havege\_src
  - havege.c, 682
- \_hr\_time, 21
  - start, 21
- \_md2\_src
  - md2.c, 687
- \_md4\_src
  - md4.c, 693
- \_md5\_src
  - md5.c, 699
- \_sha1\_src
  - sha1.c, 716
- \_sha2\_src
  - sha2.c, 724
- \_sha4\_src
  - sha4.c, 730
- \_ssl\_cli\_src
  - ssl\_cli.c, 734
- \_ssl\_srv\_src
  - ssl\_srv.c, 738
- \_ssl\_tls\_src
  - ssl\_tls.c, 747
- \_timing\_src
  - timing.c, 750
- \_x509\_buf, 22
  - len, 22
  - p, 22



- tag, 22
- \_x509\_cert, 23
  - ca\_istrue, 25
  - issuer, 24
  - issuer\_id, 25
  - issuer\_raw, 24
  - max\_pathlen, 25
  - next, 25
  - pk\_oid, 24
  - raw, 23
  - rsa, 24
  - serial, 23
  - sig, 25
  - sig\_oid1, 24
  - sig\_oid2, 25
  - subject, 24
  - subject\_id, 25
  - subject\_raw, 24
  - tbs, 23
  - v3\_ext, 25
  - valid\_from, 24
  - valid\_to, 24
  - version, 23
- \_x509\_name, 27
  - next, 27
  - oid, 27
  - val, 27
- \_x509\_read\_src
  - x509read.c, 760
- \_x509\_time, 28
  - day, 28
  - hour, 28
  - min, 28
  - mon, 28
  - sec, 28
  - year, 28
- acc
  - mc\_platform\_s, 93
- acc.c
  - acc\_Destroy, 200
  - acc\_Initialize, 200
  - acc\_MessageHandlerThread, 200
  - acc\_Start, 200
  - acc\_Thread, 201
  - BACKLOG, 200
  - listen\_Thread, 201
- acc.h
  - acc\_Destroy, 246
  - acc\_Initialize, 246
  - acc\_MessageHandlerThread, 246
  - acc\_Start, 247
  - acc\_Thread, 247
  - addr, 248
    - client\_fd, 248
    - listen\_Thread, 247
    - port, 248
    - server\_fd, 248
    - STRUCT, 247
- acc\_Destroy
  - acc.c, 200
  - acc.h, 246
- acc\_Initialize
  - acc.c, 200
  - acc.h, 246
- acc\_MessageHandlerThread
  - acc.c, 200
  - acc.h, 246
- acc\_Start
  - acc.c, 200
  - acc.h, 247
- acc\_Thread
  - acc.c, 201
  - acc.h, 247
- accept\_ranges
  - mtp\_http\_s, 161
- ACCEPTED
  - mtp\_http.h, 378
- acl\_representation
  - fipa\_acl\_Param\_s, 64
- AcIMsg
  - LibMC::MCAclMessage, 103
- aclmsg\_p
  - LibMC::MCAclMessage, 103
- AcIPost
  - LibMC::MCAgent, 143
- AcIRetrieve
  - LibMC::MCAgent, 143
- AcISend
  - LibMC::MCAgency, 123
- AcIWaitRetrieve
  - LibMC::MCAgent, 143
- add\_variable
  - mxmldoc.c, 491
- AddAgent
  - LibMC::MCAgency, 122
- addr
  - acc.h, 248
  - connection\_s, 51
  - message\_s, 154
- AddReceiver
  - LibMC::MCAclMessage, 101
- AddReplyTo
  - LibMC::MCAclMessage, 101
- addresses
  - fipa\_agent\_identifier\_s, 66
- aes.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, 629

- [\\_aes\\_src](#), 633
- [aes\\_cbc\\_decrypt](#), 631
- [aes\\_cbc\\_encrypt](#), 631
- [aes\\_decrypt](#), 632
- [aes\\_encrypt](#), 632
- [AES\\_FROUND](#), 629
- [aes\\_gen\\_tables](#), 632
- [AES\\_RROUND](#), 629
- [aes\\_self\\_test](#), 632
- [aes\\_set\\_key](#), 632
- [FSb](#), 633
- [FT0](#), 633
- [FT1](#), 633
- [FT2](#), 633
- [FT3](#), 633
- [GET\\_UINT32\\_BE](#), 630
- [KT0](#), 633
- [KT1](#), 633
- [KT2](#), 634
- [KT3](#), 634
- [MUL](#), 630
- [PUT\\_UINT32\\_BE](#), 630
- [RCON](#), 634
- [ROTR8](#), 630
- [RSb](#), 634
- [RT0](#), 634
- [RT1](#), 634
- [RT2](#), 634
- [RT3](#), 634
- [uint32](#), 631
- [uint8](#), 631
- [XTIME](#), 631
- [aes.h](#)
  - [aes\\_cbc\\_decrypt](#), 510
  - [aes\\_cbc\\_encrypt](#), 511
  - [aes\\_decrypt](#), 511
  - [aes\\_encrypt](#), 511
  - [aes\\_self\\_test](#), 511
  - [aes\\_set\\_key](#), 512
- [aes\\_cbc\\_decrypt](#)
  - [aes.c](#), 631
  - [aes.h](#), 510
- [aes\\_cbc\\_encrypt](#)
  - [aes.c](#), 631
  - [aes.h](#), 511
- [aes\\_context](#), 30
  - [drk](#), 30
  - [erk](#), 30
  - [nr](#), 30
- [aes\\_decrypt](#)
  - [aes.c](#), 632
  - [aes.h](#), 511
- [aes\\_encrypt](#)
  - [aes.c](#), 632
  - [aes.h](#), 511
- [AES\\_FROUND](#)
  - [aes.c](#), 629
- [aes\\_gen\\_tables](#)
  - [aes.c](#), 632
- [AES\\_RROUND](#)
  - [aes.c](#), 629
- [aes\\_self\\_test](#)
  - [aes.c](#), 632
  - [aes.h](#), 511
- [aes\\_set\\_key](#)
  - [aes.c](#), 632
  - [aes.h](#), 512
- [aescript2.c](#)
  - [\\_CRT\\_SECURE\\_NO\\_DEPRECATED](#), 761
  - [main](#), 762
  - [MODE\\_DECRYPT](#), 761
  - [MODE\\_ENCRYPT](#), 761
  - [scanf\\_argv](#), 762
  - [USAGE](#), 761
- [Agency](#)
  - [LibMC::MCAgency](#), 130
- [agency\\_p](#)
  - [libmc.h](#), 327
  - [LibMC::MCAgency](#), 129
- [agency\\_s](#), 31
  - [client](#), 31
  - [default\\_agentstatus](#), 32
  - [enable\\_security](#), 32
  - [filename](#), 31
  - [hostName](#), 31
  - [last\\_error](#), 33
  - [mc\\_platform](#), 32
  - [portno](#), 31
  - [portnoc](#), 32
  - [server](#), 31
  - [stack\\_size](#), 32
  - [threads](#), 32
- [agency\\_t](#)
  - [libmc.h](#), 327
- [Agent](#)
  - [LibMC::MCAgent](#), 146
- [agent.c](#)
  - [agent\\_AddPersistentVariable](#), 202
  - [agent\\_Copy](#), 202
  - [agent\\_Destroy](#), 202
  - [agent\\_Initialize](#), 203
  - [agent\\_New](#), 203
  - [agent\\_RunChScript](#), 203
  - [agent\\_RunChScriptThread](#), 203
- [agent.h](#)
  - [agent\\_AddPersistentVariable](#), 249
  - [agent\\_Copy](#), 249
  - [agent\\_Destroy](#), 249

- agent\_Initialize, 250
- agent\_New, 250
- agent\_RunChScript, 250
- agent\_RunChScriptThread, 250
- AGENT\_UPDATE
  - message.h, 374
- agent\_AddPersistentVariable
  - agent.c, 202
  - agent.h, 249
- agent\_code
  - agent\_datastate\_s, 34
- agent\_code\_ids
  - agent\_datastate\_s, 34
- agent\_codes
  - agent\_datastate\_s, 34
- agent\_Copy
  - agent.c, 202
  - agent.h, 249
- agent\_datastate.c
  - agent\_datastate\_Copy, 205
  - agent\_datastate\_Destroy, 205
  - agent\_datastate\_New, 205
- agent\_datastate.h
  - agent\_datastate\_Copy, 252
  - agent\_datastate\_Destroy, 252
  - agent\_datastate\_New, 253
  - agent\_datastate\_p, 252
  - agent\_datastate\_t, 252
- agent\_datastate\_Copy
  - agent\_datastate.c, 205
  - agent\_datastate.h, 252
- agent\_datastate\_Destroy
  - agent\_datastate.c, 205
  - agent\_datastate.h, 252
- agent\_datastate\_New
  - agent\_datastate.c, 205
  - agent\_datastate.h, 253
- agent\_datastate\_p
  - agent\_datastate.h, 252
- agent\_datastate\_s, 34
  - agent\_code, 34
  - agent\_code\_ids, 34
  - agent\_codes, 34
  - init\_agent\_status, 35
  - number\_of\_tasks, 35
  - persistent, 35
  - return\_data, 35
  - task\_progress, 35
  - tasks, 34
  - xml\_agent\_root, 35
  - xml\_root, 35
- agent\_datastate\_t
  - agent\_datastate.h, 252
- agent\_Destroy
  - agent.c, 202
  - agent.h, 249
- agent\_Initialize
  - agent.c, 203
  - agent.h, 250
- agent\_interp
  - agent\_s, 40
- agent\_lib.h
  - MC\_AclAddReceiver\_chdl, 255
  - MC\_AclAddReplyTo\_chdl, 255
  - MC\_AclDestroy\_chdl, 255
  - MC\_AclNew\_chdl, 255
  - MC\_AclPost\_chdl, 255
  - MC\_AclReply\_chdl, 255
  - MC\_AclRetrieve\_chdl, 256
  - MC\_AclSend\_chdl, 256
  - MC\_AclSetContent\_chdl, 256
  - MC\_AclSetPerformative\_chdl, 256
  - MC\_AclSetSender\_chdl, 256
  - MC\_AclWaitRetrieve\_chdl, 256
  - MC\_AddAgent\_chdl, 256
  - MC\_AgentVariableRetrieve\_chdl, 257
  - MC\_AgentVariableSave\_chdl, 257
  - MC\_Barrier\_chdl, 257
  - MC\_BarrierDelete\_chdl, 257
  - MC\_BarrierInit\_chdl, 257
  - MC\_CallAgentFunc\_chdl, 257
  - MC\_ComposeAgent\_chdl, 257
  - MC\_CondBroadcast\_chdl, 258
  - MC\_CondReset\_chdl, 258
  - MC\_CondSignal\_chdl, 258
  - MC\_CondWait\_chdl, 258
  - MC\_DeleteAgent\_chdl, 258
  - MC\_DeregisterService\_chdl, 258
  - MC\_DestroyServiceSearchResult\_chdl, 258
  - MC\_End\_chdl, 259
  - MC\_FindAgentByID\_chdl, 259
  - MC\_FindAgentByName\_chdl, 259
  - MC\_GetAgentID\_chdl, 259
  - MC\_GetAgentName\_chdl, 259
  - MC\_GetAgentNumTasks\_chdl, 259
  - MC\_GetAgentStatus\_chdl, 259
  - MC\_GetAgentXMLString\_chdl, 259
  - MC\_GetTimeOfDay\_chdl, 260
  - MC\_HaltAgency\_chdl, 260
  - MC\_MutexLock\_chdl, 260
  - MC\_MutexUnlock\_chdl, 260
  - MC\_PrintAgentCode\_chdl, 260
  - MC\_RegisterService\_chdl, 260
  - MC\_ResumeAgency\_chdl, 260
  - MC\_RetrieveAgent\_chdl, 260
  - MC\_RetrieveAgentCode\_chdl, 261
  - MC\_SearchForService\_chdl, 261
  - MC\_SemaphorePost\_chdl, 261

- MC\_SemaphoreWait\_chdl, 261
- MC\_SendAgentMigrationMessage\_chdl, 261
- MC\_SendAgentMigrationMessageFile\_chdl, 261
- MC\_SendSteerCommand\_chdl, 261
- MC\_SetAgentStatus\_chdl, 261
- MC\_SetDefaultAgentStatus\_chdl, 261
- MC\_SyncDelete\_chdl, 262
- MC\_SyncInit\_chdl, 262
- MC\_TerminateAgent\_chdl, 262
- agent\_mailbox.c
  - agent\_mailbox\_Copy, 206
  - agent\_mailbox\_Destroy, 206
  - agent\_mailbox\_New, 206
  - agent\_mailbox\_Post, 206
  - agent\_mailbox\_Retrieve, 206
  - agent\_mailbox\_WaitRetrieve, 206
- agent\_mailbox.h
  - agent\_mailbox\_Copy, 263
  - agent\_mailbox\_Destroy, 263
  - agent\_mailbox\_New, 263
  - agent\_mailbox\_p, 263
  - agent\_mailbox\_Post, 264
  - agent\_mailbox\_Retrieve, 264
  - agent\_mailbox\_t, 263
  - agent\_mailbox\_WaitRetrieve, 264
- agent\_mailbox\_Copy
  - agent\_mailbox.c, 206
  - agent\_mailbox.h, 263
- agent\_mailbox\_Destroy
  - agent\_mailbox.c, 206
  - agent\_mailbox.h, 263
- agent\_mailbox\_New
  - agent\_mailbox.c, 206
  - agent\_mailbox.h, 263
- agent\_mailbox\_p
  - agent\_mailbox.h, 263
  - data\_structures.h, 285
- agent\_mailbox\_Post
  - agent\_mailbox.c, 206
  - agent\_mailbox.h, 264
- agent\_mailbox\_Retrieve
  - agent\_mailbox.c, 206
  - agent\_mailbox.h, 264
- agent\_mailbox\_s, 37
  - mail\_queue, 37
- agent\_mailbox\_t
  - agent\_mailbox.h, 263
- agent\_mailbox\_WaitRetrieve
  - agent\_mailbox.c, 206
  - agent\_mailbox.h, 264
- agent\_New
  - agent.c, 203
  - agent.h, 250
- agent\_p
  - libmc.h, 327
  - LibMC::MCAgent, 145
- agent\_persistent
  - agent\_s, 41
- agent\_pipe\_active
  - agent\_s, 41
- agent\_pipe\_ready\_to\_read
  - agent\_s, 41
- agent\_queue
  - mc\_platform\_s, 92
- agent\_ready\_to\_send
  - agent\_s, 41
- agent\_return\_data
  - agent\_task\_s, 44
- agent\_return\_data.c
  - interpreter\_variable\_data\_Copy, 208
  - interpreter\_variable\_data\_Destroy, 208
  - interpreter\_variable\_data\_Initialize, 208
  - interpreter\_variable\_data\_InitializeFromAgent, 208
  - interpreter\_variable\_data\_New, 209
- agent\_return\_xml\_parse
  - xml\_parser.c, 804
- agent\_RunChScript
  - agent.c, 203
  - agent.h, 250
- agent\_RunChScriptThread
  - agent.c, 203
  - agent.h, 250
- agent\_s, 38
  - agent\_interp, 40
  - agent\_persistent, 41
  - agent\_pipe\_active, 41
  - agent\_pipe\_ready\_to\_read, 41
  - agent\_ready\_to\_send, 41
  - agent\_script\_ready, 41
  - agent\_status, 40
  - agent\_thread, 40
  - agent\_thread\_id, 40
  - agent\_type, 40
  - arrival\_time, 39
  - connect\_id, 39
  - datastate, 39
  - home, 39
  - home\_port, 39
  - id, 38
  - lock, 41
  - mailbox, 41
  - mc\_platform, 41
  - name, 38
  - orphan, 39
  - owner, 39
  - return\_data, 40

- run\_lock, 40
- agent\_script\_ready
  - agent\_s, 41
- agent\_status
  - agent\_s, 40
- agent\_t
  - libmc.h, 327
- agent\_task.c
  - agent\_task\_Copy, 210
  - agent\_task\_Destroy, 210
  - agent\_task\_New, 210
- agent\_task.h
  - agent\_task\_Copy, 265
  - agent\_task\_Destroy, 265
  - agent\_task\_New, 265
  - agent\_task\_p, 265
  - agent\_task\_t, 265
- agent\_task\_Copy
  - agent\_task.c, 210
  - agent\_task.h, 265
- agent\_task\_Destroy
  - agent\_task.c, 210
  - agent\_task.h, 265
- agent\_task\_New
  - agent\_task.c, 210
  - agent\_task.h, 265
- agent\_task\_p
  - agent\_task.h, 265
- agent\_task\_s, 43
  - agent\_return\_data, 44
  - agent\_variable\_list, 44
  - code\_id, 44
  - init\_agent\_status, 43
  - num\_saved\_variables, 44
  - number\_of\_elements, 43
  - persistent, 43
  - saved\_variables, 44
  - server\_name, 44
  - size\_of\_element\_array, 43
  - task\_completed, 43
  - var\_name, 44
- agent\_task\_t
  - agent\_task.h, 265
- agent\_thread
  - agent\_s, 40
- agent\_thread\_id
  - agent\_s, 40
- agent\_type
  - agent\_s, 40
- agent\_variable\_list
  - agent\_task\_s, 44
- agent\_xml\_compose
  - xml\_compose.c, 797
  - xml\_compose.h, 382
- agent\_xml\_compose\_\_agent\_code
  - xml\_compose.c, 797
  - xml\_compose.h, 382
- agent\_xml\_compose\_\_agent\_data
  - xml\_compose.c, 797
  - xml\_compose.h, 382
- agent\_xml\_compose\_\_create\_row\_nodes
  - xml\_compose.c, 798
  - xml\_compose.h, 382
- agent\_xml\_compose\_\_data
  - xml\_compose.c, 798
  - xml\_compose.h, 383
- agent\_xml\_compose\_\_gaf\_message
  - xml\_compose.c, 798
  - xml\_compose.h, 383
- agent\_xml\_compose\_\_home
  - xml\_compose.c, 798
  - xml\_compose.h, 383
- agent\_xml\_compose\_\_message
  - xml\_compose.c, 798
  - xml\_compose.h, 383
- agent\_xml\_compose\_\_mobile\_agent
  - xml\_compose.c, 799
  - xml\_compose.h, 383
- agent\_xml\_compose\_\_name
  - xml\_compose.c, 799
  - xml\_compose.h, 384
- agent\_xml\_compose\_\_owner
  - xml\_compose.c, 799
  - xml\_compose.h, 384
- agent\_xml\_compose\_\_row
  - xml\_compose.c, 799
  - xml\_compose.h, 384
- agent\_xml\_compose\_\_task
  - xml\_compose.c, 799
  - xml\_compose.h, 384
- agent\_xml\_compose\_\_tasks
  - xml\_compose.c, 799
  - xml\_compose.h, 384
- agent\_xml\_flag
  - message\_s, 156
- agent\_xml\_parse
  - xml\_parser.c, 804
  - xml\_parser.h, 387
- agent\_xml\_parse\_\_agent\_code
  - xml\_parser.c, 804
  - xml\_parser.h, 387
- agent\_xml\_parse\_\_agent\_data
  - xml\_parser.c, 804
  - xml\_parser.h, 387
- agent\_xml\_parse\_\_data
  - xml\_parser.c, 804
  - xml\_parser.h, 388
- agent\_xml\_parse\_\_fill\_row\_data

- xml\_parser.c, 804
- xml\_parser.h, 388
- agent\_xml\_parse\_\_home
  - xml\_parser.c, 805
  - xml\_parser.h, 388
- agent\_xml\_parse\_\_mobile\_agent
  - xml\_parser.c, 805
  - xml\_parser.h, 388
- agent\_xml\_parse\_\_name
  - xml\_parser.c, 805
  - xml\_parser.h, 388
- agent\_xml\_parse\_\_owner
  - xml\_parser.c, 805
  - xml\_parser.h, 389
- agent\_xml\_parse\_\_row
  - xml\_parser.c, 805
  - xml\_parser.h, 389
- agent\_xml\_parse\_\_task
  - xml\_parser.c, 805
  - xml\_parser.h, 389
- agent\_xml\_parse\_\_tasks
  - xml\_parser.c, 806
  - xml\_parser.h, 389
- AgentID
  - LibMC::MCAgent, 146
- AgentName
  - LibMC::MCAgent, 146
- AgentNumTasks
  - LibMC::MCAgent, 147
- AgentStatus
  - LibMC::MCAgent, 147
- AgentType
  - LibMC::MCAgent, 147
- alarmed
  - timing.c, 750
  - timing.h, 614
- alloc\_nodes
  - mxml\_index\_s, 166
- ams
  - mc\_platform\_s, 92
- ams.c
  - ams\_Destroy, 211
  - ams\_Initialize, 211
  - ams\_ManageAgentList, 211
  - ams\_Print, 211
  - ams\_Start, 211
  - ams\_Thread, 212
- ams.h
  - ams\_Destroy, 267
  - ams\_ManageAgentList, 267
  - ams\_Print, 267
  - ams\_RemoveFinishedAgents, 267
  - ams\_Start, 267
  - ams\_Thread, 267
- STRUCT, 268
- ams\_Destroy
  - ams.c, 211
  - ams.h, 267
- ams\_Initialize
  - ams.c, 211
- ams\_ManageAgentList
  - ams.c, 211
  - ams.h, 267
- ams\_Print
  - ams.c, 211
  - ams.h, 267
- ams\_RemoveFinishedAgents
  - ams.h, 267
- ams\_Start
  - ams.c, 211
  - ams.h, 267
- ams\_Thread
  - ams.c, 212
  - ams.h, 267
- AP\_QUEUE\_DECL\_TEMPLATE
  - ap\_queue\_template.h, 269
  - data\_structures.h, 284
- AP\_QUEUE\_GENERIC\_DECL\_TEMPLATE
  - ap\_queue\_template.h, 269
  - data\_structures.h, 284
- AP\_QUEUE\_REMOVE\_TEMPLATE
  - ap\_queue\_template.h, 269
- AP\_QUEUE\_SEARCH\_TEMPLATE
  - ap\_queue\_template.h, 270
  - data\_structures.c, 219
- AP\_QUEUE\_STD\_DEFN\_TEMPLATE
  - ap\_queue\_template.h, 270
  - data\_structures.c, 219
- ap\_queue\_template.h
  - AP\_QUEUE\_DECL\_TEMPLATE, 269
  - AP\_QUEUE\_GENERIC\_DECL\_TEMPLATE, 269
  - AP\_QUEUE\_REMOVE\_TEMPLATE, 269
  - AP\_QUEUE\_SEARCH\_TEMPLATE, 270
  - AP\_QUEUE\_STD\_DEFN\_TEMPLATE, 270
- arc4.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, 635
  - \_arc4\_src, 636
  - arc4\_crypt, 635
  - arc4\_self\_test, 635
  - arc4\_setup, 636
- arc4.h
  - arc4\_crypt, 513
  - arc4\_self\_test, 513
  - arc4\_setup, 513
- arc4\_context, 46
  - m, 46
  - x, 46

- y, [46](#)
- arc4\_crypt
  - arc4.c, [635](#)
  - arc4.h, [513](#)
- arc4\_self\_test
  - arc4.c, [635](#)
  - arc4.h, [513](#)
- arc4\_setup
  - arc4.c, [636](#)
  - arc4.h, [513](#)
- args
  - command\_s, [50](#)
- array\_dim
  - interpreter\_variable\_data\_s, [87](#)
- array\_extent
  - interpreter\_variable\_data\_s, [87](#)
- arrival\_time
  - agent\_s, [39](#)
- asm\_message\_queue
  - mc\_platform\_s, [92](#)
- ASN1\_BIT\_STRING
  - x509.h, [617](#)
- ASN1\_BMP\_STRING
  - x509.h, [617](#)
- ASN1\_BOOLEAN
  - x509.h, [617](#)
- ASN1\_CONSTRUCTED
  - x509.h, [617](#)
- ASN1\_CONTEXT\_SPECIFIC
  - x509.h, [617](#)
- asn1\_get\_bool
  - x509read.c, [753](#)
- asn1\_get\_int
  - x509read.c, [753](#)
- asn1\_get\_len
  - x509read.c, [754](#)
- asn1\_get\_mpi
  - x509read.c, [754](#)
- asn1\_get\_tag
  - x509read.c, [754](#)
- ASN1\_HASH\_MDX
  - rsa.h, [569](#)
- ASN1\_HASH\_SHA1
  - rsa.h, [569](#)
- ASN1\_IA5\_STRING
  - x509.h, [618](#)
- ASN1\_INTEGER
  - x509.h, [618](#)
- ASN1\_NULL
  - x509.h, [618](#)
- ASN1\_OCTET\_STRING
  - x509.h, [618](#)
- ASN1\_OID
  - x509.h, [618](#)
- ASN1\_PRIMITIVE
  - x509.h, [618](#)
- ASN1\_PRINTABLE\_STRING
  - x509.h, [618](#)
- ASN1\_SEQUENCE
  - x509.h, [618](#)
- ASN1\_SET
  - x509.h, [619](#)
- ASN1\_T61\_STRING
  - x509.h, [619](#)
- ASN1\_UNIVERSAL\_STRING
  - x509.h, [619](#)
- ASN1\_UTC\_TIME
  - x509.h, [619](#)
- ASN1\_UTF8\_STRING
  - x509.h, [619](#)
- attr
  - mxml\_index\_s, [166](#)
- attrs
  - mxml\_value\_s, [171](#)
- authmode
  - ssl\_context, [190](#)
- BACKLOG
  - acc.c, [200](#)
- BAD\_REQUEST
  - mtp\_http.h, [378](#)
- BADCERT\_CN\_MISMATCH
  - x509.h, [619](#)
- BADCERT\_HAS\_EXPIRED
  - x509.h, [619](#)
- BADCERT\_NOT\_TRUSTED
  - x509.h, [619](#)
- bar
  - class.cxx, [494](#)
  - foo\_c, [79](#)
  - foo\_s, [82](#)
- barfoo
  - foo\_c, [80](#)
- barrier.c
  - barrier\_node\_Destroy, [213](#)
  - barrier\_node\_Initialize, [213](#)
  - barrier\_queue\_Add, [213](#)
  - barrier\_queue\_Delete, [213](#)
  - barrier\_queue\_Destroy, [214](#)
  - barrier\_queue\_Get, [214](#)
  - barrier\_queue\_New, [214](#)
  - barrier\_queue\_Pop, [214](#)
- barrier.h
  - barrier\_node\_Destroy, [273](#)
  - barrier\_node\_Initialize, [273](#)
  - barrier\_node\_p, [272](#)
  - barrier\_node\_t, [272](#)
  - barrier\_queue\_Add, [273](#)

- barrier\_queue\_Delete, [273](#)
- barrier\_queue\_Destroy, [273](#)
- barrier\_queue\_Get, [273](#)
- barrier\_queue\_New, [274](#)
- barrier\_queue\_p, [272](#)
- barrier\_queue\_Pop, [274](#)
- barrier\_queue\_t, [272](#)
- barrier\_node\_Destroy
  - barrier.c, [213](#)
  - barrier.h, [273](#)
- barrier\_node\_Initialize
  - barrier.c, [213](#)
  - barrier.h, [273](#)
- barrier\_node\_p
  - barrier.h, [272](#)
- barrier\_node\_s, [47](#)
  - cond, [47](#)
  - id, [47](#)
  - lock, [47](#)
  - num\_registered, [47](#)
  - num\_waiting, [47](#)
- barrier\_node\_t
  - barrier.h, [272](#)
- barrier\_queue
  - mc\_platform\_s, [93](#)
- barrier\_queue\_Add
  - barrier.c, [213](#)
  - barrier.h, [273](#)
- barrier\_queue\_Delete
  - barrier.c, [213](#)
  - barrier.h, [273](#)
- barrier\_queue\_Destroy
  - barrier.c, [214](#)
  - barrier.h, [273](#)
- barrier\_queue\_Get
  - barrier.c, [214](#)
  - barrier.h, [273](#)
- barrier\_queue\_New
  - barrier.c, [214](#)
  - barrier.h, [274](#)
- barrier\_queue\_p
  - barrier.h, [272](#)
- barrier\_queue\_Pop
  - barrier.c, [214](#)
  - barrier.h, [274](#)
- barrier\_queue\_s, [48](#)
  - list, [48](#)
  - lock, [48](#)
  - size, [48](#)
- barrier\_queue\_t
  - barrier.h, [272](#)
- BarrierDelete
  - LibMC::MCAgency, [119](#)
- BarrierInit
  - LibMC::MCAgency, [120](#)
- base64.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, [637](#)
  - \_base64\_src, [638](#)
  - base64\_dec\_map, [638](#)
  - base64\_decode, [637](#)
  - base64\_enc\_map, [639](#)
  - base64\_encode, [638](#)
  - base64\_self\_test, [638](#)
- base64.h
  - base64\_decode, [515](#)
  - base64\_encode, [516](#)
  - base64\_self\_test, [516](#)
  - ERR\_BASE64\_BUFFER\_TOO\_SMALL, [515](#)
  - ERR\_BASE64\_INVALID\_CHARACTER, [515](#)
- base64\_dec\_map
  - base64.c, [638](#)
- base64\_decode
  - base64.c, [637](#)
  - base64.h, [515](#)
- base64\_enc\_map
  - base64.c, [639](#)
- base64\_encode
  - base64.c, [638](#)
  - base64.h, [516](#)
- base64\_self\_test
  - base64.c, [638](#)
  - base64.h, [516](#)
- benchmark.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, [786](#)
  - BUFSIZE, [786](#)
  - main, [786](#)
  - myrand, [787](#)
- bignum.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, [642](#)
  - \_bignum\_src, [654](#)
  - BITS\_TO\_LIMBS, [642](#)
  - CHARS\_TO\_LIMBS, [643](#)
  - mpi\_add\_abs, [643](#)
  - mpi\_add\_int, [643](#)
  - mpi\_add\_mpi, [643](#)
  - mpi\_cmp\_abs, [643](#)
  - mpi\_cmp\_int, [644](#)
  - mpi\_cmp\_mpi, [644](#)
  - mpi\_copy, [644](#)
  - mpi\_div\_int, [644](#)
  - mpi\_div\_mpi, [645](#)
  - mpi\_exp\_mod, [645](#)
  - mpi\_free, [645](#)
  - mpi\_gcd, [646](#)
  - mpi\_gen\_prime, [646](#)
  - mpi\_get\_digit, [646](#)
  - mpi\_grow, [647](#)



- mpi\_init, 647
- mpi\_inv\_mod, 647
- mpi\_is\_prime, 647
- mpi\_lsb, 648
- mpi\_lset, 648
- mpi\_mod\_int, 648
- mpi\_mod\_mpi, 648
- mpi\_montg\_init, 648
- mpi\_montmul, 649
- mpi\_montred, 649
- mpi\_msb, 649
- mpi\_mul\_hlp, 649
- mpi\_mul\_int, 649
- mpi\_mul\_mpi, 649
- mpi\_read\_binary, 650
- mpi\_read\_file, 650
- mpi\_read\_string, 650
- mpi\_self\_test, 651
- mpi\_shift\_l, 651
- mpi\_shift\_r, 651
- mpi\_sub\_abs, 652
- mpi\_sub\_hlp, 652
- mpi\_sub\_int, 652
- mpi\_sub\_mpi, 652
- mpi\_swap, 652
- mpi\_write\_binary, 653
- mpi\_write\_file, 653
- mpi\_write\_hlp, 653
- mpi\_write\_string, 654
- small\_prime, 654
- bignum.h
  - biH, 520
  - biL, 520
  - CHK, 520
  - ciL, 520
  - ERR\_MPI\_BUFFER\_TOO\_SMALL, 520
  - ERR\_MPI\_DIVISION\_BY\_ZERO, 520
  - ERR\_MPI\_FILE\_IO\_ERROR, 520
  - ERR\_MPI\_INVALID\_CHARACTER, 520
  - ERR\_MPI\_INVALID\_PARAMETER, 521
  - ERR\_MPI\_NEGATIVE\_VALUE, 521
  - ERR\_MPI\_NOT\_ACCEPTABLE, 521
  - mpi\_add\_abs, 521
  - mpi\_add\_int, 521
  - mpi\_add\_mpi, 522
  - mpi\_cmp\_abs, 522
  - mpi\_cmp\_int, 522
  - mpi\_cmp\_mpi, 522
  - mpi\_copy, 523
  - mpi\_div\_int, 523
  - mpi\_div\_mpi, 523
  - mpi\_exp\_mod, 524
  - mpi\_free, 524
  - mpi\_gcd, 524
  - mpi\_gen\_prime, 525
  - mpi\_grow, 525
  - mpi\_init, 525
  - mpi\_inv\_mod, 525
  - mpi\_is\_prime, 526
  - mpi\_lsb, 526
  - mpi\_lset, 526
  - mpi\_mod\_int, 526
  - mpi\_mod\_mpi, 527
  - mpi\_msb, 527
  - mpi\_mul\_int, 527
  - mpi\_mul\_mpi, 527
  - mpi\_read\_binary, 528
  - mpi\_read\_file, 528
  - mpi\_read\_string, 528
  - mpi\_self\_test, 529
  - mpi\_shift\_l, 529
  - mpi\_shift\_r, 529
  - mpi\_sub\_abs, 530
  - mpi\_sub\_int, 530
  - mpi\_sub\_mpi, 530
  - mpi\_swap, 530
  - mpi\_write\_binary, 530
  - mpi\_write\_file, 531
  - mpi\_write\_string, 531
  - t\_dbl, 521
  - t\_int, 521
- biH
  - bignum.h, 520
- biL
  - bignum.h, 520
- BITS\_TO\_LIMBS
  - bignum.c, 642
- bn\_asm.h
  - MULADDC\_CORE, 533
  - MULADDC\_INIT, 533
  - MULADDC\_STOP, 533
- boundary
  - mtp\_http\_s, 162
- buffer
  - md2\_context, 148
  - md4\_context, 150
  - md5\_context, 152
  - mxml\_fdbuf\_s, 165
  - sha1\_context, 179
  - sha2\_context, 181
  - sha4\_context, 183
- BUFSIZE
  - benchmark.c, 786
- ca\_chain
  - ssl\_context, 190
- ca\_istrue
  - \_x509\_cert, 25

- cache\_control
  - mtp\_http\_s, [162](#)
- CallAgentFunc
  - LibMC::MCAgency, [144](#)
- callbacks
  - mxml-entity.c, [459](#)
- CANCEL
  - message.h, [374](#)
- certs.c
  - \_certs\_src, [655](#)
  - test\_ca\_cert, [655](#)
  - test\_ca\_key, [655](#)
  - test\_ca\_pwd, [655](#)
  - test\_cli\_cert, [656](#)
  - test\_cli\_key, [656](#)
  - test\_srv\_cert, [657](#)
  - test\_srv\_key, [657](#)
  - xyssl\_ca\_cert, [658](#)
- certs.h
  - test\_ca\_cert, [535](#)
  - test\_ca\_key, [535](#)
  - test\_ca\_pwd, [535](#)
  - test\_cli\_cert, [535](#)
  - test\_cli\_key, [535](#)
  - test\_srv\_cert, [535](#)
  - test\_srv\_key, [536](#)
  - xyssl\_ca\_cert, [536](#)
- CH\_REGULARCH
  - LibMC::MCAgency, [109](#)
- CH\_SAFECH
  - LibMC::MCAgency, [109](#)
- CH\_DATATYPE\_SIZE
  - macros.h, [357](#)
- CH\_DATATYPE\_STR\_TO\_VAL
  - macros.h, [357](#)
- CH\_DATATYPE\_STRING
  - macros.h, [358](#)
- CH\_DATATYPE\_VALUE\_STRING
  - macros.h, [358](#)
- CH\_STRING\_DATATYPE
  - macros.h, [359](#)
- CHARS\_TO\_LIMBS
  - bignum.c, [643](#)
- CHECK\_NULL
  - macros.h, [359](#)
- chhome
  - LibMC::MCAgency::ChOptions\_t, [132](#)
- child
  - mxml\_node\_s, [168](#)
- ChInitializeOptions
  - LibMC::MCAgency, [111](#)
- CHK
  - bignum.h, [520](#)
- ChShellType
  - LibMC::MCAgency, [109](#)
- ciL
  - bignum.h, [520](#)
- cipher
  - ssl\_context, [193](#)
- cipherlist
  - ssl\_context, [193](#)
- cksum
  - md2\_context, [148](#)
- class.cxx
  - bar, [494](#)
  - f, [494](#)
  - foo, [494](#)
- client
  - agency\_s, [31](#)
  - LibMC::MCAgency::MCAgency\_t, [133](#)
- client\_auth
  - ssl\_context, [190](#)
- client\_fd
  - acc.h, [248](#)
- client\_thread\_arg
  - mc\_platform\_s, [93](#)
- clientfd
  - connection\_s, [51](#)
- cmd\_handler\_t
  - commands.h, [278](#)
- cmd\_handlers
  - commands.h, [278](#)
- cmd\_prompt
  - mc\_platform\_s, [93](#)
- cmd\_prompt.c
  - cmd\_prompt\_Destroy, [215](#)
  - cmd\_prompt\_Initialize, [215](#)
  - cmd\_prompt\_Start, [215](#)
  - cmd\_prompt\_Thread, [216](#)
  - dealloc\_command, [216](#)
  - exec\_command, [216](#)
  - handler\_HELP, [216](#)
  - handler\_PRINT\_CONNECTLIST, [216](#)
  - handler\_PRINTLIST\_AGENTS, [216](#)
  - handler\_PRINTLIST\_MESSAGE, [216](#)
  - handler\_QUIT, [216](#)
  - handler\_SEND, [216](#)
  - process\_command, [217](#)
  - split\_string, [217](#)
- cmd\_prompt.h
  - cmd\_prompt\_Destroy, [276](#)
  - cmd\_prompt\_Initialize, [276](#)
  - cmd\_prompt\_p, [275](#)
  - cmd\_prompt\_Start, [276](#)
  - cmd\_prompt\_t, [275](#)
  - cmd\_prompt\_Thread, [276](#)
  - CMDLINE\_SIZE, [275](#)
  - command\_p, [276](#)

- command\_t, 276
- dealloc\_command, 276
- exec\_command, 276
- process\_command, 276
- split\_string, 276
- cmd\_prompt\_Destroy
  - cmd\_prompt.c, 215
  - cmd\_prompt.h, 276
- cmd\_prompt\_Initialize
  - cmd\_prompt.c, 215
  - cmd\_prompt.h, 276
- cmd\_prompt\_p
  - cmd\_prompt.h, 275
- cmd\_prompt\_s, 49
  - thread, 49
- cmd\_prompt\_Start
  - cmd\_prompt.c, 215
  - cmd\_prompt.h, 276
- cmd\_prompt\_t
  - cmd\_prompt.h, 275
- cmd\_prompt\_Thread
  - cmd\_prompt.c, 216
  - cmd\_prompt.h, 276
- CMDLINE\_SIZE
  - cmd\_prompt.h, 275
- code\_id
  - agent\_task\_s, 44
- COLLECT\_SIZE
  - havege.h, 547
- COMMAND
  - commands.h, 277, 278
  - commands.x.h, 279
- command\_cmds
  - commands.h, 278
- command\_descriptions
  - commands.h, 278
- command\_indices\_e
  - commands.h, 278
- command\_p
  - cmd\_prompt.h, 276
- command\_s, 50
  - args, 50
  - index, 50
  - num\_args, 50
- command\_t
  - cmd\_prompt.h, 276
- commands.h
  - cmd\_handler\_t, 278
  - cmd\_handlers, 278
  - COMMAND, 277, 278
  - command\_cmds, 278
  - command\_descriptions, 278
  - command\_indices\_e, 278
- commands.x.h
  - COMMAND, 279
  - help, 279
  - is, 279
  - print\_connectlist, 279
  - printlist\_agents, 279
  - send, 280
- comments
  - fipa\_acl\_Param\_s, 64
- COMPOSE\_BLOCKSIZE
  - dynstring.h, 293
- cond
  - barrier\_node\_s, 47
  - df.h, 290
  - mc\_rwlock\_s, 96
  - syncListNode\_s, 197
- COND\_BROADCAST
  - macros.h, 360
- COND\_DESTROY
  - macros.h, 360
- COND\_INIT
  - macros.h, 360
- COND\_RESET
  - macros.h, 360
- COND\_SIGNAL
  - macros.h, 360
- COND\_SLEEP
  - macros.h, 361
- COND\_SLEEP\_ACTION
  - macros.h, 361
- COND\_T
  - macros.h, 361
- COND\_WAIT
  - macros.h, 361
- CondBroadcast
  - LibMC::MCAgency, 116
- CondReset
  - LibMC::MCAgency, 116
- CondSignal
  - LibMC::MCAgency, 116
- CondWait
  - LibMC::MCAgency, 116
- config.h
  - HAVE\_STRDUP, 453
  - HAVE\_VSNPRINTF, 453
  - mxml\_strdupf, 453
  - MXML\_VERSION, 453
- CONFLICT
  - mtp\_http.h, 378
- connect\_id
  - agent\_s, 39
  - connection\_s, 51
  - message\_s, 154
- connection
  - mtp\_http\_s, 161

- connection.c
  - connection\_Copy, 218
  - connection\_Destroy, 218
  - connection\_New, 218
- connection.h
  - connection\_Close, 281
  - connection\_Copy, 281
  - connection\_CreateMessage, 281
  - connection\_Destroy, 282
  - connection\_New, 282
  - connection\_p, 281
  - connection\_RecvFromConnection, 282
  - connection\_SendtoConnection, 282
  - connection\_SetHosts, 282
  - connection\_t, 281
- connection\_Close
  - connection.h, 281
- connection\_Copy
  - connection.c, 218
  - connection.h, 281
- connection\_CreateMessage
  - connection.h, 281
- connection\_Destroy
  - connection.c, 218
  - connection.h, 282
- connection\_New
  - connection.c, 218
  - connection.h, 282
- connection\_p
  - connection.h, 281
- connection\_queue
  - mc\_platform\_s, 92
- connection\_RecvFromConnection
  - connection.h, 282
- connection\_s, 51
  - addr, 51
  - clientfd, 51
  - connect\_id, 51
  - remote\_hostname, 51
  - serverfd, 51
- connection\_SendtoConnection
  - connection.h, 282
- connection\_SetHosts
  - connection.h, 282
- connection\_t
  - connection.h, 281
- content
  - fipa\_acl\_message\_s, 62
  - fipa\_expression\_s, 70
  - fipa\_string\_s, 74
  - fipa\_word\_s, 77
  - mtp\_http\_s, 162
- content\_length
  - mtp\_http\_s, 161
- content\_type
  - mtp\_http\_content\_s, 159
  - mtp\_http\_s, 161
- CONTINUE
  - mtp\_http.h, 378
- conversation\_id
  - fipa\_acl\_message\_s, 62
- CREATED
  - mtp\_http.h, 378
- ctx\_dec
  - ssl\_context, 194
- ctx\_enc
  - ssl\_context, 194
- ctxlen
  - ssl\_context, 193
- cur\_node
  - mxml\_index\_s, 166
- current
  - mxml\_fdbuf\_s, 165
- custom
  - mxml\_value\_u, 173
- D
  - rsa\_context, 175
- DATA
  - list.h, 433
- data
  - interpreter\_variable\_data\_s, 88
  - mtp\_http\_content\_s, 159
  - mxml\_custom\_s, 164
- data\_structures.c
  - AP\_QUEUE\_SEARCH\_TEMPLATE, 219
  - AP\_QUEUE\_STD\_DEFN\_TEMPLATE, 219
- data\_structures.h
  - agent\_mailbox\_p, 285
  - AP\_QUEUE\_DECL\_TEMPLATE, 284
  - AP\_QUEUE\_GENERIC\_DECL\_TEMPLATE, 284
  - int, 285
  - Remove, 285
  - RemoveID, 285
  - RemoveName, 285
  - Search, 285
  - SearchName, 285
- data\_type
  - interpreter\_variable\_data\_s, 87
- datastate
  - agent\_s, 39
- date
  - fipa\_acl\_Param\_s, 65
  - mtp\_http\_s, 161
- datetime
  - fipa\_expression\_s::content\_u, 71
- day

- [\\_x509\\_time](#), 28
  - [fipa\\_DateTime\\_s](#), 68
- [dealloc\\_command](#)
  - [cmd\\_prompt.c](#), 216
  - [cmd\\_prompt.h](#), 276
- [Default](#)
  - [LibMC::Properties::Settings](#), 178
- [default\\_agent\\_status](#)
  - [LibMC::MCAgency::MCAgencyOptions\\_t](#), 135
  - [MCAgencyOptions\\_s](#), 136
- [default\\_agentstatus](#)
  - [agency\\_s](#), 32
  - [LibMC::MCAgency::MCAgency\\_t](#), 134
  - [mc\\_platform\\_s](#), 93
- [default\\_callback](#)
  - [mxml-entity.c](#), 458
- [DEFAULT\\_HOSTNAME\\_LENGTH](#)
  - [mc\\_platform.c](#), 436
- [defaultInstance](#)
  - [LibMC::Properties::Settings](#), 178
- [DeleteAgent](#)
  - [LibMC::MCAgent](#), 141
- [des.c](#)
  - [\\_CRT\\_SECURE\\_NO\\_DEPRECATED](#), 660
  - [\\_des\\_src](#), 666
  - [des3\\_cbc\\_decrypt](#), 662
  - [des3\\_cbc\\_encrypt](#), 662
  - [des3\\_crypt](#), 662
  - [des3\\_decrypt](#), 663
  - [des3\\_encrypt](#), 663
  - [des3\\_set\\_2keys](#), 663
  - [des3\\_set\\_3keys](#), 663
  - [des\\_cbc\\_decrypt](#), 664
  - [des\\_cbc\\_encrypt](#), 664
  - [des\\_crypt](#), 664
  - [des\\_decrypt](#), 665
  - [des\\_encrypt](#), 665
  - [DES\\_FP](#), 660
  - [DES\\_IP](#), 660
  - [des\\_main\\_ks](#), 665
  - [DES\\_ROUND](#), 661
  - [des\\_self\\_test](#), 665
  - [des\\_set\\_key](#), 665
  - [GET\\_UINT32\\_BE](#), 661
  - [LHs](#), 666
  - [PUT\\_UINT32\\_BE](#), 661
  - [RHs](#), 666
  - [SB1](#), 666
  - [SB2](#), 667
  - [SB3](#), 667
  - [SB4](#), 668
  - [SB5](#), 668
  - [SB6](#), 668
  - [SB7](#), 669
  - [SB8](#), 669
- [des.h](#)
  - [des3\\_cbc\\_decrypt](#), 538
  - [des3\\_cbc\\_encrypt](#), 538
  - [des3\\_decrypt](#), 538
  - [des3\\_encrypt](#), 539
  - [des3\\_set\\_2keys](#), 539
  - [des3\\_set\\_3keys](#), 539
  - [des\\_cbc\\_decrypt](#), 539
  - [des\\_cbc\\_encrypt](#), 540
  - [des\\_decrypt](#), 540
  - [des\\_encrypt](#), 540
  - [des\\_self\\_test](#), 541
  - [des\\_set\\_key](#), 541
- [des3\\_cbc\\_decrypt](#)
  - [des.c](#), 662
  - [des.h](#), 538
- [des3\\_cbc\\_encrypt](#)
  - [des.c](#), 662
  - [des.h](#), 538
- [des3\\_context](#), 53
  - [dsk](#), 53
  - [esk](#), 53
- [des3\\_crypt](#)
  - [des.c](#), 662
- [des3\\_decrypt](#)
  - [des.c](#), 663
  - [des.h](#), 538
- [des3\\_encrypt](#)
  - [des.c](#), 663
  - [des.h](#), 539
- [des3\\_set\\_2keys](#)
  - [des.c](#), 663
  - [des.h](#), 539
- [des3\\_set\\_3keys](#)
  - [des.c](#), 663
  - [des.h](#), 539
- [des\\_cbc\\_decrypt](#)
  - [des.c](#), 664
  - [des.h](#), 539
- [des\\_cbc\\_encrypt](#)
  - [des.c](#), 664
  - [des.h](#), 540
- [des\\_context](#), 54
  - [dsk](#), 54
  - [esk](#), 54
- [des\\_crypt](#)
  - [des.c](#), 664
- [des\\_decrypt](#)
  - [des.c](#), 665
  - [des.h](#), 540
- [des\\_encrypt](#)
  - [des.c](#), 665

- des.h, 540
- DES\_FP
  - des.c, 660
- DES\_IP
  - des.c, 660
- des\_main\_ks
  - des.c, 665
- DES\_ROUND
  - des.c, 661
- des\_self\_test
  - des.c, 665
  - des.h, 541
- des\_set\_key
  - des.c, 665
  - des.h, 541
- Destroy
  - LibMC::MCAclMessage, 102
- destroy
  - mxml\_custom\_s, 164
- df
  - mc\_platform\_s, 92
- df.c
  - df\_Add, 222
  - df\_AddRequest, 222
  - df\_Destroy, 222
  - df\_Initialize, 222
  - df\_node\_Destroy, 222
  - df\_ProcessRequest, 222
  - df\_request\_list\_Destroy, 222
  - df\_request\_list\_New, 223
  - df\_request\_list\_node\_Destroy, 223
  - df\_request\_list\_node\_New, 223
  - df\_request\_list\_Pop, 223
  - df\_request\_search\_Destroy, 223
  - df\_request\_search\_New, 223
  - df\_SearchForService, 223
  - df\_Start, 224
  - df\_Thread, 224
  - REQUEST, 221
  - request\_handler\_DEREGISTER, 224
  - request\_handler\_REGISTER, 224
  - request\_handler\_SEARCH, 224
  - request\_handler\_SUBSCRIBE, 224
- df.h
  - cond, 290
  - df\_AddRequest, 288
  - df\_Destroy, 288
  - df\_Initialize, 288
  - df\_node\_Destroy, 288
  - df\_ProcessRequest, 288
  - df\_request\_list\_Destroy, 288
  - df\_request\_list\_index\_e, 287
  - df\_request\_list\_New, 288
  - df\_request\_list\_node\_Destroy, 288
  - df\_request\_list\_node\_New, 289
  - df\_request\_list\_Pop, 289
  - df\_request\_search\_Destroy, 289
  - df\_request\_search\_New, 289
  - df\_SearchForService, 289
  - df\_Start, 289
  - df\_Thread, 289
  - HERE, 287
  - INSERT, 287
  - lock, 290
  - mc\_platform, 290
  - MISC, 287
  - node, 290
  - num\_entries, 290
  - REQUEST, 287
  - request\_list, 290
  - search\_results, 291
  - search\_string, 291
  - service\_list, 291
  - service\_types\_e, 287
  - size, 291
  - SOME, 287
  - STRUCT, 289, 290
  - thread, 291
  - TYPES, 287
  - waiting, 291
  - waiting\_cond, 291
  - waiting\_lock, 291
  - ZERO, 287
- df\_Add
  - df.c, 222
- df\_AddRequest
  - df.c, 222
  - df.h, 288
- df\_Destroy
  - df.c, 222
  - df.h, 288
- df\_Initialize
  - df.c, 222
  - df.h, 288
- df\_node\_Destroy
  - df.c, 222
  - df.h, 288
- df\_ProcessRequest
  - df.c, 222
  - df.h, 288
- df\_request.x.h
  - register, 292
  - REQUEST, 292
  - search, 292
- df\_request\_list\_Destroy
  - df.c, 222
  - df.h, 288
- df\_request\_list\_index\_e

- df.h, [287](#)
- df\_request\_list\_New
  - df.c, [223](#)
  - df.h, [288](#)
- df\_request\_list\_node\_Destroy
  - df.c, [223](#)
  - df.h, [288](#)
- df\_request\_list\_node\_New
  - df.c, [223](#)
  - df.h, [289](#)
- df\_request\_list\_Pop
  - df.c, [223](#)
  - df.h, [289](#)
- df\_request\_search\_Destroy
  - df.c, [223](#)
  - df.h, [289](#)
- df\_request\_search\_New
  - df.c, [223](#)
  - df.h, [289](#)
- df\_SearchForService
  - df.c, [223](#)
  - df.h, [289](#)
- df\_Start
  - df.c, [224](#)
  - df.h, [289](#)
- df\_Thread
  - df.c, [224](#)
  - df.h, [289](#)
- dh\_client.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, [770](#)
  - main, [770](#)
  - SERVER\_NAME, [770](#)
  - SERVER\_PORT, [770](#)
- dh\_genprime.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, [772](#)
  - DH\_P\_SIZE, [772](#)
  - GENERATOR, [772](#)
  - main, [772](#)
- DH\_P\_SIZE
  - dh\_genprime.c, [772](#)
- dh\_server.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, [773](#)
  - main, [773](#)
  - PLAINTEXT, [773](#)
  - SERVER\_PORT, [773](#)
- dhm.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, [671](#)
  - \_dhm\_src, [674](#)
  - dhm\_calc\_secret, [672](#)
  - dhm\_free, [672](#)
  - dhm\_make\_params, [672](#)
  - dhm\_make\_public, [673](#)
  - DHM\_MPI\_EXPORT, [671](#)
  - dhm\_read\_bignum, [673](#)
  - dhm\_read\_params, [673](#)
  - dhm\_read\_public, [674](#)
  - dhm\_self\_test, [674](#)
- dhm.h
  - dhm\_calc\_secret, [543](#)
  - dhm\_free, [544](#)
  - dhm\_make\_params, [544](#)
  - dhm\_make\_public, [544](#)
  - dhm\_read\_params, [545](#)
  - dhm\_read\_public, [545](#)
  - dhm\_self\_test, [546](#)
  - ERR\_DHM\_BAD\_INPUT\_DATA, [543](#)
  - ERR\_DHM\_CALC\_SECRET\_FAILED, [543](#)
  - ERR\_DHM\_MAKE\_PARAMS\_FAILED, [543](#)
  - ERR\_DHM\_MAKE\_PUBLIC\_FAILED, [543](#)
  - ERR\_DHM\_READ\_PARAMS\_FAILED, [543](#)
  - ERR\_DHM\_READ\_PUBLIC\_FAILED, [543](#)
- dhm\_calc\_secret
  - dhm.c, [672](#)
  - dhm.h, [543](#)
- dhm\_context, [55](#)
  - G, [55](#)
  - GX, [56](#)
  - GY, [56](#)
  - K, [56](#)
  - len, [55](#)
  - P, [55](#)
  - RP, [56](#)
  - X, [55](#)
- dhm\_ctx
  - ssl\_context, [192](#)
- dhm\_free
  - dhm.c, [672](#)
  - dhm.h, [544](#)
- dhm\_G
  - ssl\_server.c, [784](#)
- dhm\_make\_params
  - dhm.c, [672](#)
  - dhm.h, [544](#)
- dhm\_make\_public
  - dhm.c, [673](#)
  - dhm.h, [544](#)
- DHM\_MPI\_EXPORT
  - dhm.c, [671](#)
- dhm\_P
  - ssl\_server.c, [784](#)
- dhm\_read\_bignum
  - dhm.c, [673](#)
- dhm\_read\_params
  - dhm.c, [673](#)
  - dhm.h, [545](#)
- dhm\_read\_public
  - dhm.c, [674](#)
  - dhm.h, [545](#)

- dhm\_self\_test
  - dhm.c, [674](#)
  - dhm.h, [546](#)
- DP
  - rsa\_context, [175](#)
- DQ
  - rsa\_context, [175](#)
- drk
  - aes\_context, [30](#)
- dsa.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, [676](#)
  - rsa\_check\_privkey, [676](#)
  - rsa\_check\_pubkey, [676](#)
  - rsa\_free, [676](#)
  - rsa\_gen\_key, [676](#)
  - rsa\_pkcs1\_decrypt, [677](#)
  - rsa\_pkcs1\_encrypt, [677](#)
  - rsa\_pkcs1\_sign, [677](#)
  - rsa\_pkcs1\_verify, [678](#)
  - rsa\_private, [678](#)
  - rsa\_public, [679](#)
  - rsa\_self\_test, [679](#)
- dsk
  - des3\_context, [53](#)
  - des\_context, [54](#)
- dynstring.c
  - dynstring\_Append, [225](#)
  - dynstring\_Destroy, [225](#)
  - dynstring\_New, [225](#)
- dynstring.h
  - COMPOSE\_BLOCKSIZE, [293](#)
  - dynstring\_Append, [293](#)
  - dynstring\_Destroy, [293](#)
  - dynstring\_New, [294](#)
  - dynstring\_p, [293](#)
  - dynstring\_t, [293](#)
- dynstring\_Append
  - dynstring.c, [225](#)
  - dynstring.h, [293](#)
- dynstring\_Destroy
  - dynstring.c, [225](#)
  - dynstring.h, [293](#)
- dynstring\_New
  - dynstring.c, [225](#)
  - dynstring.h, [294](#)
- dynstring\_p
  - dynstring.h, [293](#)
- dynstring\_s, [57](#)
  - len, [57](#)
  - message, [57](#)
  - size, [57](#)
- dynstring\_t
  - dynstring.h, [293](#)
- E
  - rsa\_context, [175](#)
- element
  - mxml\_value\_u, [172](#)
- enable\_security
  - agency\_s, [32](#)
  - LibMC::MCAgency::MCAgency\_t, [134](#)
  - LibMC::MCAgency::MCAgencyOptions\_t, [135](#)
  - MCAgencyOptions\_s, [136](#)
- ENCODE\_UTF16BE
  - mxml-file.c, [461](#)
- ENCODE\_UTF16LE
  - mxml-file.c, [461](#)
- ENCODE\_UTF8
  - mxml-file.c, [461](#)
- encoding
  - fipa\_acl\_message\_s, [62](#)
- ENCRYPTED\_DATA
  - message.h, [374](#)
- ENCRYPTION\_INITIALIZE
  - message.h, [374](#)
- End
  - LibMC::MCAgency, [111](#)
- end
  - mxml\_fdbuf\_s, [165](#)
- Ended
  - LibMC::MCAgency, [109](#)
- endpoint
  - ssl\_context, [190](#)
- enum.cxx
  - FOO\_BLUE, [495](#)
  - FOO\_ONE, [495](#)
  - FOO\_RED, [495](#)
  - FOO\_TWO, [495](#)
  - foo\_enum\_e, [495](#)
  - foo\_enum\_t, [495](#)
- erk
  - aes\_context, [30](#)
- ERR
  - libmc.h, [328](#)
  - mc\_error.h, [367](#)
- err
  - mc\_platform\_s, [91](#)
- ERR\_ASN1\_INVALID\_DATA
  - x509.h, [619](#)
- ERR\_ASN1\_INVALID\_LENGTH
  - x509.h, [620](#)
- ERR\_ASN1\_LENGTH\_MISMATCH
  - x509.h, [620](#)
- ERR\_ASN1\_OUT\_OF\_DATA
  - x509.h, [620](#)
- ERR\_ASN1\_UNEXPECTED\_TAG
  - x509.h, [620](#)



- ERR\_BASE64\_BUFFER\_TOO\_SMALL  
base64.h, [515](#)
- ERR\_BASE64\_INVALID\_CHARACTER  
base64.h, [515](#)
- ERR\_DHM\_BAD\_INPUT\_DATA  
dhm.h, [543](#)
- ERR\_DHM\_CALC\_SECRET\_FAILED  
dhm.h, [543](#)
- ERR\_DHM\_MAKE\_PARAMS\_FAILED  
dhm.h, [543](#)
- ERR\_DHM\_MAKE\_PUBLIC\_FAILED  
dhm.h, [543](#)
- ERR\_DHM\_READ\_PARAMS\_FAILED  
dhm.h, [543](#)
- ERR\_DHM\_READ\_PUBLIC\_FAILED  
dhm.h, [543](#)
- ERR\_MPI\_BUFFER\_TOO\_SMALL  
bignum.h, [520](#)
- ERR\_MPI\_DIVISION\_BY\_ZERO  
bignum.h, [520](#)
- ERR\_MPI\_FILE\_IO\_ERROR  
bignum.h, [520](#)
- ERR\_MPI\_INVALID\_CHARACTER  
bignum.h, [520](#)
- ERR\_MPI\_INVALID\_PARAMETER  
bignum.h, [521](#)
- ERR\_MPI\_NEGATIVE\_VALUE  
bignum.h, [521](#)
- ERR\_MPI\_NOT\_ACCEPTABLE  
bignum.h, [521](#)
- ERR\_NET\_ACCEPT\_FAILED  
net.h, [563](#)
- ERR\_NET\_BIND\_FAILED  
net.h, [563](#)
- ERR\_NET\_CONN\_RESET  
net.h, [563](#)
- ERR\_NET\_CONNECT\_FAILED  
net.h, [563](#)
- ERR\_NET\_LISTEN\_FAILED  
net.h, [563](#)
- ERR\_NET\_RECV\_FAILED  
net.h, [563](#)
- ERR\_NET\_SEND\_FAILED  
net.h, [563](#)
- ERR\_NET\_SOCKET\_FAILED  
net.h, [563](#)
- ERR\_NET\_UNKNOWN\_HOST  
net.h, [563](#)
- ERR\_NET\_WOULD\_BLOCK  
net.h, [564](#)
- ERR\_RSA\_BAD\_INPUT\_DATA  
rsa.h, [569](#)
- ERR\_RSA\_INVALID\_PADDING  
rsa.h, [569](#)
- ERR\_RSA\_KEY\_CHK\_FAILED  
rsa.h, [569](#)
- ERR\_RSA\_KEY\_GEN\_FAILED  
rsa.h, [569](#)
- ERR\_RSA\_KEY\_RD\_FAILED  
rsa.h, [569](#)
- ERR\_RSA\_KEY\_WR\_FAILED  
rsa.h, [569](#)
- ERR\_RSA\_PRIVATE\_FAILED  
rsa.h, [570](#)
- ERR\_RSA\_PUBLIC\_FAILED  
rsa.h, [570](#)
- ERR\_RSA\_VERIFY\_FAILED  
rsa.h, [570](#)
- ERR\_SSL\_BAD\_HS\_CERTIFICATE  
ssl.h, [597](#)
- ERR\_SSL\_BAD\_HS\_CERTIFICATE\_REQUEST  
ssl.h, [597](#)
- ERR\_SSL\_BAD\_HS\_CERTIFICATE\_VERIFY  
ssl.h, [597](#)
- ERR\_SSL\_BAD\_HS\_CHANGE\_CIPHER\_SPEC  
ssl.h, [597](#)
- ERR\_SSL\_BAD\_HS\_CLIENT\_HELLO  
ssl.h, [597](#)
- ERR\_SSL\_BAD\_HS\_CLIENT\_KEY\_-  
EXCHANGE  
ssl.h, [598](#)
- ERR\_SSL\_BAD\_HS\_FINISHED  
ssl.h, [598](#)
- ERR\_SSL\_BAD\_HS\_SERVER\_HELLO  
ssl.h, [598](#)
- ERR\_SSL\_BAD\_HS\_SERVER\_HELLO\_DONE  
ssl.h, [598](#)
- ERR\_SSL\_BAD\_HS\_SERVER\_KEY\_-  
EXCHANGE  
ssl.h, [598](#)
- ERR\_SSL\_CA\_CHAIN\_REQUIRED  
ssl.h, [598](#)
- ERR\_SSL\_CERTIFICATE\_REQUIRED  
ssl.h, [598](#)
- ERR\_SSL\_CERTIFICATE\_TOO\_LARGE  
ssl.h, [598](#)
- ERR\_SSL\_FATAL\_ALERT\_MESSAGE  
ssl.h, [598](#)
- ERR\_SSL\_FEATURE\_UNAVAILABLE  
ssl.h, [599](#)
- ERR\_SSL\_INVALID\_MAC  
ssl.h, [599](#)
- ERR\_SSL\_INVALID\_MODULUS\_SIZE  
ssl.h, [599](#)
- ERR\_SSL\_INVALID\_RECORD  
ssl.h, [599](#)
- ERR\_SSL\_NO\_CIPHER\_CHOSEN  
ssl.h, [599](#)

- ERR\_SSL\_NO\_CLIENT\_CERTIFICATE
  - ssl.h, [599](#)
- ERR\_SSL\_NO\_SESSION\_FOUND
  - ssl.h, [599](#)
- ERR\_SSL\_PEER\_CLOSE\_NOTIFY
  - ssl.h, [599](#)
- ERR\_SSL\_PEER\_VERIFY\_FAILED
  - ssl.h, [599](#)
- ERR\_SSL\_PRIVATE\_KEY\_REQUIRED
  - ssl.h, [600](#)
- ERR\_SSL\_UNEXPECTED\_MESSAGE
  - ssl.h, [600](#)
- ERR\_SSL\_UNKNOWN\_CIPHER
  - ssl.h, [600](#)
- ERR\_X509\_CERT\_INVALID\_ALG
  - x509.h, [620](#)
- ERR\_X509\_CERT\_INVALID\_DATE
  - x509.h, [620](#)
- ERR\_X509\_CERT\_INVALID\_EXTENSIONS
  - x509.h, [620](#)
- ERR\_X509\_CERT\_INVALID\_FORMAT
  - x509.h, [620](#)
- ERR\_X509\_CERT\_INVALID\_NAME
  - x509.h, [621](#)
- ERR\_X509\_CERT\_INVALID\_PEM
  - x509.h, [621](#)
- ERR\_X509\_CERT\_INVALID\_PUBKEY
  - x509.h, [621](#)
- ERR\_X509\_CERT\_INVALID\_SERIAL
  - x509.h, [621](#)
- ERR\_X509\_CERT\_INVALID\_SIGNATURE
  - x509.h, [621](#)
- ERR\_X509\_CERT\_INVALID\_VERSION
  - x509.h, [621](#)
- ERR\_X509\_CERT\_SIG\_MISMATCH
  - x509.h, [621](#)
- ERR\_X509\_CERT\_UNKNOWN\_PK\_ALG
  - x509.h, [621](#)
- ERR\_X509\_CERT\_UNKNOWN\_SIG\_ALG
  - x509.h, [621](#)
- ERR\_X509\_CERT\_UNKNOWN\_VERSION
  - x509.h, [622](#)
- ERR\_X509\_FEATURE\_UNAVAILABLE
  - x509.h, [622](#)
- ERR\_X509\_KEY\_INVALID\_ENC\_IV
  - x509.h, [622](#)
- ERR\_X509\_KEY\_INVALID\_FORMAT
  - x509.h, [622](#)
- ERR\_X509\_KEY\_INVALID\_PEM
  - x509.h, [622](#)
- ERR\_X509\_KEY\_INVALID\_VERSION
  - x509.h, [622](#)
- ERR\_X509\_KEY\_PASSWORD\_MISMATCH
  - x509.h, [622](#)
- ERR\_X509\_KEY\_PASSWORD\_REQUIRED
  - x509.h, [622](#)
- ERR\_X509\_KEY\_UNKNOWN\_ENC\_ALG
  - x509.h, [622](#)
- ERR\_X509\_SIG\_VERIFY\_FAILED
  - x509.h, [623](#)
- error\_code\_e
  - libmc.h, [327](#)
  - mc\_error.h, [367](#)
- error\_code\_t
  - libmc.h, [327](#)
  - mc\_error.h, [367](#)
- esk
  - des3\_context, [53](#)
  - des\_context, [54](#)
- exec\_command
  - cmd\_prompt.c, [216](#)
  - cmd\_prompt.h, [276](#)
- EXPECTATION\_FAILED
  - mtp\_http.h, [379](#)
- EXPONENT
  - rsa\_genkey.c, [776](#)
- EXPORTMC
  - libmc.h, [326](#)
- expression
  - fipa\_expression\_s::content\_u, [71](#)
- F
  - md4.c, [689](#)
  - md5.c, [695](#)
  - sha1.c, [712](#)
- f
  - class.cxx, [494](#)
- F0
  - sha2.c, [719](#)
  - sha4.c, [726](#)
- F1
  - sha2.c, [719](#)
  - sha4.c, [726](#)
- FAILED\_DEPENDANCY
  - mtp\_http.h, [379](#)
- fd
  - mxml\_fdbuf\_s, [165](#)
- filename
  - agency\_s, [31](#)
  - LibMC::MCAgency::MCAgency\_t, [133](#)
- FindAgentByID
  - LibMC::MCAgency, [122](#)
- FindAgentByName
  - LibMC::MCAgency, [122](#)
- FIPA\_ACCEPT\_PROPOSAL
  - fipa\_acl.h, [299](#)
  - LibMC::MCAclMessage, [99](#)
- FIPA\_ACL

- message.h, 374
- fipa\_acl.h
  - FIPA\_ACCEPT\_PROPOSAL, 299
  - FIPA\_AGREE, 299
  - FIPA\_CALL\_FOR\_PROPOSAL, 299
  - FIPA\_CANCEL, 299
  - FIPA\_CONFIRM, 299
  - FIPA\_DISCONFIRM, 299
  - FIPA\_ERROR, 299
  - FIPA\_EXPR\_DATETIME, 299
  - FIPA\_EXPR\_EXPRESSION, 299
  - FIPA\_EXPR\_NUMBER, 299
  - FIPA\_EXPR\_STRING, 299
  - FIPA\_EXPR\_WORD, 299
  - FIPA\_EXPR\_ZERO, 299
  - FIPA\_FAILURE, 299
  - FIPA\_INFORM, 299
  - FIPA\_INFORM\_IF, 299
  - FIPA\_INFORM\_REF, 299
  - FIPA\_NOT\_UNDERSTOOD, 299
  - FIPA\_PROPOGATE, 299
  - FIPA\_PROPOSE, 299
  - FIPA\_PROXY, 299
  - FIPA\_QUERY\_IF, 299
  - FIPA\_QUERY\_REF, 299
  - FIPA\_REFUSE, 300
  - FIPA\_REJECT\_PROPOSAL, 300
  - FIPA\_REQUEST, 300
  - FIPA\_REQUEST\_WHEN, 300
  - FIPA\_REQUEST\_WHENEVER, 300
  - FIPA\_SUBSCRIBE, 300
  - FIPA\_ZERO, 299
- FIPA\_AGREE
  - fipa\_acl.h, 299
  - LibMC::MCAclMessage, 99
- FIPA\_CALL\_FOR\_PROPOSAL
  - fipa\_acl.h, 299
  - LibMC::MCAclMessage, 99
- FIPA\_CANCEL
  - fipa\_acl.h, 299
  - LibMC::MCAclMessage, 99
- FIPA\_CONFIRM
  - fipa\_acl.h, 299
  - LibMC::MCAclMessage, 99
- FIPA\_DISCONFIRM
  - fipa\_acl.h, 299
  - LibMC::MCAclMessage, 99
- FIPA\_ERROR
  - fipa\_acl.h, 299
  - LibMC::MCAclMessage, 99
- FIPA\_EXPR\_DATETIME
  - fipa\_acl.h, 299
- FIPA\_EXPR\_EXPRESSION
  - fipa\_acl.h, 299
- FIPA\_EXPR\_NUMBER
  - fipa\_acl.h, 299
- FIPA\_EXPR\_STRING
  - fipa\_acl.h, 299
- FIPA\_EXPR\_WORD
  - fipa\_acl.h, 299
- FIPA\_EXPR\_ZERO
  - fipa\_acl.h, 299
- FIPA\_FAILURE
  - fipa\_acl.h, 299
  - LibMC::MCAclMessage, 99
- FIPA\_INFORM
  - fipa\_acl.h, 299
  - LibMC::MCAclMessage, 99
- FIPA\_INFORM\_IF
  - fipa\_acl.h, 299
  - LibMC::MCAclMessage, 99
- FIPA\_INFORM\_REF
  - fipa\_acl.h, 299
  - LibMC::MCAclMessage, 99
- FIPA\_NOT\_UNDERSTOOD
  - fipa\_acl.h, 299
  - LibMC::MCAclMessage, 99
- FIPA\_PROPOGATE
  - fipa\_acl.h, 299
  - LibMC::MCAclMessage, 99
- FIPA\_PROPOSE
  - fipa\_acl.h, 299
  - LibMC::MCAclMessage, 99
- FIPA\_PROXY
  - fipa\_acl.h, 299
  - LibMC::MCAclMessage, 99
- FIPA\_QUERY\_IF
  - fipa\_acl.h, 299
  - LibMC::MCAclMessage, 99
- FIPA\_QUERY\_REF
  - fipa\_acl.h, 299
  - LibMC::MCAclMessage, 99
- FIPA\_REFUSE
  - fipa\_acl.h, 300
  - LibMC::MCAclMessage, 99
- FIPA\_REJECT\_PROPOSAL
  - fipa\_acl.h, 300
  - LibMC::MCAclMessage, 99
- FIPA\_REQUEST
  - fipa\_acl.h, 300
  - LibMC::MCAclMessage, 99
- FIPA\_REQUEST\_WHEN
  - fipa\_acl.h, 300
  - LibMC::MCAclMessage, 99
- FIPA\_REQUEST\_WHENEVER
  - fipa\_acl.h, 300
  - LibMC::MCAclMessage, 99
- FIPA\_SUBSCRIBE
  - LibMC::MCAclMessage, 99

- fipa\_acl.h, 300
  - LibMC::MCAclMessage, 99
- FIPA\_ZERO
  - fipa\_acl.h, 299
  - LibMC::MCAclMessage, 99
- fipa\_acl.c
  - fipa\_acl\_Compose, 227
  - fipa\_acl\_message\_Copy, 228
  - fipa\_acl\_message\_Destroy, 228
  - fipa\_acl\_message\_New, 228
  - fipa\_acl\_Parse, 228
  - fipa\_agent\_identifier\_Compose, 228
  - fipa\_agent\_identifier\_Copy, 228
  - fipa\_agent\_identifier\_Destroy, 229
  - fipa\_agent\_identifier\_New, 229
  - fipa\_agent\_identifier\_Parse, 229
  - fipa\_agent\_identifier\_set\_Compose, 229
  - fipa\_agent\_identifier\_set\_Copy, 229
  - fipa\_agent\_identifier\_set\_Destroy, 230
  - fipa\_agent\_identifier\_set\_New, 230
  - fipa\_agent\_identifier\_set\_Parse, 230
  - fipa\_CheckNextToken, 230
  - fipa\_DateTime\_Compose, 230
  - fipa\_DateTime\_Copy, 230
  - fipa\_DateTime\_Destroy, 231
  - fipa\_DateTime\_New, 231
  - fipa\_datetime\_Parse, 231
  - fipa\_expression\_Compose, 231
  - fipa\_expression\_Copy, 231
  - fipa\_expression\_Destroy, 231
  - fipa\_expression\_New, 232
  - fipa\_expression\_Parse, 232
  - fipa\_GetAtom, 232
  - fipa\_GetNextWord, 232
  - fipa\_GetWholeToken, 232
  - fipa\_message\_parameter\_Parse, 232
  - fipa\_message\_string\_Copy, 233
  - fipa\_message\_string\_Destroy, 233
  - fipa\_message\_string\_New, 233
  - fipa\_message\_type\_Parse, 233
  - fipa\_number\_Compose, 233
  - fipa\_number\_Copy, 234
  - fipa\_number\_Destroy, 234
  - fipa\_number\_New, 234
  - fipa\_performative\_Compose, 234
  - fipa\_Reply, 234
  - fipa\_string\_Compose, 234
  - fipa\_string\_Copy, 235
  - fipa\_string\_Destroy, 235
  - fipa\_string\_New, 235
  - fipa\_string\_Parse, 235
  - fipa\_url\_Compose, 235
  - fipa\_url\_Copy, 235
  - fipa\_url\_Destroy, 235
  - fipa\_url\_New, 235
  - fipa\_url\_Parse, 236
  - fipa\_url\_sequence\_Compose, 236
  - fipa\_url\_sequence\_Copy, 236
  - fipa\_url\_sequence\_Destroy, 236
  - fipa\_url\_sequence\_New, 236
  - fipa\_url\_sequence\_Parse, 236
  - fipa\_word\_Compose, 237
  - fipa\_word\_Copy, 237
  - fipa\_word\_Destroy, 237
  - fipa\_word\_New, 237
  - fipa\_word\_Parse, 237
  - FREEMEM, 227
- fipa\_acl.h
  - fipa\_acl\_Compose, 300
  - fipa\_acl\_message\_Copy, 300
  - fipa\_acl\_message\_Destroy, 300
  - fipa\_acl\_message\_New, 300
  - fipa\_acl\_message\_p, 297
  - fipa\_acl\_message\_t, 297
  - fipa\_acl\_Parse, 301
  - fipa\_agent\_identifier\_Compose, 301
  - fipa\_agent\_identifier\_Copy, 301
  - fipa\_agent\_identifier\_Destroy, 301
  - fipa\_agent\_identifier\_New, 301
  - fipa\_agent\_identifier\_p, 297
  - fipa\_agent\_identifier\_Parse, 301
  - fipa\_agent\_identifier\_set\_Compose, 301
  - fipa\_agent\_identifier\_set\_Copy, 302
  - fipa\_agent\_identifier\_set\_Destroy, 302
  - fipa\_agent\_identifier\_set\_New, 302
  - fipa\_agent\_identifier\_set\_p, 297
  - fipa\_agent\_identifier\_set\_Parse, 302
  - fipa\_agent\_identifier\_set\_t, 297
  - fipa\_agent\_identifier\_t, 298
  - fipa\_CheckNextToken, 302
  - fipa\_DateTime\_Compose, 302
  - fipa\_DateTime\_Copy, 303
  - fipa\_DateTime\_Destroy, 303
  - fipa\_DateTime\_New, 303
  - fipa\_DateTime\_p, 298
  - fipa\_datetime\_Parse, 303
  - fipa\_DateTime\_t, 298
  - fipa\_expression\_Compose, 303
  - fipa\_expression\_Copy, 303
  - fipa\_expression\_Destroy, 304
  - fipa\_expression\_New, 304
  - fipa\_expression\_p, 298
  - fipa\_expression\_Parse, 304
  - fipa\_expression\_t, 298
  - fipa\_expression\_type\_e, 299
  - fipa\_expression\_type\_t, 298
  - fipa\_GetAtom, 304
  - fipa\_GetNextWord, 304

- fipa\_GetWholeToken, 305
- fipa\_message\_parameter\_Parse, 305
- fipa\_message\_string\_Copy, 305
- fipa\_message\_string\_Destroy, 305
- fipa\_message\_string\_New, 305
- fipa\_message\_string\_p, 298
- fipa\_message\_string\_t, 298
- fipa\_message\_type\_Parse, 305
- fipa\_number\_Compose, 305
- fipa\_number\_Copy, 306
- fipa\_number\_Destroy, 306
- fipa\_number\_New, 306
- fipa\_number\_p, 298
- fipa\_number\_t, 298
- fipa\_performative\_Compose, 306
- fipa\_performative\_e, 299
- fipa\_Reply, 306
- fipa\_string\_Compose, 306
- fipa\_string\_Copy, 307
- fipa\_string\_Destroy, 307
- fipa\_string\_New, 307
- fipa\_string\_p, 298
- fipa\_string\_Parse, 307
- fipa\_string\_t, 298
- fipa\_url\_Compose, 307
- fipa\_url\_Copy, 307
- fipa\_url\_Destroy, 307
- fipa\_url\_New, 308
- fipa\_url\_p, 298
- fipa\_url\_Parse, 308
- fipa\_url\_sequence\_Compose, 308
- fipa\_url\_sequence\_Copy, 308
- fipa\_url\_sequence\_Destroy, 308
- fipa\_url\_sequence\_New, 308
- fipa\_url\_sequence\_p, 298
- fipa\_url\_sequence\_Parse, 308
- fipa\_url\_sequence\_t, 298
- fipa\_url\_t, 299
- fipa\_word\_Compose, 309
- fipa\_word\_Copy, 309
- fipa\_word\_Destroy, 309
- fipa\_word\_New, 309
- fipa\_word\_p, 299
- fipa\_word\_Parse, 309
- fipa\_word\_t, 299
- fipa\_acl\_Compose
  - fipa\_acl.c, 227
  - fipa\_acl.h, 300
- fipa\_acl\_envelope.h
  - fipa\_acl\_envelope\_Copy, 311
  - fipa\_acl\_envelope\_Destroy, 311
  - fipa\_acl\_envelope\_New, 311
  - fipa\_acl\_envelope\_p, 311
  - fipa\_acl\_envelope\_Received\_Copy, 312
  - fipa\_acl\_envelope\_Received\_Destroy, 312
  - fipa\_acl\_envelope\_Received\_New, 312
  - fipa\_acl\_envelope\_Received\_t, 311
  - fipa\_acl\_envelope\_t, 311
  - fipa\_acl\_Param\_Copy, 312
  - fipa\_acl\_Param\_Destroy, 312
  - fipa\_acl\_Param\_New, 313
  - fipa\_acl\_Param\_t, 311
  - fipa\_envelope\_Compose, 313
  - fipa\_envelope\_Compose\_\_acl\_representation, 313
  - fipa\_envelope\_Compose\_\_date, 313
  - fipa\_envelope\_Compose\_\_envelope, 313
  - fipa\_envelope\_Compose\_\_from, 313
  - fipa\_envelope\_Compose\_\_intended\_receiver, 313
  - fipa\_envelope\_Compose\_\_params, 314
  - fipa\_envelope\_Compose\_\_payload\_encoding, 314
  - fipa\_envelope\_Compose\_\_to, 314
  - fipa\_envelope\_HandleAclRepresentation, 314
  - fipa\_envelope\_HandleComments, 314
  - fipa\_envelope\_HandleDate, 315
  - fipa\_envelope\_HandleEnvelope, 315
  - fipa\_envelope\_HandleFrom, 315
  - fipa\_envelope\_HandleIntendedReceiver, 315
  - fipa\_envelope\_HandleParams, 315
  - fipa\_envelope\_HandlePayloadEncoding, 316
  - fipa\_envelope\_HandlePayloadLength, 316
  - fipa\_envelope\_HandleReceived, 316
  - fipa\_envelope\_HandleTo, 316
  - fipa\_envelope\_Parse, 317
  - fipa\_envelope\_ParseAddresses, 317
  - fipa\_envelope\_ParseAgentIdentifier, 317
  - fipa\_envelope\_ParseResolvers, 317
- fipa\_acl\_envelope\_Copy
  - fipa\_acl\_envelope.h, 311
  - fipa\_envelope.c, 239
- fipa\_acl\_envelope\_Destroy
  - fipa\_acl\_envelope.h, 311
  - fipa\_envelope.c, 239
- fipa\_acl\_envelope\_New
  - fipa\_acl\_envelope.h, 311
  - fipa\_envelope.c, 239
- fipa\_acl\_envelope\_p
  - fipa\_acl\_envelope.h, 311
- fipa\_acl\_envelope\_Received\_Copy
  - fipa\_acl\_envelope.h, 312
  - fipa\_envelope.c, 239
- fipa\_acl\_envelope\_Received\_Destroy
  - fipa\_acl\_envelope.h, 312
  - fipa\_envelope.c, 239
- fipa\_acl\_envelope\_Received\_New
  - fipa\_acl\_envelope.h, 312

- fipa\_envelope.c, 240
- fipa\_acl\_envelope\_Received\_s, 58
  - received\_by, 58
  - received\_date, 58
  - received\_from, 58
  - received\_id, 58
  - received\_via, 58
- fipa\_acl\_envelope\_Received\_t
  - fipa\_acl\_envelope.h, 311
- fipa\_acl\_envelope\_s, 60
  - num\_params, 60
  - params, 60
- fipa\_acl\_envelope\_t
  - fipa\_acl\_envelope.h, 311
- fipa\_acl\_message\_Copy
  - fipa\_acl.c, 228
  - fipa\_acl.h, 300
- fipa\_acl\_message\_Destroy
  - fipa\_acl.c, 228
  - fipa\_acl.h, 300
- fipa\_acl\_message\_New
  - fipa\_acl.c, 228
  - fipa\_acl.h, 300
- fipa\_acl\_message\_p
  - fipa\_acl.h, 297
- fipa\_acl\_message\_s, 61
  - content, 62
  - conversation\_id, 62
  - encoding, 62
  - in\_reply\_to, 63
  - language, 62
  - ontology, 62
  - performative, 61
  - protocol, 62
  - receiver, 61
  - receiver\_num, 61
  - reply\_by, 63
  - reply\_to, 62
  - reply\_with, 63
  - sender, 61
- fipa\_acl\_message\_t
  - fipa\_acl.h, 297
- fipa\_acl\_Param\_Copy
  - fipa\_acl\_envelope.h, 312
  - fipa\_envelope.c, 240
- fipa\_acl\_Param\_Destroy
  - fipa\_acl\_envelope.h, 312
  - fipa\_envelope.c, 240
- fipa\_acl\_Param\_New
  - fipa\_acl\_envelope.h, 313
  - fipa\_envelope.c, 240
- fipa\_acl\_Param\_s, 64
  - acl\_representation, 64
  - comments, 64
  - date, 65
  - from, 64
  - intended\_receiver, 65
  - payload\_encoding, 65
  - payload\_length, 64
  - received, 65
  - to, 64
- fipa\_acl\_Param\_t
  - fipa\_acl\_envelope.h, 311
- fipa\_acl\_Parse
  - fipa\_acl.c, 228
  - fipa\_acl.h, 301
- fipa\_agent\_identifier\_Compose
  - fipa\_acl.c, 228
  - fipa\_acl.h, 301
- fipa\_agent\_identifier\_Copy
  - fipa\_acl.c, 228
  - fipa\_acl.h, 301
- fipa\_agent\_identifier\_Destroy
  - fipa\_acl.c, 229
  - fipa\_acl.h, 301
- fipa\_agent\_identifier\_New
  - fipa\_acl.c, 229
  - fipa\_acl.h, 301
- fipa\_agent\_identifier\_p
  - fipa\_acl.h, 297
- fipa\_agent\_identifier\_Parse
  - fipa\_acl.c, 229
  - fipa\_acl.h, 301
- fipa\_agent\_identifier\_s, 66
  - addresses, 66
  - name, 66
  - resolvers, 66
- fipa\_agent\_identifier\_set\_Compose
  - fipa\_acl.c, 229
  - fipa\_acl.h, 301
- fipa\_agent\_identifier\_set\_Copy
  - fipa\_acl.c, 229
  - fipa\_acl.h, 302
- fipa\_agent\_identifier\_set\_Destroy
  - fipa\_acl.c, 230
  - fipa\_acl.h, 302
- fipa\_agent\_identifier\_set\_New
  - fipa\_acl.c, 230
  - fipa\_acl.h, 302
- fipa\_agent\_identifier\_set\_p
  - fipa\_acl.h, 297
- fipa\_agent\_identifier\_set\_Parse
  - fipa\_acl.c, 230
  - fipa\_acl.h, 302
- fipa\_agent\_identifier\_set\_s, 67
  - fipa\_agent\_identifiers, 67
  - num, 67
  - retain\_order, 67

- fipa\_agent\_identifier\_set\_t
  - fipa\_acl.h, 297
- fipa\_agent\_identifier\_t
  - fipa\_acl.h, 298
- fipa\_agent\_identifiers
  - fipa\_agent\_identifier\_set\_s, 67
- fipa\_CheckNextToken
  - fipa\_acl.c, 230
  - fipa\_acl.h, 302
- fipa\_DateTime\_Compose
  - fipa\_acl.c, 230
  - fipa\_acl.h, 302
- fipa\_DateTime\_Copy
  - fipa\_acl.c, 230
  - fipa\_acl.h, 303
- fipa\_DateTime\_Destroy
  - fipa\_acl.c, 231
  - fipa\_acl.h, 303
- fipa\_DateTime\_New
  - fipa\_acl.c, 231
  - fipa\_acl.h, 303
- fipa\_DateTime\_p
  - fipa\_acl.h, 298
- fipa\_datetime\_Parse
  - fipa\_acl.c, 231
  - fipa\_acl.h, 303
- fipa\_DateTime\_s, 68
  - day, 68
  - hour, 68
  - is\_utc, 69
  - millisecond, 69
  - minute, 68
  - month, 68
  - second, 68
  - sign, 69
  - year, 68
- fipa\_DateTime\_t
  - fipa\_acl.h, 298
- fipa\_envelope.c
  - fipa\_acl\_envelope\_Copy, 239
  - fipa\_acl\_envelope\_Destroy, 239
  - fipa\_acl\_envelope\_New, 239
  - fipa\_acl\_envelope\_Received\_Copy, 239
  - fipa\_acl\_envelope\_Received\_Destroy, 239
  - fipa\_acl\_envelope\_Received\_New, 240
  - fipa\_acl\_Param\_Copy, 240
  - fipa\_acl\_Param\_Destroy, 240
  - fipa\_acl\_Param\_New, 240
  - fipa\_envelope\_Compose, 240
  - fipa\_envelope\_Compose\_\_acl\_representation, 240
  - fipa\_envelope\_Compose\_\_date, 241
  - fipa\_envelope\_Compose\_\_envelope, 241
  - fipa\_envelope\_Compose\_\_from, 241
  - fipa\_envelope\_Compose\_\_intended\_receiver, 241
  - fipa\_envelope\_Compose\_\_params, 241
  - fipa\_envelope\_Compose\_\_payload\_encoding, 242
  - fipa\_envelope\_Compose\_\_to, 242
  - fipa\_envelope\_HandleAclRepresentation, 242
  - fipa\_envelope\_HandleComments, 242
  - fipa\_envelope\_HandleDate, 242
  - fipa\_envelope\_HandleEnvelope, 242
  - fipa\_envelope\_HandleFrom, 243
  - fipa\_envelope\_HandleIntendedReceiver, 243
  - fipa\_envelope\_HandleParams, 243
  - fipa\_envelope\_HandlePayloadEncoding, 243
  - fipa\_envelope\_HandlePayloadLength, 243
  - fipa\_envelope\_HandleReceived, 244
  - fipa\_envelope\_HandleTo, 244
  - fipa\_envelope\_Parse, 244
  - fipa\_envelope\_ParseAddresses, 244
  - fipa\_envelope\_ParseAgentIdentifier, 245
  - fipa\_envelope\_ParseResolvers, 245
- fipa\_envelope\_Compose
  - fipa\_acl\_envelope.h, 313
  - fipa\_envelope.c, 240
- fipa\_envelope\_Compose\_\_acl\_representation
  - fipa\_acl\_envelope.h, 313
  - fipa\_envelope.c, 240
- fipa\_envelope\_Compose\_\_date
  - fipa\_acl\_envelope.h, 313
  - fipa\_envelope.c, 241
- fipa\_envelope\_Compose\_\_envelope
  - fipa\_acl\_envelope.h, 313
  - fipa\_envelope.c, 241
- fipa\_envelope\_Compose\_\_from
  - fipa\_acl\_envelope.h, 313
  - fipa\_envelope.c, 241
- fipa\_envelope\_Compose\_\_intended\_receiver
  - fipa\_acl\_envelope.h, 313
  - fipa\_envelope.c, 241
- fipa\_envelope\_Compose\_\_params
  - fipa\_acl\_envelope.h, 314
  - fipa\_envelope.c, 241
- fipa\_envelope\_Compose\_\_payload\_encoding
  - fipa\_acl\_envelope.h, 314
  - fipa\_envelope.c, 242
- fipa\_envelope\_Compose\_\_to
  - fipa\_acl\_envelope.h, 314
  - fipa\_envelope.c, 242
- fipa\_envelope\_HandleAclRepresentation
  - fipa\_acl\_envelope.h, 314
  - fipa\_envelope.c, 242
- fipa\_envelope\_HandleComments
  - fipa\_acl\_envelope.h, 314
  - fipa\_envelope.c, 242



- fipa\_envelope\_HandleDate
  - fipa\_acl\_envelope.h, 315
  - fipa\_envelope.c, 242
- fipa\_envelope\_HandleEnvelope
  - fipa\_acl\_envelope.h, 315
  - fipa\_envelope.c, 242
- fipa\_envelope\_HandleFrom
  - fipa\_acl\_envelope.h, 315
  - fipa\_envelope.c, 243
- fipa\_envelope\_HandleIntendedReceiver
  - fipa\_acl\_envelope.h, 315
  - fipa\_envelope.c, 243
- fipa\_envelope\_HandleParams
  - fipa\_acl\_envelope.h, 315
  - fipa\_envelope.c, 243
- fipa\_envelope\_HandlePayloadEncoding
  - fipa\_acl\_envelope.h, 316
  - fipa\_envelope.c, 243
- fipa\_envelope\_HandlePayloadLength
  - fipa\_acl\_envelope.h, 316
  - fipa\_envelope.c, 243
- fipa\_envelope\_HandleReceived
  - fipa\_acl\_envelope.h, 316
  - fipa\_envelope.c, 244
- fipa\_envelope\_HandleTo
  - fipa\_acl\_envelope.h, 316
  - fipa\_envelope.c, 244
- fipa\_envelope\_Parse
  - fipa\_acl\_envelope.h, 317
  - fipa\_envelope.c, 244
- fipa\_envelope\_ParseAddresses
  - fipa\_acl\_envelope.h, 317
  - fipa\_envelope.c, 244
- fipa\_envelope\_ParseAgentIdentifier
  - fipa\_acl\_envelope.h, 317
  - fipa\_envelope.c, 245
- fipa\_envelope\_ParseResolvers
  - fipa\_acl\_envelope.h, 317
  - fipa\_envelope.c, 245
- fipa\_expression\_Compose
  - fipa\_acl.c, 231
  - fipa\_acl.h, 303
- fipa\_expression\_Copy
  - fipa\_acl.c, 231
  - fipa\_acl.h, 303
- fipa\_expression\_Destroy
  - fipa\_acl.c, 231
  - fipa\_acl.h, 304
- fipa\_expression\_New
  - fipa\_acl.c, 232
  - fipa\_acl.h, 304
- fipa\_expression\_p
  - fipa\_acl.h, 298
- fipa\_expression\_Parse
  - fipa\_acl.c, 232
  - fipa\_acl.h, 304
- fipa\_expression\_s, 70
  - content, 70
  - type, 70
- fipa\_expression\_s::content\_u, 71
  - datetime, 71
  - expression, 71
  - number, 71
  - string, 71
  - word, 71
- fipa\_expression\_t
  - fipa\_acl.h, 298
- fipa\_expression\_type\_e
  - fipa\_acl.h, 299
- fipa\_expression\_type\_t
  - fipa\_acl.h, 298
- fipa\_GetAtom
  - fipa\_acl.c, 232
  - fipa\_acl.h, 304
- fipa\_GetNextWord
  - fipa\_acl.c, 232
  - fipa\_acl.h, 304
- fipa\_GetWholeToken
  - fipa\_acl.c, 232
  - fipa\_acl.h, 305
- fipa\_message\_parameter\_Parse
  - fipa\_acl.c, 232
  - fipa\_acl.h, 305
- fipa\_message\_string\_Copy
  - fipa\_acl.c, 233
  - fipa\_acl.h, 305
- fipa\_message\_string\_Destroy
  - fipa\_acl.c, 233
  - fipa\_acl.h, 305
- fipa\_message\_string\_New
  - fipa\_acl.c, 233
  - fipa\_acl.h, 305
- fipa\_message\_string\_p
  - fipa\_acl.h, 298
- fipa\_message\_string\_s, 72
  - message, 72
  - parse, 72
- fipa\_message\_string\_t
  - fipa\_acl.h, 298
- fipa\_message\_type\_Parse
  - fipa\_acl.c, 233
  - fipa\_acl.h, 305
- fipa\_number\_Compose
  - fipa\_acl.c, 233
  - fipa\_acl.h, 305
- fipa\_number\_Copy
  - fipa\_acl.c, 234
  - fipa\_acl.h, 306



- fipa\_number\_Destroy
  - fipa\_acl.c, [234](#)
  - fipa\_acl.h, [306](#)
- fipa\_number\_New
  - fipa\_acl.c, [234](#)
  - fipa\_acl.h, [306](#)
- fipa\_number\_p
  - fipa\_acl.h, [298](#)
- fipa\_number\_s, [73](#)
  - str, [73](#)
- fipa\_number\_t
  - fipa\_acl.h, [298](#)
- fipa\_performative\_Compose
  - fipa\_acl.c, [234](#)
  - fipa\_acl.h, [306](#)
- fipa\_performative\_e
  - fipa\_acl.h, [299](#)
- fipa\_Reply
  - fipa\_acl.c, [234](#)
  - fipa\_acl.h, [306](#)
- fipa\_string\_Compose
  - fipa\_acl.c, [234](#)
  - fipa\_acl.h, [306](#)
- fipa\_string\_Copy
  - fipa\_acl.c, [235](#)
  - fipa\_acl.h, [307](#)
- fipa\_string\_Destroy
  - fipa\_acl.c, [235](#)
  - fipa\_acl.h, [307](#)
- fipa\_string\_New
  - fipa\_acl.c, [235](#)
  - fipa\_acl.h, [307](#)
- fipa\_string\_p
  - fipa\_acl.h, [298](#)
- fipa\_string\_Parse
  - fipa\_acl.c, [235](#)
  - fipa\_acl.h, [307](#)
- fipa\_string\_s, [74](#)
  - content, [74](#)
- fipa\_string\_t
  - fipa\_acl.h, [298](#)
- fipa\_url\_Compose
  - fipa\_acl.c, [235](#)
  - fipa\_acl.h, [307](#)
- fipa\_url\_Copy
  - fipa\_acl.c, [235](#)
  - fipa\_acl.h, [307](#)
- fipa\_url\_Destroy
  - fipa\_acl.c, [235](#)
  - fipa\_acl.h, [307](#)
- fipa\_url\_New
  - fipa\_acl.c, [235](#)
  - fipa\_acl.h, [308](#)
- fipa\_url\_p
  - fipa\_acl.h, [298](#)
- fipa\_url\_Parse
  - fipa\_acl.c, [236](#)
  - fipa\_acl.h, [308](#)
- fipa\_url\_s, [75](#)
  - str, [75](#)
- fipa\_url\_sequence\_Compose
  - fipa\_acl.c, [236](#)
  - fipa\_acl.h, [308](#)
- fipa\_url\_sequence\_Copy
  - fipa\_acl.c, [236](#)
  - fipa\_acl.h, [308](#)
- fipa\_url\_sequence\_Destroy
  - fipa\_acl.c, [236](#)
  - fipa\_acl.h, [308](#)
- fipa\_url\_sequence\_New
  - fipa\_acl.c, [236](#)
  - fipa\_acl.h, [308](#)
- fipa\_url\_sequence\_p
  - fipa\_acl.h, [298](#)
- fipa\_url\_sequence\_Parse
  - fipa\_acl.c, [236](#)
  - fipa\_acl.h, [308](#)
- fipa\_url\_sequence\_s, [76](#)
  - num, [76](#)
  - urls, [76](#)
- fipa\_url\_sequence\_t
  - fipa\_acl.h, [298](#)
- fipa\_url\_t
  - fipa\_acl.h, [299](#)
- fipa\_word\_Compose
  - fipa\_acl.c, [237](#)
  - fipa\_acl.h, [309](#)
- fipa\_word\_Copy
  - fipa\_acl.c, [237](#)
  - fipa\_acl.h, [309](#)
- fipa\_word\_Destroy
  - fipa\_acl.c, [237](#)
  - fipa\_acl.h, [309](#)
- fipa\_word\_New
  - fipa\_acl.c, [237](#)
  - fipa\_acl.h, [309](#)
- fipa\_word\_p
  - fipa\_acl.h, [299](#)
- fipa\_word\_Parse
  - fipa\_acl.c, [237](#)
  - fipa\_acl.h, [309](#)
- fipa\_word\_s, [77](#)
  - content, [77](#)
- fipa\_word\_t
  - fipa\_acl.h, [299](#)
- foo
  - class.cxx, [494](#)
  - foo\_c, [79](#)

- foo\_s, 82
- FOO\_BLUE
  - enum.cxx, 495
- FOO\_ONE
  - enum.cxx, 495
- FOO\_RED
  - enum.cxx, 495
- FOO\_TWO
  - enum.cxx, 495
- foo\_c, 78
  - ~foo\_c, 78
  - bar, 79
  - barfoo, 80
  - foo, 79
  - foo\_c, 78
  - foo\_c, 78
  - get\_bar, 78
  - get\_barfoo, 79
  - get\_foo, 78
  - get\_global, 79
  - global, 79
  - set\_bar, 79
  - set\_foo, 79
  - setfoobar, 79
- foo\_default\_int
  - function.cxx, 496
- foo\_default\_string
  - function.cxx, 496
- foo\_enum\_e
  - enum.cxx, 495
- foo\_enum\_t
  - enum.cxx, 495
- foo\_float\_function
  - function.cxx, 496
- foo\_s, 81
  - ~foo\_s, 81
  - bar, 82
  - foo, 82
  - foo\_s, 81
  - foo\_s, 81
  - get\_bar, 81
  - get\_foo, 81
  - set\_bar, 81
  - set\_foo, 82
- foo\_t
  - struct.cxx, 497
- foo\_void\_function
  - function.cxx, 496
- FORBIDDEN
  - mtp\_http.h, 378
- FREEMEM
  - fipa\_acl.c, 227
- from
  - fipa\_acl\_Param\_s, 64

- from\_address
  - message\_s, 155
- FSb
  - aes.c, 633
- FT0
  - aes.c, 633
- FT1
  - aes.c, 633
- FT2
  - aes.c, 633
- FT3
  - aes.c, 633
- function.cxx
  - foo\_default\_int, 496
  - foo\_default\_string, 496
  - foo\_float\_function, 496
  - foo\_void\_function, 496
- G
  - dhm\_context, 55
- g\_mc\_platform
  - libmc.c, 430
- GENERATOR
  - dh\_genprime.c, 772
- get\_bar
  - foo\_c, 78
  - foo\_s, 81
- get\_barfoo
  - foo\_c, 79
- get\_foo
  - foo\_c, 78
  - foo\_s, 81
- get\_global
  - foo\_c, 79
- GET\_REQUEST
  - ssl\_client1.c, 779
  - ssl\_client2.c, 781
- GET\_THREAD\_MODE
  - macros.h, 361
- GET\_UINT32\_BE
  - aes.c, 630
  - des.c, 661
  - sha1.c, 712
  - sha2.c, 719
- GET\_UINT32\_LE
  - md4.c, 689
  - md5.c, 695
- GET\_UINT64\_BE
  - sha4.c, 726
- GetAgentExecEngine
  - LibMC::MCAgent, 144
- GetAgentFields
  - LibMC::MCAgent, 141
- GetAgentReturnData

- LibMC::MCAgent, 145
- GetAgentXMLString
  - LibMC::MCAgent, 142
- giant
  - mc\_platform\_s, 95
- giant\_cond
  - mc\_platform\_s, 95
- giant\_lock
  - mc\_platform\_s, 95
  - syncList\_s, 196
- global
  - foo\_c, 79
- GONE
  - mtp\_http.h, 378
- GX
  - dhm\_context, 56
- GY
  - dhm\_context, 56
- HaltAgency
  - LibMC::MCAgency, 113
- Halted
  - LibMC::MCAgency, 109
- handler\_HELP
  - cmd\_prompt.c, 216
- handler\_PRINT\_CONNECTLIST
  - cmd\_prompt.c, 216
- handler\_PRINTLIST\_AGENTS
  - cmd\_prompt.c, 216
- handler\_PRINTLIST\_MESSAGE
  - cmd\_prompt.c, 216
- handler\_QUIT
  - cmd\_prompt.c, 216
- handler\_SEND
  - cmd\_prompt.c, 216
- hardclock
  - timing.c, 750
  - timing.h, 613
- hardclock\_init
  - timing.c, 750
- HAVE\_STRDUP
  - config.h, 453
  - vcnet/config.h, 454
  - vcnet2005/config.h, 455
- HAVE\_VSNPRINTF
  - config.h, 453
- havege.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, 680
  - \_havege\_src, 682
  - havege\_fill, 681
  - havege\_init, 681
  - havege\_rand, 681
  - ONE\_ITERATION, 680
  - SWAP, 680
  - TST1\_ENTER, 680
  - TST1\_LEAVE, 681
  - TST2\_ENTER, 681
  - TST2\_LEAVE, 681
- havege.h
  - COLLECT\_SIZE, 547
  - havege\_init, 547
  - havege\_rand, 547
- havege\_fill
  - havege.c, 681
- havege\_init
  - havege.c, 681
  - havege.h, 547
- havege\_rand
  - havege.c, 681
  - havege.h, 547
- havege\_state, 83
  - offset, 83
  - pool, 83
  - PT1, 83
  - PT2, 83
  - WALK, 83
- hello.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, 763
  - main, 763
- help
  - commands.x.h, 279
- HERE
  - df.h, 287
- home
  - agent\_s, 39
- home\_port
  - agent\_s, 39
- host
  - mtp\_http\_s, 160
- host\_id.h
  - host\_id\_Compare, 318
  - host\_id\_Destroy, 318
  - host\_id\_Initialize, 318
  - host\_id\_p, 318
  - host\_id\_t, 318
- host\_id\_Compare
  - host\_id.h, 318
- host\_id\_Destroy
  - host\_id.h, 318
- host\_id\_Initialize
  - host\_id.h, 318
- host\_id\_p
  - host\_id.h, 318
- host\_id\_s, 85
  - hostname, 85
  - port, 85
- host\_id\_t
  - host\_id.h, 318

- HOST\_NAME\_MAX
  - libmc.c, 397
- hostName
  - agency\_s, 31
  - LibMC::MCAgency::MCAgency\_t, 133
- hostname
  - host\_id\_s, 85
  - mc\_platform\_s, 91
- hour
  - \_x509\_time, 28
  - fipa\_DateTime\_s, 68
- hr\_time, 86
  - opaque, 86
- hs\_md5
  - ssl\_context, 191
- hs\_sha1
  - ssl\_context, 192
- HTTP\_CONNECT
  - mtp\_http.h, 378
- HTTP\_DELETE
  - mtp\_http.h, 378
- HTTP\_GET
  - mtp\_http.h, 377
- HTTP\_HEAD
  - mtp\_http.h, 377
- HTTP\_NUM\_PERFORMATIVES
  - mtp\_http.h, 378
- HTTP\_OPTIONS
  - mtp\_http.h, 378
- HTTP\_PERFORMATIVE\_UNDEF
  - mtp\_http.h, 377
- HTTP\_PERFORMATIVE\_ZERO
  - mtp\_http.h, 377
- HTTP\_POST
  - mtp\_http.h, 377
- HTTP\_PUT
  - mtp\_http.h, 377
- HTTP\_RESPONSE
  - mtp\_http.h, 378
- HTTP\_TRACE
  - mtp\_http.h, 378
- http\_GetExpression
  - mtp\_http.c, 450
  - mtp\_http.h, 379
- http\_GetToken
  - mtp\_http.c, 450
  - mtp\_http.h, 379
- HTTP\_PARSE\_EXPR
  - mtp\_http.c, 449
- http\_ParseExpression
  - mtp\_http.c, 450
  - mtp\_http.h, 379
- http\_ParseHeader
  - mtp\_http.c, 451
  - mtp\_http.h, 380
- http\_performative
  - mtp\_http\_s, 160
- http\_performative\_e
  - mtp\_http.h, 377
- HTTP\_RESPONSE
  - ssl\_server.c, 783
- http\_status\_code
  - mtp\_http\_s, 160
- http\_status\_code\_e
  - mtp\_http.h, 378
- http\_to\_hostport
  - message.c, 446
  - message.h, 374
- http\_type
  - message\_s, 155
- http\_version
  - mtp\_http\_s, 160
- id
  - agent\_s, 38
  - barrier\_node\_s, 47
  - LibMC::MCAgent, 145
  - syncListNode\_s, 197
- in\_ctr
  - ssl\_context, 187
- in\_hdr
  - ssl\_context, 187
- in\_hslen
  - ssl\_context, 188
- in\_left
  - ssl\_context, 188
- in\_msg
  - ssl\_context, 187
- in\_msglen
  - ssl\_context, 188
- in\_msgtype
  - ssl\_context, 187
- in\_offt
  - ssl\_context, 187
- in\_reply\_to
  - fipa\_acl\_message\_s, 63
- index
  - command\_s, 50
- index\_compare
  - mxml-index.c, 467
- index\_find
  - mxml-index.c, 467
- index\_sort
  - mxml-index.c, 467
- init\_agent\_status
  - agent\_datastate\_s, 35
  - agent\_task\_s, 43
- Initialize

- LibMC::MCAgency, [111](#)
- Initialized
  - LibMC::MCAgency, [109](#)
- INSERT
  - df.h, [287](#)
- int
  - data\_structures.h, [285](#)
- integer
  - mxml\_value\_u, [172](#)
- intended\_receiver
  - fipa\_acl\_Param\_s, [65](#)
- interp\_options
  - mc\_platform\_s, [94](#)
- interpreter\_variable\_data.h
  - interpreter\_variable\_data\_Copy, [319](#)
  - interpreter\_variable\_data\_Destroy, [319](#)
  - interpreter\_variable\_data\_Initialize, [320](#)
  - interpreter\_variable\_data\_-
    - InitializeFromAgent, [320](#)
  - interpreter\_variable\_data\_New, [320](#)
  - interpreter\_variable\_data\_p, [319](#)
  - interpreter\_variable\_data\_t, [319](#)
- interpreter\_variable\_data\_Copy
  - agent\_return\_data.c, [208](#)
  - interpreter\_variable\_data.h, [319](#)
- interpreter\_variable\_data\_Destroy
  - agent\_return\_data.c, [208](#)
  - interpreter\_variable\_data.h, [319](#)
- interpreter\_variable\_data\_Initialize
  - agent\_return\_data.c, [208](#)
  - interpreter\_variable\_data.h, [320](#)
- interpreter\_variable\_data\_InitializeFromAgent
  - agent\_return\_data.c, [208](#)
  - interpreter\_variable\_data.h, [320](#)
- interpreter\_variable\_data\_New
  - agent\_return\_data.c, [209](#)
  - interpreter\_variable\_data.h, [320](#)
- interpreter\_variable\_data\_p
  - interpreter\_variable\_data.h, [319](#)
- interpreter\_variable\_data\_s, [87](#)
  - array\_dim, [87](#)
  - array\_extent, [87](#)
  - data, [88](#)
  - data\_type, [87](#)
  - name, [87](#)
  - size, [87](#)
- interpreter\_variable\_data\_t
  - interpreter\_variable\_data.h, [319](#)
- ipad
  - md2\_context, [148](#)
  - md4\_context, [150](#)
  - md5\_context, [152](#)
  - sha1\_context, [179](#)
  - sha2\_context, [181](#)
  - sha4\_context, [183](#)
- is
  - commands.x.h, [279](#)
- is224
  - sha2\_context, [182](#)
- is384
  - sha4\_context, [184](#)
- is\_utc
  - fipa\_DateTime\_s, [69](#)
- isHTTP
  - message\_s, [154](#)
- issuer
  - \_x509\_cert, [24](#)
- issuer\_id
  - \_x509\_cert, [25](#)
- issuer\_raw
  - \_x509\_cert, [24](#)
- iv\_dec
  - ssl\_context, [194](#)
- iv\_enc
  - ssl\_context, [194](#)
- ivlen
  - ssl\_context, [194](#)
- K
  - dhm\_context, [56](#)
  - sha1.c, [712](#)
  - sha4.c, [730](#)
- KEY\_LEN
  - rsa.h, [570](#)
- KEY\_SIZE
  - rsa\_genkey.c, [776](#)
- keylen
  - ssl\_context, [193](#)
- KT0
  - aes.c, [633](#)
- KT1
  - aes.c, [633](#)
- KT2
  - aes.c, [634](#)
- KT3
  - aes.c, [634](#)
- language
  - fipa\_acl\_message\_s, [62](#)
- last\_child
  - mxml\_node\_s, [169](#)
- last\_error
  - agency\_s, [33](#)
  - LibMC::MCAgency::MCAgency\_t, [134](#)
- left
  - md2\_context, [149](#)
- len
  - \_x509\_buf, [22](#)

- dhm\_context, 55
- dynstring\_s, 57
- rsa\_context, 174
- LENGTH\_REQUIRED
  - mtp\_http.h, 378
- LHs
  - des.c, 666
- LibMC, 13
- libmc.c
  - g\_mc\_platform, 430
  - HOST\_NAME\_MAX, 397
  - MC\_AclAddReceiver, 397
  - MC\_AclAddReceiver\_chdl, 397
  - MC\_AclAddReplyTo, 397
  - MC\_AclAddReplyTo\_chdl, 397
  - MC\_AclDestroy, 397
  - MC\_AclDestroy\_chdl, 398
  - MC\_AclNew, 398
  - MC\_AclNew\_chdl, 398
  - MC\_AclPost, 398
  - MC\_AclPost\_chdl, 399
  - MC\_AclReply, 399
  - MC\_AclReply\_chdl, 399
  - MC\_AclRetrieve, 399
  - MC\_AclRetrieve\_chdl, 400
  - MC\_AclSend, 400
  - MC\_AclSend\_chdl, 400
  - MC\_AclSetContent, 400
  - MC\_AclSetContent\_chdl, 401
  - MC\_AclSetPerformative, 401
  - MC\_AclSetPerformative\_chdl, 401
  - MC\_AclSetSender, 401
  - MC\_AclSetSender\_chdl, 401
  - MC\_AclWaitRetrieve, 401
  - MC\_AclWaitRetrieve\_chdl, 402
  - MC\_AddAgent, 402
  - MC\_AddAgent\_chdl, 402
  - MC\_AgentVariableRetrieve, 402
  - MC\_AgentVariableRetrieve\_chdl, 403
  - MC\_AgentVariableSave, 403
  - MC\_AgentVariableSave\_chdl, 404
  - MC\_Barrier, 404
  - MC\_Barrier\_chdl, 404
  - MC\_BarrierDelete, 404
  - MC\_BarrierDelete\_chdl, 404
  - MC\_BarrierInit, 405
  - MC\_BarrierInit\_chdl, 405
  - MC\_CallAgentFunc, 405
  - MC\_CallAgentFunc\_chdl, 406
  - MC\_ChInitializeOptions, 406
  - MC\_ComposeAgent, 406
  - MC\_ComposeAgent\_chdl, 407
  - MC\_CondBroadcast, 407
  - MC\_CondBroadcast\_chdl, 407
  - MC\_CondReset, 407
  - MC\_CondReset\_chdl, 408
  - MC\_CondSignal, 408
  - MC\_CondSignal\_chdl, 409
  - MC\_CondWait, 409
  - MC\_CondWait\_chdl, 409
  - MC\_CopyAgent, 409
  - MC\_DeleteAgent, 410
  - MC\_DeleteAgent\_chdl, 410
  - MC\_DeregisterService, 410
  - MC\_DeregisterService\_chdl, 410
  - MC\_DestroyServiceSearchResult, 411
  - MC\_DestroyServiceSearchResult\_chdl, 411
  - MC\_End, 411
  - MC\_End\_chdl, 412
  - MC\_FindAgentByID, 412
  - MC\_FindAgentByID\_chdl, 412
  - MC\_FindAgentByName, 412
  - MC\_FindAgentByName\_chdl, 413
  - MC\_GetAgentArrivalTime, 413
  - MC\_GetAgentArrivalTime\_chdl, 413
  - MC\_GetAgentExecEngine, 413
  - MC\_GetAgentID, 413
  - MC\_GetAgentID\_chdl, 414
  - MC\_GetAgentName, 414
  - MC\_GetAgentName\_chdl, 414
  - MC\_GetAgentNumTasks, 414
  - MC\_GetAgentNumTasks\_chdl, 414
  - MC\_GetAgentReturnData, 414
  - MC\_GetAgentStatus, 415
  - MC\_GetAgentStatus\_chdl, 415
  - MC\_GetAgentType, 415
  - MC\_GetAgentXMLString, 415
  - MC\_GetAgentXMLString\_chdl, 416
  - MC\_GetAllAgents, 416
  - MC\_GetTimeOfDay\_chdl, 416
  - MC\_HaltAgency, 416
  - MC\_HaltAgency\_chdl, 416
  - MC\_Initialize, 417
  - MC\_InitializeAgencyOptions, 417
  - MC\_LoadAgentFromFile, 417
  - MC\_MainLoop, 418
  - MC\_MutexLock, 418
  - MC\_MutexLock\_chdl, 418
  - MC\_MutexUnlock, 419
  - MC\_MutexUnlock\_chdl, 419
  - MC\_PrintAgentCode, 419
  - MC\_PrintAgentCode\_chdl, 419
  - MC\_RegisterService, 419
  - MC\_RegisterService\_chdl, 420
  - MC\_ResetSignal, 420
  - MC\_ResumeAgency, 420
  - MC\_ResumeAgency\_chdl, 421
  - MC\_RetrieveAgent, 421

- MC\_RetrieveAgent\_chdl, [421](#)
- MC\_RetrieveAgentCode, [421](#)
- MC\_RetrieveAgentCode\_chdl, [421](#)
- MC\_SearchForService, [421](#)
- MC\_SearchForService\_chdl, [422](#)
- MC\_SemaphorePost, [422](#)
- MC\_SemaphorePost\_chdl, [422](#)
- MC\_SemaphoreWait, [423](#)
- MC\_SemaphoreWait\_chdl, [423](#)
- MC\_SendAgentMigrationMessage, [423](#)
- MC\_SendAgentMigrationMessage\_chdl, [423](#)
- MC\_SendAgentMigrationMessageFile, [424](#)
- MC\_SendAgentMigrationMessageFile\_chdl, [424](#)
- MC\_SendCh, [424](#)
- MC\_SendSteerCommand, [424](#)
- MC\_SendSteerCommand\_chdl, [424](#)
- MC\_SetAgentStatus, [424](#)
- MC\_SetAgentStatus\_chdl, [425](#)
- MC\_SetDefaultAgentStatus, [425](#)
- MC\_SetDefaultAgentStatus\_chdl, [425](#)
- MC\_SetThreadOff, [425](#)
- MC\_SetThreadOn, [426](#)
- MC\_SetThreadsAllOff, [426](#)
- MC\_SetThreadsAllOn, [426](#)
- MC\_Steer, [427](#)
- MC\_SteerControl, [427](#)
- MC\_SyncDelete, [428](#)
- MC\_SyncDelete\_chdl, [428](#)
- MC\_SyncInit, [428](#)
- MC\_SyncInit\_chdl, [429](#)
- MC\_TerminateAgent, [429](#)
- MC\_TerminateAgent\_chdl, [429](#)
- MC\_WaitAgent, [429](#)
- MC\_WaitRetrieveAgent, [429](#)
- MC\_WaitSignal, [430](#)
- libmc.h
  - agency\_p, [327](#)
  - agency\_t, [327](#)
  - agent\_p, [327](#)
  - agent\_t, [327](#)
  - ERR, [328](#)
  - error\_code\_e, [327](#)
  - error\_code\_t, [327](#)
  - EXPORTMC, [326](#)
  - MC\_AGENT\_ACTIVE, [328](#)
  - MC\_AGENT\_NEUTRAL, [328](#)
  - MC\_AGENT\_SUSPENDED, [328](#)
  - MC\_ALL\_SIGNALS, [329](#)
  - MC\_ERR, [327](#)
  - MC\_ERR\_CONNECT, [327](#), [328](#)
  - MC\_ERR\_EMPTY, [327](#), [328](#)
  - MC\_ERR\_INVALID, [327](#), [328](#)
  - MC\_ERR\_INVALID\_ARGS, [327](#), [328](#)
  - MC\_ERR\_MEMORY, [327](#), [328](#)
  - MC\_ERR\_NOT\_FOUND, [327](#), [328](#)
  - MC\_ERR\_PARSE, [327](#), [328](#)
  - MC\_ERR\_SEND, [327](#), [328](#)
  - MC\_EXEC\_AGENT, [329](#)
  - MC\_LOCAL\_AGENT, [328](#)
  - MC\_NO\_SIGNAL, [329](#)
  - MC\_NO\_STATUS, [328](#)
  - MC\_NONE, [328](#)
  - MC\_RECV\_AGENT, [329](#)
  - MC\_RECV\_CONNECTION, [329](#)
  - MC\_RECV\_MESSAGE, [329](#)
  - MC\_RECV\_RETURN, [329](#)
  - MC\_REMOTE\_AGENT, [328](#)
  - MC\_RESTART, [329](#)
  - MC\_RETURN\_AGENT, [328](#)
  - MC\_RUN, [329](#)
  - MC\_STOP, [329](#)
  - MC\_SUCCESS, [327](#)
  - MC\_SUSPEND, [329](#)
  - MC\_THREAD\_ACC, [329](#)
  - MC\_THREAD\_AGENT, [329](#)
  - MC\_THREAD\_ALL, [329](#)
  - MC\_THREAD\_AMS, [329](#)
  - MC\_THREAD\_CP, [329](#)
  - MC\_THREAD\_DF, [329](#)
  - MC\_WAIT\_CH, [328](#)
  - MC\_WAIT\_FINISHED, [328](#)
  - MC\_WAIT\_MESSGSEND, [328](#)
  - MC\_WARN\_DUPLICATE, [327](#), [328](#)
  - MC\_AclAddReceiver, [330](#)
  - MC\_AclAddReplyTo, [330](#)
  - MC\_AclDestroy, [330](#)
  - MC\_AclNew, [330](#)
  - MC\_AclPost, [330](#)
  - MC\_AclReply, [331](#)
  - MC\_AclRetrieve, [331](#)
  - MC\_AclSend, [331](#)
  - MC\_AclSetContent, [332](#)
  - MC\_AclSetPerformative, [332](#)
  - MC\_AclSetSender, [332](#)
  - MC\_AclWaitRetrieve, [332](#)
  - MC\_AddAgent, [333](#)
  - MC\_AgentStatus\_e, [328](#)
  - MC\_AgentType\_e, [328](#)
  - MC\_AgentVariableRetrieve, [333](#)
  - MC\_AgentVariableSave, [334](#)
  - MC\_BarrierDelete, [334](#)
  - MC\_BarrierInit, [335](#)
  - MC\_CallAgentFunc, [335](#)
  - MC\_ChInitializeOptions, [336](#)
  - MC\_ComposeAgent, [336](#)
  - MC\_CondBroadcast, [337](#)
  - MC\_CondReset, [337](#)

- MC\_CondSignal, 337
- MC\_CondWait, 338
- MC\_CopyAgent, 339
- MC\_DeleteAgent, 339
- MC\_DestroyServiceSearchResult, 339
- MC\_End, 340
- MC\_FindAgentByID, 340
- MC\_FindAgentByName, 340
- MC\_GetAgentExecEngine, 341
- MC\_GetAgentID, 341
- MC\_GetAgentName, 341
- MC\_GetAgentNumTasks, 341
- MC\_GetAgentReturnData, 342
- MC\_GetAgentStatus, 342
- MC\_GetAgentType, 342
- MC\_GetAgentXMLString, 343
- MC\_HaltAgency, 343
- MC\_Initialize, 343
- MC\_InitializeAgencyOptions, 344
- MC\_LoadAgentFromFile, 344
- MC\_MainLoop, 344
- MC\_MutexLock, 345
- MC\_MutexUnlock, 345
- MC\_PrintAgentCode, 345
- MC\_RegisterService, 346
- MC\_ResetSignal, 346
- MC\_ResumeAgency, 346
- MC\_RetrieveAgent, 347
- MC\_RetrieveAgentCode, 347
- MC\_SearchForService, 347
- MC\_SemaphorePost, 348
- MC\_SemaphoreWait, 348
- MC\_SendAgentMigrationMessage, 349
- MC\_SendAgentMigrationMessageFile, 349
- MC\_SetAgentStatus, 349
- MC\_SetDefaultAgentStatus, 350
- MC\_SetThreadOff, 350
- MC\_SetThreadOn, 350
- MC\_SetThreadsAllOff, 351
- MC\_SetThreadsAllOn, 351
- MC\_Signal\_e, 328
- MC\_Steer, 351
- MC\_SteerCommand\_e, 329
- MC\_SteerCommand\_t, 327
- MC\_SteerControl, 352
- MC\_SyncDelete, 352
- MC\_SyncInit, 353
- MC\_TerminateAgent, 353
- MC\_ThreadIndex\_e, 329
- MC\_Wait, 326
- MC\_WaitAgent, 353
- MC\_WaitRetrieveAgent, 354
- MC\_WaitSignal, 354
- MCAgency\_t, 327
- MCAgencyOptions\_t, 327
- MCAgent\_t, 327
- LibMC::MCAclMessage, 98
  - AclMsg, 103
  - aclmsg\_p, 103
  - AddReceiver, 101
  - AddReplyTo, 101
  - Destroy, 102
  - FIPA\_ACCEPT\_PROPOSAL, 99
  - FIPA\_AGREE, 99
  - FIPA\_CALL\_FOR\_PROPOSAL, 99
  - FIPA\_CANCEL, 99
  - FIPA\_CONFIRM, 99
  - FIPA\_DISCONFIRM, 99
  - FIPA\_ERROR, 99
  - FIPA\_FAILURE, 99
  - FIPA\_INFORM, 99
  - FIPA\_INFORM\_IF, 99
  - FIPA\_INFORM\_REF, 99
  - FIPA\_NOT\_UNDERSTOOD, 99
  - FIPA\_PROPOGATE, 99
  - FIPA\_PROPOSE, 99
  - FIPA\_PROXY, 99
  - FIPA\_QUERY\_IF, 99
  - FIPA\_QUERY\_REF, 99
  - FIPA\_REFUSE, 99
  - FIPA\_REJECT\_PROPOSAL, 99
  - FIPA\_REQUEST, 99
  - FIPA\_REQUEST\_WHEN, 99
  - FIPA\_REQUEST\_WHENEVER, 99
  - FIPA\_SUBSCRIBE, 99
  - FIPA\_ZERO, 99
  - MC\_FipaPerformative\_e, 99
  - MCAclMessage, 100
  - New, 100
  - Reply, 100
  - SetContent, 102
  - SetPerformative, 100
  - SetSender, 101
- LibMC::MCAgency, 104
  - \_MC\_AclAddReceiver, 129
  - \_MC\_AclAddReplyTo, 129
  - \_MC\_AclDestroy, 129
  - \_MC\_AclNew, 129
  - \_MC\_AclPost, 129
  - \_MC\_AclReply, 129
  - \_MC\_AclRetrieve, 129
  - \_MC\_AclSend, 129
  - \_MC\_AclSetContent, 129
  - \_MC\_AclSetPerformative, 129
  - \_MC\_AclSetSender, 129
  - \_MC\_AclWaitRetrieve, 129
  - \_MC\_AddAgent, 127
  - \_MC\_BarrierDelete, 126



- [\\_MC\\_BarrierInit, 126](#)
- [\\_MC\\_CallAgentFunc, 129](#)
- [\\_MC\\_ChInitializeOptions, 124](#)
- [\\_MC\\_CondBroadcast, 125](#)
- [\\_MC\\_CondReset, 125](#)
- [\\_MC\\_CondSignal, 125](#)
- [\\_MC\\_CondWait, 125](#)
- [\\_MC\\_DeleteAgent, 127](#)
- [\\_MC\\_End, 124](#)
- [\\_MC\\_FindAgentByID, 127](#)
- [\\_MC\\_FindAgentByName, 127](#)
- [\\_MC\\_GetAgentExecEngine, 129](#)
- [\\_MC\\_GetAgentID, 127](#)
- [\\_MC\\_GetAgentName, 129](#)
- [\\_MC\\_GetAgentNumTasks, 129](#)
- [\\_MC\\_GetAgentReturnData, 129](#)
- [\\_MC\\_GetAgentStatus, 129](#)
- [\\_MC\\_GetAgentType, 129](#)
- [\\_MC\\_GetAgentXMLString, 129](#)
- [\\_MC\\_HaltAgency, 124](#)
- [\\_MC\\_Initialize, 124](#)
- [\\_MC\\_InitializeAgencyOptions, 124](#)
- [\\_MC\\_MainLoop, 125](#)
- [\\_MC\\_MutexLock, 125](#)
- [\\_MC\\_MutexUnlock, 125](#)
- [\\_MC\\_PrintAgentCode, 129](#)
- [\\_MC\\_RegisterService, 127](#)
- [\\_MC\\_ResetSignal, 126](#)
- [\\_MC\\_ResumeAgency, 124](#)
- [\\_MC\\_RetrieveAgent, 127](#)
- [\\_MC\\_RetrieveAgentCode, 129](#)
- [\\_MC\\_SearchForService, 127](#)
- [\\_MC\\_SemaphorePost, 126](#)
- [\\_MC\\_SemaphoreWait, 126](#)
- [\\_MC\\_SendAgentMigrationMessage, 125](#)
- [\\_MC\\_SendAgentMigrationMessageFile, 125](#)
- [\\_MC\\_SetAgentStatus, 129](#)
- [\\_MC\\_SetDefaultAgentStatus, 124](#)
- [\\_MC\\_SetThreadOff, 124](#)
- [\\_MC\\_SetThreadOn, 124](#)
- [\\_MC\\_SetThreadsAllOff, 124](#)
- [\\_MC\\_SetThreadsAllOn, 124](#)
- [\\_MC\\_Steer, 126](#)
- [\\_MC\\_SteerControl, 126](#)
- [\\_MC\\_SyncDelete, 126](#)
- [\\_MC\\_SyncInit, 126](#)
- [\\_MC\\_TerminateAgent, 129](#)
- [\\_MC\\_WaitAgent, 125](#)
- [\\_MC\\_WaitRetrieveAgent, 125](#)
- [\\_MC\\_WaitSignal, 126](#)
- [AclSend, 123](#)
- [AddAgent, 122](#)
- [Agency, 130](#)
- [agency\\_p, 129](#)
- [BarrierDelete, 119](#)
- [BarrierInit, 120](#)
- [CH\\_REGULARCH, 109](#)
- [CH\\_SAFECH, 109](#)
- [ChInitializeOptions, 111](#)
- [ChShellType, 109](#)
- [CondBroadcast, 116](#)
- [CondReset, 116](#)
- [CondSignal, 116](#)
- [CondWait, 116](#)
- [End, 111](#)
- [Ended, 109](#)
- [FindAgentByID, 122](#)
- [FindAgentByName, 122](#)
- [HaltAgency, 113](#)
- [Halted, 109](#)
- [Initialize, 111](#)
- [Initialized, 109](#)
- [LoadAgentMigrationMessageFile, 115](#)
- [MainLoop, 123](#)
- [MC\\_ALL\\_SIGNALS, 110](#)
- [MC\\_EXEC\\_AGENT, 110](#)
- [MC\\_NO\\_SIGNAL, 110](#)
- [MC\\_RECV\\_AGENT, 110](#)
- [MC\\_RECV\\_CONNECTION, 110](#)
- [MC\\_RECV\\_MESSAGE, 110](#)
- [MC\\_RECV\\_RETURN, 110](#)
- [MC\\_RESTART, 110](#)
- [MC\\_RUN, 110](#)
- [MC\\_STOP, 110](#)
- [MC\\_SUSPEND, 110](#)
- [MC\\_THREAD\\_ACC, 110](#)
- [MC\\_THREAD\\_AGENT, 110](#)
- [MC\\_THREAD\\_ALL, 110](#)
- [MC\\_THREAD\\_AMS, 109](#)
- [MC\\_THREAD\\_CP, 110](#)
- [MC\\_THREAD\\_DF, 109](#)
- [MC\\_Signal\\_e, 110](#)
- [MC\\_SteerCommand\\_e, 110](#)
- [MC\\_ThreadIndex\\_e, 109](#)
- [MCAgency, 111](#)
- [MCAgencyState, 109](#)
- [mcdll, 130](#)
- [MutexLock, 117](#)
- [MutexUnlock, 117](#)
- [NoState, 109](#)
- [options, 129](#)
- [Port, 130](#)
- [port, 130](#)
- [RegisterService, 121](#)
- [ResetSignal, 118](#)
- [ResumeAgency, 113](#)
- [RetrieveAgent, 123](#)
- [Running, 109](#)

- SearchForService, 121
- SemaphorePost, 117
- SemaphoreWait, 118
- SendAgentMigrationMessage, 115
- SendAgentMigrationMessageFile, 114
- SetDefaultAgentStatus, 114
- SetThreadOff, 113
- SetThreadOn, 112
- SetThreadsAllOff, 112
- SetThreadsAllOn, 112
- State, 131
- state, 130
- Steer, 120
- SteerControl, 120
- SyncDelete, 118
- SyncInit, 119
- WaitAgent, 114
- WaitRetrieveAgent, 114
- WaitSignal, 119
- LibMC::MCAgency::ChOptions\_t, 132
  - chhome, 132
  - shelltype, 132
- LibMC::MCAgency::MCAgency\_t, 133
  - client, 133
  - default\_agentstatus, 134
  - enable\_security, 134
  - filename, 133
  - hostName, 133
  - last\_error, 134
  - mc\_platform, 134
  - portno, 133
  - portnoc, 133
  - server, 133
  - stack\_size, 134
  - threads, 134
- LibMC::MCAgency::MCAgencyOptions\_t, 135
  - default\_agent\_status, 135
  - enable\_security, 135
  - modified, 135
  - stack\_size, 135
  - threads, 135
- LibMC::MCAgent, 138
  - AcIPost, 143
  - AcIRetrieve, 143
  - AcIWaitRetrieve, 143
  - Agent, 146
  - agent\_p, 145
  - AgentID, 146
  - AgentName, 146
  - AgentNumTasks, 147
  - AgentStatus, 147
  - AgentType, 147
  - CallAgentFunc, 144
  - DeleteAgent, 141
  - GetAgentExecEngine, 144
  - GetAgentFields, 141
  - GetAgentReturnData, 145
  - GetAgentXMLString, 142
  - id, 145
  - MC\_AGENT\_ACTIVE, 140
  - MC\_AGENT\_NEUTRAL, 140
  - MC\_AGENT\_SUSPENDED, 140
  - MC\_LOCAL\_AGENT, 140
  - MC\_NO\_STATUS, 140
  - MC\_NONE, 140
  - MC\_REMOTE\_AGENT, 140
  - MC\_RETURN\_AGENT, 140
  - MC\_WAIT\_CH, 140
  - MC\_WAIT\_FINISHED, 140
  - MC\_WAIT\_MESSGSEND, 140
  - MC\_AgentStatus\_e, 140
  - MC\_AgentType\_e, 140
  - MCAgent, 141
  - name, 145
  - numTasks, 145
  - operator IntPtr, 141
  - operator MCAgent, 141
  - PrintAgentCode, 142
  - RetrieveAgentCode, 142
  - status, 146
  - TerminateAgent, 142
  - ToString, 141
  - type, 146
- LibMC::Properties, 14
- LibMC::Properties::Settings, 177
  - Default, 178
  - defaultInstance, 178
  - SettingChangingEventHandler, 177
  - Settings, 177
  - SettingsSavingEventHandler, 177
- list
  - barrier\_queue\_s, 48
  - syncList\_s, 196
- list.c
  - list\_pGetSize, 431
  - ListAdd, 431
  - ListDelete, 431
  - ListGetHead, 431
  - ListInitialize, 431
  - ListInsert, 432
  - ListPop, 432
  - ListSearch, 432
  - ListTerminate, 432
- list.h
  - DATA, 433
  - list\_p, 434
  - list\_t, 434
  - ListAdd, 434

- ListDelete, 434
- ListGetHead, 434
- ListGetSize, 434
- ListInitialize, 434
- ListInsert, 434
- listNode\_p, 434
- listNode\_t, 434
- ListPop, 434
- ListSearch, 435
- ListTerminate, 435
- QUEUE\_TEMPLATE, 433
- list\_p
  - list.h, 434
- list\_pGetSize
  - list.c, 431
- list\_s, 89
  - listhead, 89
  - size, 89
- list\_t
  - list.h, 434
- ListAdd
  - list.c, 431
  - list.h, 434
- ListDelete
  - list.c, 431
  - list.h, 434
- listen\_Thread
  - acc.c, 201
  - acc.h, 247
- listen\_thread\_arg
  - mc\_platform\_s, 93
- ListGetHead
  - list.c, 431
  - list.h, 434
- ListGetSize
  - list.h, 434
- listhead
  - list\_s, 89
- ListInitialize
  - list.c, 431
  - list.h, 434
- ListInsert
  - list.c, 432
  - list.h, 434
- listNode\_p
  - list.h, 434
- listNode\_s, 90
  - next, 90
  - node\_data, 90
- listNode\_t
  - list.h, 434
- ListPop
  - list.c, 432
  - list.h, 434
- ListSearch
  - list.c, 432
  - list.h, 435
- ListTerminate
  - list.c, 432
  - list.h, 435
- LoadAgentMigrationMessageFile
  - LibMC::MCAgency, 115
- lock
  - agent\_s, 41
  - barrier\_node\_s, 47
  - barrier\_queue\_s, 48
  - df.h, 290
  - mc\_rwlock\_s, 96
  - syncList\_s, 196
  - syncListNode\_s, 197
- LOCKED
  - mtp\_http.h, 379
- m
  - arc4\_context, 46
- mac\_dec
  - ssl\_context, 195
- mac\_enc
  - ssl\_context, 194
- maclen
  - ssl\_context, 194
- macros.h
  - CH\_DATATYPE\_SIZE, 357
  - CH\_DATATYPE\_STR\_TO\_VAL, 357
  - CH\_DATATYPE\_STRING, 358
  - CH\_DATATYPE\_VALUE\_STRING, 358
  - CH\_STRING\_DATATYPE, 359
  - CHECK\_NULL, 359
  - COND\_BROADCAST, 360
  - COND\_DESTROY, 360
  - COND\_INIT, 360
  - COND\_RESET, 360
  - COND\_SIGNAL, 360
  - COND\_SLEEP, 361
  - COND\_SLEEP\_ACTION, 361
  - COND\_T, 361
  - COND\_WAIT, 361
  - GET\_THREAD\_MODE, 361
  - MUTEX\_DESTROY, 361
  - MUTEX\_INIT, 362
  - MUTEX\_LOCK, 362
  - MUTEX\_NEW, 362
  - MUTEX\_T, 362
  - MUTEX\_UNLOCK, 363
  - PTHREAD\_STACK\_SIZE, 363
  - RWLOCK\_DESTROY, 363
  - RWLOCK\_INIT, 363
  - RWLOCK\_RDLOCK, 363

- RWLOCK\_RDUNLOCK, 363
- RWLOCK\_T, 363
- RWLOCK\_WRLOCK, 364
- RWLOCK\_WRUNLOCK, 364
- SEMAPHORE\_DESTROY, 364
- SEMAPHORE\_INIT, 364
- SEMAPHORE\_POST, 364
- SEMAPHORE\_T, 364
- SEMAPHORE\_WAIT, 364
- SET\_THREAD\_OFF, 364
- SET\_THREAD\_ON, 364
- SIGNAL, 365
- SLEEP\_QUEUE, 365
- SLEEP\_RESET, 365
- STRUCT, 365
- THREAD\_CANCEL, 365
- THREAD\_CREATE, 365
- THREAD\_JOIN, 366
- THREAD\_T, 366
- WAKE\_QUEUE, 366
- WARN, 366
- mail\_queue
  - agent\_mailbox\_s, 37
- mailbox
  - agent\_s, 41
- main
  - aescript2.c, 762
  - benchmark.c, 786
  - dh\_client.c, 770
  - dh\_genprime.c, 772
  - dh\_server.c, 773
  - hello.c, 763
  - md5sum.c, 764
  - mpi\_demo.c, 775
  - mxmldoc.c, 491
  - rsa\_genkey.c, 776
  - rsa\_sign.c, 777
  - rsa\_verify.c, 778
  - selftest.c, 788
  - sha1sum.c, 766
  - sha2sum.c, 768
  - ssl\_client1.c, 780
  - ssl\_client2.c, 782
  - ssl\_server.c, 784
  - testmxml.c, 498
- MainLoop
  - LibMC::MCAgency, 123
- major\_ver
  - ssl\_context, 186
- master
  - ssl\_context, 193
- max\_pathlen
  - \_x509\_cert, 25
- max\_ver
  - ssl\_context, 186
- MC\_AGENT\_ACTIVE
  - libmc.h, 328
  - LibMC::MCAgent, 140
- MC\_AGENT\_NEUTRAL
  - libmc.h, 328
  - LibMC::MCAgent, 140
- MC\_AGENT\_SUSPENDED
  - libmc.h, 328
  - LibMC::MCAgent, 140
- MC\_ALL\_SIGNALS
  - libmc.h, 329
  - LibMC::MCAgency, 110
- MC\_ERR
  - libmc.h, 327
  - mc\_error.h, 367
- MC\_ERR\_CONNECT
  - libmc.h, 327, 328
  - mc\_error.h, 367
- MC\_ERR\_EMPTY
  - libmc.h, 327, 328
  - mc\_error.h, 367
- MC\_ERR\_INVALID
  - libmc.h, 327, 328
  - mc\_error.h, 367
- MC\_ERR\_INVALID\_ARGS
  - libmc.h, 327, 328
  - mc\_error.h, 367, 368
- MC\_ERR\_MEMORY
  - libmc.h, 327, 328
  - mc\_error.h, 367, 368
- MC\_ERR\_NOT\_FOUND
  - libmc.h, 327, 328
  - mc\_error.h, 367, 368
- MC\_ERR\_PARSE
  - libmc.h, 327, 328
  - mc\_error.h, 367
- MC\_ERR\_SEND
  - libmc.h, 327, 328
  - mc\_error.h, 367, 368
- mc\_error.h
  - ERR, 367
  - MC\_ERR, 367
  - MC\_ERR\_CONNECT, 367
  - MC\_ERR\_EMPTY, 367
  - MC\_ERR\_INVALID, 367
  - MC\_ERR\_INVALID\_ARGS, 367, 368
  - MC\_ERR\_MEMORY, 367, 368
  - MC\_ERR\_NOT\_FOUND, 367, 368
  - MC\_ERR\_PARSE, 367
  - MC\_ERR\_SEND, 367, 368
  - MC\_SUCCESS, 367
  - MC\_WARN\_DUPLICATE, 367, 368
- MC\_EXEC\_AGENT

- libmc.h, 329
- LibMC::MCAgency, 110
- MC\_LOCAL\_AGENT
  - libmc.h, 328
  - LibMC::MCAgent, 140
- MC\_NO\_SIGNAL
  - libmc.h, 329
  - LibMC::MCAgency, 110
- MC\_NO\_STATUS
  - libmc.h, 328
  - LibMC::MCAgent, 140
- MC\_NONE
  - libmc.h, 328
  - LibMC::MCAgent, 140
- MC\_RECV\_AGENT
  - libmc.h, 329
  - LibMC::MCAgency, 110
- MC\_RECV\_CONNECTION
  - libmc.h, 329
  - LibMC::MCAgency, 110
- MC\_RECV\_MESSAGE
  - libmc.h, 329
  - LibMC::MCAgency, 110
- MC\_RECV\_RETURN
  - libmc.h, 329
  - LibMC::MCAgency, 110
- MC\_REMOTE\_AGENT
  - libmc.h, 328
  - LibMC::MCAgent, 140
- MC\_RESTART
  - libmc.h, 329
  - LibMC::MCAgency, 110
- MC\_RETURN\_AGENT
  - libmc.h, 328
  - LibMC::MCAgent, 140
- MC\_RUN
  - libmc.h, 329
  - LibMC::MCAgency, 110
- MC\_STOP
  - libmc.h, 329
  - LibMC::MCAgency, 110
- MC\_SUCCESS
  - libmc.h, 327
  - mc\_error.h, 367
- MC\_SUSPEND
  - libmc.h, 329
  - LibMC::MCAgency, 110
- MC\_THREAD\_ACC
  - libmc.h, 329
  - LibMC::MCAgency, 110
- MC\_THREAD\_AGENT
  - libmc.h, 329
  - LibMC::MCAgency, 110
- MC\_THREAD\_ALL
  - libmc.h, 329
  - LibMC::MCAgency, 110
- MC\_THREAD\_AMS
  - libmc.h, 329
  - LibMC::MCAgency, 109
- MC\_THREAD\_CP
  - libmc.h, 329
  - LibMC::MCAgency, 110
- MC\_THREAD\_DF
  - libmc.h, 329
  - LibMC::MCAgency, 109
- MC\_WAIT\_CH
  - libmc.h, 328
  - LibMC::MCAgent, 140
- MC\_WAIT\_FINISHED
  - libmc.h, 328
  - LibMC::MCAgent, 140
- MC\_WAIT\_MESSGSEND
  - libmc.h, 328
  - LibMC::MCAgent, 140
- MC\_WARN\_DUPLICATE
  - libmc.h, 327, 328
  - mc\_error.h, 367, 368
- MC\_AclAddReceiver
  - libmc.c, 397
  - libmc.h, 330
- MC\_AclAddReceiver\_chdl
  - agent\_lib.h, 255
  - libmc.c, 397
- MC\_AclAddReplyTo
  - libmc.c, 397
  - libmc.h, 330
- MC\_AclAddReplyTo\_chdl
  - agent\_lib.h, 255
  - libmc.c, 397
- MC\_AclDestroy
  - libmc.c, 397
  - libmc.h, 330
- MC\_AclDestroy\_chdl
  - agent\_lib.h, 255
  - libmc.c, 398
- MC\_AclNew
  - libmc.c, 398
  - libmc.h, 330
- MC\_AclNew\_chdl
  - agent\_lib.h, 255
  - libmc.c, 398
- MC\_AclPost
  - libmc.c, 398
  - libmc.h, 330
- MC\_AclPost\_chdl
  - agent\_lib.h, 255
  - libmc.c, 399
- MC\_AclReply

- libmc.c, 399
- libmc.h, 331
- MC\_AclReply\_chdl
  - agent\_lib.h, 255
  - libmc.c, 399
- MC\_AclRetrieve
  - libmc.c, 399
  - libmc.h, 331
- MC\_AclRetrieve\_chdl
  - agent\_lib.h, 256
  - libmc.c, 400
- MC\_AclSend
  - libmc.c, 400
  - libmc.h, 331
- MC\_AclSend\_chdl
  - agent\_lib.h, 256
  - libmc.c, 400
- MC\_AclSetContent
  - libmc.c, 400
  - libmc.h, 332
- MC\_AclSetContent\_chdl
  - agent\_lib.h, 256
  - libmc.c, 401
- MC\_AclSetPerformative
  - libmc.c, 401
  - libmc.h, 332
- MC\_AclSetPerformative\_chdl
  - agent\_lib.h, 256
  - libmc.c, 401
- MC\_AclSetSender
  - libmc.c, 401
  - libmc.h, 332
- MC\_AclSetSender\_chdl
  - agent\_lib.h, 256
  - libmc.c, 401
- MC\_AclWaitRetrieve
  - libmc.c, 401
  - libmc.h, 332
- MC\_AclWaitRetrieve\_chdl
  - agent\_lib.h, 256
  - libmc.c, 402
- MC\_AddAgent
  - libmc.c, 402
  - libmc.h, 333
- MC\_AddAgent\_chdl
  - agent\_lib.h, 256
  - libmc.c, 402
- MC\_AgentStatus\_e
  - libmc.h, 328
  - LibMC::MCAgent, 140
- MC\_AgentType\_e
  - libmc.h, 328
  - LibMC::MCAgent, 140
- MC\_AgentVariableRetrieve
  - libmc.c, 402
  - libmc.h, 333
- MC\_AgentVariableRetrieve\_chdl
  - agent\_lib.h, 257
  - libmc.c, 403
- MC\_AgentVariableSave
  - libmc.c, 403
  - libmc.h, 334
- MC\_AgentVariableSave\_chdl
  - agent\_lib.h, 257
  - libmc.c, 404
- MC\_Barrier
  - libmc.c, 404
- MC\_Barrier\_chdl
  - agent\_lib.h, 257
  - libmc.c, 404
- MC\_BarrierDelete
  - libmc.c, 404
  - libmc.h, 334
- MC\_BarrierDelete\_chdl
  - agent\_lib.h, 257
  - libmc.c, 404
- MC\_BarrierInit
  - libmc.c, 405
  - libmc.h, 335
- MC\_BarrierInit\_chdl
  - agent\_lib.h, 257
  - libmc.c, 405
- MC\_CallAgentFunc
  - libmc.c, 405
  - libmc.h, 335
- MC\_CallAgentFunc\_chdl
  - agent\_lib.h, 257
  - libmc.c, 406
- MC\_ChInitializeOptions
  - libmc.c, 406
  - libmc.h, 336
- MC\_ComposeAgent
  - libmc.c, 406
  - libmc.h, 336
- MC\_ComposeAgent\_chdl
  - agent\_lib.h, 257
  - libmc.c, 407
- MC\_CondBroadcast
  - libmc.c, 407
  - libmc.h, 337
- MC\_CondBroadcast\_chdl
  - agent\_lib.h, 258
  - libmc.c, 407
- MC\_CondReset
  - libmc.c, 407
  - libmc.h, 337
- MC\_CondReset\_chdl
  - agent\_lib.h, 258

- libmc.c, 408
- MC\_CondSignal
  - libmc.c, 408
  - libmc.h, 337
- MC\_CondSignal\_chdl
  - agent\_lib.h, 258
  - libmc.c, 409
- MC\_CondWait
  - libmc.c, 409
  - libmc.h, 338
- MC\_CondWait\_chdl
  - agent\_lib.h, 258
  - libmc.c, 409
- MC\_CopyAgent
  - libmc.c, 409
  - libmc.h, 339
- MC\_DeleteAgent
  - libmc.c, 410
  - libmc.h, 339
- MC\_DeleteAgent\_chdl
  - agent\_lib.h, 258
  - libmc.c, 410
- MC\_DeregisterService
  - libmc.c, 410
- MC\_DeregisterService\_chdl
  - agent\_lib.h, 258
  - libmc.c, 410
- MC\_DestroyServiceSearchResult
  - libmc.c, 411
  - libmc.h, 339
- MC\_DestroyServiceSearchResult\_chdl
  - agent\_lib.h, 258
  - libmc.c, 411
- MC\_End
  - libmc.c, 411
  - libmc.h, 340
- MC\_End\_chdl
  - agent\_lib.h, 259
  - libmc.c, 412
- mc\_error.h
  - error\_code\_e, 367
  - error\_code\_t, 367
- MC\_FindAgentByID
  - libmc.c, 412
  - libmc.h, 340
- MC\_FindAgentByID\_chdl
  - agent\_lib.h, 259
  - libmc.c, 412
- MC\_FindAgentByName
  - libmc.c, 412
  - libmc.h, 340
- MC\_FindAgentByName\_chdl
  - agent\_lib.h, 259
  - libmc.c, 413
- MC\_FipaPerformative\_e
  - LibMC::MCAclMessage, 99
- MC\_GetAgentArrivalTime
  - libmc.c, 413
- MC\_GetAgentArrivalTime\_chdl
  - libmc.c, 413
- MC\_GetAgentExecEngine
  - libmc.c, 413
  - libmc.h, 341
- MC\_GetAgentID
  - libmc.c, 413
  - libmc.h, 341
- MC\_GetAgentID\_chdl
  - agent\_lib.h, 259
  - libmc.c, 414
- MC\_GetAgentName
  - libmc.c, 414
  - libmc.h, 341
- MC\_GetAgentName\_chdl
  - agent\_lib.h, 259
  - libmc.c, 414
- MC\_GetAgentNumTasks
  - libmc.c, 414
  - libmc.h, 341
- MC\_GetAgentNumTasks\_chdl
  - agent\_lib.h, 259
  - libmc.c, 414
- MC\_GetAgentReturnData
  - libmc.c, 414
  - libmc.h, 342
- MC\_GetAgentStatus
  - libmc.c, 415
  - libmc.h, 342
- MC\_GetAgentStatus\_chdl
  - agent\_lib.h, 259
  - libmc.c, 415
- MC\_GetAgentType
  - libmc.c, 415
  - libmc.h, 342
- MC\_GetAgentXMLString
  - libmc.c, 415
  - libmc.h, 343
- MC\_GetAgentXMLString\_chdl
  - agent\_lib.h, 259
  - libmc.c, 416
- MC\_GetAllAgents
  - libmc.c, 416
- MC\_GetTimeOfDay\_chdl
  - agent\_lib.h, 260
  - libmc.c, 416
- MC\_HaltAgency
  - libmc.c, 416
  - libmc.h, 343
- MC\_HaltAgency\_chdl

- agent\_lib.h, 260
- libmc.c, 416
- MC\_Initialize
  - libmc.c, 417
  - libmc.h, 343
- MC\_InitializeAgencyOptions
  - libmc.c, 417
  - libmc.h, 344
- MC\_LoadAgentFromFile
  - libmc.c, 417
  - libmc.h, 344
- MC\_MainLoop
  - libmc.c, 418
  - libmc.h, 344
- MC\_MutexLock
  - libmc.c, 418
  - libmc.h, 345
- MC\_MutexLock\_chdl
  - agent\_lib.h, 260
  - libmc.c, 418
- MC\_MutexUnlock
  - libmc.c, 419
  - libmc.h, 345
- MC\_MutexUnlock\_chdl
  - agent\_lib.h, 260
  - libmc.c, 419
- mc\_platform
  - agency\_s, 32
  - agent\_s, 41
  - df.h, 290
  - LibMC::MCAgency::MCAgency\_t, 134
- mc\_platform.c
  - DEFAULT\_HOSTNAME\_LENGTH, 436
  - mc\_platform\_Destroy, 436
  - mc\_platform\_Initialize, 436
- mc\_platform.h
  - mc\_platform\_Destroy, 369
  - mc\_platform\_Initialize, 370
  - mc\_platform\_p, 369
  - mc\_platform\_t, 369
- mc\_platform\_Destroy
  - mc\_platform.c, 436
  - mc\_platform.h, 369
- mc\_platform\_Initialize
  - mc\_platform.c, 436
  - mc\_platform.h, 370
- mc\_platform\_p
  - mc\_platform.h, 369
- mc\_platform\_s, 91
  - acc, 93
  - agent\_queue, 92
  - ams, 92
  - asm\_message\_queue, 92
  - barrier\_queue, 93
  - client\_thread\_arg, 93
  - cmd\_prompt, 93
  - connection\_queue, 92
  - default\_agentstatus, 93
  - df, 92
  - err, 91
  - giant, 95
  - giant\_cond, 95
  - giant\_lock, 95
  - hostname, 91
  - interp\_options, 94
  - listen\_thread\_arg, 93
  - MC\_signal, 94
  - MC\_signal\_cond, 94
  - MC\_signal\_lock, 94
  - MC\_steer\_command, 94
  - MC\_steer\_cond, 95
  - MC\_steer\_lock, 94
  - MC\_sync\_cond, 94
  - MC\_sync\_lock, 94
  - message\_queue, 92
  - port, 92
  - quit, 95
  - quit\_lock, 95
  - stack\_size, 93
  - syncList, 93
- mc\_platform\_t
  - mc\_platform.h, 369
- MC\_PrintAgentCode
  - libmc.c, 419
  - libmc.h, 345
- MC\_PrintAgentCode\_chdl
  - agent\_lib.h, 260
  - libmc.c, 419
- MC\_RegisterService
  - libmc.c, 419
  - libmc.h, 346
- MC\_RegisterService\_chdl
  - agent\_lib.h, 260
  - libmc.c, 420
- MC\_ResetSignal
  - libmc.c, 420
  - libmc.h, 346
- MC\_ResumeAgency
  - libmc.c, 420
  - libmc.h, 346
- MC\_ResumeAgency\_chdl
  - agent\_lib.h, 260
  - libmc.c, 421
- MC\_RetrieveAgent
  - libmc.c, 421
  - libmc.h, 347
- MC\_RetrieveAgent\_chdl
  - agent\_lib.h, 260



- libmc.c, 421
- MC\_RetrieveAgentCode
  - libmc.c, 421
  - libmc.h, 347
- MC\_RetrieveAgentCode\_chdl
  - agent\_lib.h, 261
  - libmc.c, 421
- mc\_rwlock.c
  - mc\_rwlock\_destroy, 438
  - mc\_rwlock\_init, 438
  - mc\_rwlock\_rdlock, 438
  - mc\_rwlock\_rdunlock, 438
  - mc\_rwlock\_wrlock, 438
  - mc\_rwlock\_wrunlock, 439
- mc\_rwlock.h
  - mc\_rwlock\_destroy, 371
  - mc\_rwlock\_init, 371
  - mc\_rwlock\_p, 371
  - mc\_rwlock\_rdlock, 371
  - mc\_rwlock\_rdunlock, 372
  - mc\_rwlock\_t, 371
  - mc\_rwlock\_wrlock, 372
  - mc\_rwlock\_wrunlock, 372
- mc\_rwlock\_destroy
  - mc\_rwlock.c, 438
  - mc\_rwlock.h, 371
- mc\_rwlock\_init
  - mc\_rwlock.c, 438
  - mc\_rwlock.h, 371
- mc\_rwlock\_p
  - mc\_rwlock.h, 371
- mc\_rwlock\_rdlock
  - mc\_rwlock.c, 438
  - mc\_rwlock.h, 371
- mc\_rwlock\_rdunlock
  - mc\_rwlock.c, 438
  - mc\_rwlock.h, 372
- mc\_rwlock\_s, 96
  - cond, 96
  - lock, 96
  - num\_readers, 96
  - write\_flag, 96
  - write\_request, 96
- mc\_rwlock\_t
  - mc\_rwlock.h, 371
- mc\_rwlock\_wrlock
  - mc\_rwlock.c, 438
  - mc\_rwlock.h, 372
- mc\_rwlock\_wrunlock
  - mc\_rwlock.c, 439
  - mc\_rwlock.h, 372
- MC\_SearchForService
  - libmc.c, 421
  - libmc.h, 347
- MC\_SearchForService\_chdl
  - agent\_lib.h, 261
  - libmc.c, 422
- MC\_SemaphorePost
  - libmc.c, 422
  - libmc.h, 348
- MC\_SemaphorePost\_chdl
  - agent\_lib.h, 261
  - libmc.c, 422
- MC\_SemaphoreWait
  - libmc.c, 423
  - libmc.h, 348
- MC\_SemaphoreWait\_chdl
  - agent\_lib.h, 261
  - libmc.c, 423
- MC\_SendAgentMigrationMessage
  - libmc.c, 423
  - libmc.h, 349
- MC\_SendAgentMigrationMessage\_chdl
  - agent\_lib.h, 261
  - libmc.c, 423
- MC\_SendAgentMigrationMessageFile
  - libmc.c, 424
  - libmc.h, 349
- MC\_SendAgentMigrationMessageFile\_chdl
  - agent\_lib.h, 261
  - libmc.c, 424
- MC\_SendCh
  - libmc.c, 424
- MC\_SendSteerCommand
  - libmc.c, 424
- MC\_SendSteerCommand\_chdl
  - agent\_lib.h, 261
  - libmc.c, 424
- MC\_SetAgentStatus
  - libmc.c, 424
  - libmc.h, 349
- MC\_SetAgentStatus\_chdl
  - agent\_lib.h, 261
  - libmc.c, 425
- MC\_SetDefaultAgentStatus
  - libmc.c, 425
  - libmc.h, 350
- MC\_SetDefaultAgentStatus\_chdl
  - agent\_lib.h, 261
  - libmc.c, 425
- MC\_SetThreadOff
  - libmc.c, 425
  - libmc.h, 350
- MC\_SetThreadOn
  - libmc.c, 426
  - libmc.h, 350
- MC\_SetThreadsAllOff
  - libmc.c, 426

- libmc.h, 351
- MC\_SetThreadsAllOn
  - libmc.c, 426
  - libmc.h, 351
- MC\_signal
  - mc\_platform\_s, 94
- MC\_signal\_cond
  - mc\_platform\_s, 94
- MC\_Signal\_e
  - libmc.h, 328
  - LibMC::MCAgency, 110
- MC\_signal\_lock
  - mc\_platform\_s, 94
- MC\_Steer
  - libmc.c, 427
  - libmc.h, 351
- MC\_steer\_command
  - mc\_platform\_s, 94
- MC\_steer\_cond
  - mc\_platform\_s, 95
- MC\_steer\_lock
  - mc\_platform\_s, 94
- MC\_SteerCommand\_e
  - libmc.h, 329
  - LibMC::MCAgency, 110
- MC\_SteerCommand\_t
  - libmc.h, 327
- MC\_SteerControl
  - libmc.c, 427
  - libmc.h, 352
- MC\_sync\_cond
  - mc\_platform\_s, 94
- MC\_sync\_lock
  - mc\_platform\_s, 94
- MC\_SyncDelete
  - libmc.c, 428
  - libmc.h, 352
- MC\_SyncDelete\_chdl
  - agent\_lib.h, 262
  - libmc.c, 428
- MC\_SyncInit
  - libmc.c, 428
  - libmc.h, 353
- MC\_SyncInit\_chdl
  - agent\_lib.h, 262
  - libmc.c, 429
- MC\_TerminateAgent
  - libmc.c, 429
  - libmc.h, 353
- MC\_TerminateAgent\_chdl
  - agent\_lib.h, 262
  - libmc.c, 429
- MC\_ThreadIndex\_e
  - libmc.h, 329
- LibMC::MCAgency, 109
- MC\_Wait
  - libmc.h, 326
- MC\_WaitAgent
  - libmc.c, 429
  - libmc.h, 353
- MC\_WaitRetrieveAgent
  - libmc.c, 429
  - libmc.h, 354
- MC\_WaitSignal
  - libmc.c, 430
  - libmc.h, 354
- MCAclMessage
  - LibMC::MCAclMessage, 100
- MCAgency
  - LibMC::MCAgency, 111
- MCAgency\_t
  - libmc.h, 327
- MCAgencyOptions\_s, 136
  - default\_agent\_status, 136
  - enable\_security, 136
  - modified, 136
  - stack\_size, 136
  - threads, 136
- MCAgencyOptions\_t
  - libmc.h, 327
- MCAgencyState
  - LibMC::MCAgency, 109
- MCAgent
  - LibMC::MCAgent, 141
- MCAgent\_t
  - libmc.h, 327
- mcddl
  - LibMC::MCAgency, 130
- md2
  - md2.c, 684
  - md2.h, 550
- md2.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, 684
  - \_md2\_src, 687
  - md2, 684
  - md2\_file, 684
  - md2\_finish, 684
  - md2\_hmac, 685
  - md2\_hmac\_finish, 685
  - md2\_hmac\_starts, 685
  - md2\_hmac\_update, 685
  - md2\_process, 686
  - md2\_self\_test, 686
  - md2\_starts, 686
  - md2\_update, 686
  - PI\_SUBST, 687
- md2.h
  - md2, 550

- md2\_file, 550
- md2\_finish, 550
- md2\_hmac, 550
- md2\_hmac\_finish, 551
- md2\_hmac\_starts, 551
- md2\_hmac\_update, 551
- md2\_self\_test, 552
- md2\_starts, 552
- md2\_update, 552
- md2\_context, 148
  - buffer, 148
  - cksum, 148
  - ipad, 148
  - left, 149
  - opad, 148
  - state, 148
- md2\_file
  - md2.c, 684
  - md2.h, 550
- md2\_finish
  - md2.c, 684
  - md2.h, 550
- md2\_hmac
  - md2.c, 685
  - md2.h, 550
- md2\_hmac\_finish
  - md2.c, 685
  - md2.h, 551
- md2\_hmac\_starts
  - md2.c, 685
  - md2.h, 551
- md2\_hmac\_update
  - md2.c, 685
  - md2.h, 551
- md2\_process
  - md2.c, 686
- md2\_self\_test
  - md2.c, 686
  - md2.h, 552
- md2\_starts
  - md2.c, 686
  - md2.h, 552
- md2\_update
  - md2.c, 686
  - md2.h, 552
- md4
  - md4.c, 690
  - md4.h, 554
- md4.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, 689
  - \_md4\_src, 693
  - F, 689
  - GET\_UINT32\_LE, 689
  - md4, 690
  - md4\_file, 690
  - md4\_finish, 690
  - md4\_hmac, 691
  - md4\_hmac\_finish, 691
  - md4\_hmac\_starts, 691
  - md4\_hmac\_update, 692
  - md4\_padding, 693
  - md4\_process, 692
  - md4\_self\_test, 692
  - md4\_starts, 692
  - md4\_update, 692
  - P, 689
  - PUT\_UINT32\_LE, 689
  - S, 690
- md4.h
  - md4, 554
  - md4\_file, 554
  - md4\_finish, 554
  - md4\_hmac, 554
  - md4\_hmac\_finish, 555
  - md4\_hmac\_starts, 555
  - md4\_hmac\_update, 555
  - md4\_self\_test, 556
  - md4\_starts, 556
  - md4\_update, 556
- md4\_context, 150
  - buffer, 150
  - ipad, 150
  - opad, 150
  - state, 150
  - total, 150
- md4\_file
  - md4.c, 690
  - md4.h, 554
- md4\_finish
  - md4.c, 690
  - md4.h, 554
- md4\_hmac
  - md4.c, 691
  - md4.h, 554
- md4\_hmac\_finish
  - md4.c, 691
  - md4.h, 555
- md4\_hmac\_starts
  - md4.c, 691
  - md4.h, 555
- md4\_hmac\_update
  - md4.c, 692
  - md4.h, 555
- md4\_padding
  - md4.c, 693
- md4\_process
  - md4.c, 692
- md4\_self\_test

- md4.c, 692
- md4.h, 556
- md4\_starts
  - md4.c, 692
  - md4.h, 556
- md4\_update
  - md4.c, 692
  - md4.h, 556
- md5
  - md5.c, 696
  - md5.h, 558
- md5.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, 695
  - \_md5\_src, 699
  - F, 695
  - GET\_UINT32\_LE, 695
  - md5, 696
  - md5\_file, 696
  - md5\_finish, 696
  - md5\_hmac, 697
  - md5\_hmac\_finish, 697
  - md5\_hmac\_starts, 697
  - md5\_hmac\_update, 698
  - md5\_padding, 699
  - md5\_process, 698
  - md5\_self\_test, 698
  - md5\_starts, 698
  - md5\_update, 698
  - P, 695
  - PUT\_UINT32\_LE, 695
  - S, 696
- md5.h
  - md5, 558
  - md5\_file, 558
  - md5\_finish, 558
  - md5\_hmac, 558
  - md5\_hmac\_finish, 559
  - md5\_hmac\_starts, 559
  - md5\_hmac\_update, 559
  - md5\_self\_test, 560
  - md5\_starts, 560
  - md5\_update, 560
- md5\_check
  - md5sum.c, 764
- md5\_context, 152
  - buffer, 152
  - ipad, 152
  - opad, 152
  - state, 152
  - total, 152
- md5\_file
  - md5.c, 696
  - md5.h, 558
- md5\_finish
  - md5.c, 696
  - md5.h, 558
- md5\_hmac
  - md5.c, 697
  - md5.h, 558
- md5\_hmac\_finish
  - md5.c, 697
  - md5.h, 559
- md5\_hmac\_starts
  - md5.c, 697
  - md5.h, 559
- md5\_hmac\_update
  - md5.c, 698
  - md5.h, 559
- md5\_padding
  - md5.c, 699
- md5\_print
  - md5sum.c, 764
- md5\_process
  - md5.c, 698
- md5\_self\_test
  - md5.c, 698
  - md5.h, 560
- md5\_starts
  - md5.c, 698
  - md5.h, 560
- md5\_update
  - md5.c, 698
  - md5.h, 560
- md5\_wrapper
  - md5sum.c, 764
- md5sum.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, 764
  - main, 764
  - md5\_check, 764
  - md5\_print, 764
  - md5\_wrapper, 764
- message
  - dynstring\_s, 57
  - fipa\_message\_string\_s, 72
- message.c
  - http\_to\_hostport, 446
  - message\_Copy, 446
  - message\_Destroy, 446
  - message\_InitializeFromAgent, 446
  - message\_InitializeFromConnection, 446
  - message\_InitializeFromString, 447
  - message\_New, 447
  - message\_Send, 447
  - SOCKET\_INPUT\_SIZE, 446
- message.h
  - AGENT\_UPDATE, 374
  - CANCEL, 374
  - ENCRYPTED\_DATA, 374

- ENCRYPTION\_INITIALIZE, 374
- FIPA\_ACL, 374
- http\_to\_hostport, 374
- message\_Copy, 374
- message\_Destroy, 374
- message\_InitializeFromAgent, 375
- message\_InitializeFromConnection, 375
- message\_InitializeFromString, 375
- message\_New, 375
- message\_p, 373
- message\_Send, 375
- message\_t, 373
- message\_type\_e, 374
- message\_type\_t, 374
- MOBILE\_AGENT, 374
- N\_UNDRSTD, 374
- NUM\_MESSAGE\_TYPE, 374
- QUER\_IF, 374
- QUER\_REF, 374
- RELAY, 374
- REQUEST, 374
- REQUEST\_ENCRYPTION\_INITIALIZE, 374
- RETURN\_MSG, 374
- SUBSCRIBE, 374
- message\_body
  - message\_s, 155
- message\_Copy
  - message.c, 446
  - message.h, 374
- message\_Destroy
  - message.c, 446
  - message.h, 374
- message\_id
  - message\_s, 154
- message\_InitializeFromAgent
  - message.c, 446
  - message.h, 375
- message\_InitializeFromConnection
  - message.c, 446
  - message.h, 375
- message\_InitializeFromString
  - message.c, 447
  - message.h, 375
- message\_New
  - message.c, 447
  - message.h, 375
- message\_p
  - message.h, 373
- message\_parts
  - mtp\_http\_s, 162
- message\_queue
  - mc\_platform\_s, 92
- message\_queue.c
  - message\_queue\_SendOutgoing, 448
- message\_queue\_SendOutgoing
  - message\_queue.c, 448
- message\_s, 154
  - addr, 154
  - agent\_xml\_flag, 156
  - connect\_id, 154
  - from\_address, 155
  - http\_type, 155
  - isHTTP, 154
  - message\_body, 155
  - message\_id, 154
  - message\_type, 155
  - target, 156
  - to\_address, 156
  - update\_name, 155
  - update\_num, 155
  - xml\_payload, 155
  - xml\_root, 155
- message\_Send
  - message.c, 447
  - message.h, 375
- message\_t
  - message.h, 373
- message\_type
  - message\_s, 155
- message\_type\_e
  - message.h, 374
- message\_type\_t
  - message.h, 374
- message\_xml\_parse
  - xml\_parser.c, 806
  - xml\_parser.h, 389
- message\_xml\_parse\_\_message
  - xml\_parser.c, 806
  - xml\_parser.h, 389
- METHOD\_NOT\_ALLOWED
  - mtp\_http.h, 378
- millisecond
  - fipa\_DateTime\_s, 69
- mime\_version
  - mtp\_http\_s, 162
- min
  - \_x509\_time, 28
- minlen
  - ssl\_context, 193
- minor\_ver
  - ssl\_context, 186
- minute
  - fipa\_DateTime\_s, 68
- MISC
  - df.h, 287
- MOBILE\_AGENT
  - message.h, 374

- MODE\_DECRYPT
  - aescript2.c, [761](#)
- MODE\_ENCRYPT
  - aescript2.c, [761](#)
- modified
  - LibMC::MCAgency::MCAgencyOptions\_t, [135](#)
  - MCAgencyOptions\_s, [136](#)
- mon
  - \_x509\_time, [28](#)
- month
  - fipa\_DateTime\_s, [68](#)
- mpi, [157](#)
  - n, [157](#)
  - p, [157](#)
  - s, [157](#)
- mpi\_add\_abs
  - bignum.c, [643](#)
  - bignum.h, [521](#)
- mpi\_add\_int
  - bignum.c, [643](#)
  - bignum.h, [521](#)
- mpi\_add\_mpi
  - bignum.c, [643](#)
  - bignum.h, [522](#)
- mpi\_cmp\_abs
  - bignum.c, [643](#)
  - bignum.h, [522](#)
- mpi\_cmp\_int
  - bignum.c, [644](#)
  - bignum.h, [522](#)
- mpi\_cmp\_mpi
  - bignum.c, [644](#)
  - bignum.h, [522](#)
- mpi\_copy
  - bignum.c, [644](#)
  - bignum.h, [523](#)
- mpi\_demo.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, [775](#)
  - main, [775](#)
- mpi\_div\_int
  - bignum.c, [644](#)
  - bignum.h, [523](#)
- mpi\_div\_mpi
  - bignum.c, [645](#)
  - bignum.h, [523](#)
- mpi\_exp\_mod
  - bignum.c, [645](#)
  - bignum.h, [524](#)
- mpi\_free
  - bignum.c, [645](#)
  - bignum.h, [524](#)
- mpi\_gcd
  - bignum.c, [646](#)
- bignum.h, [524](#)
- mpi\_gen\_prime
  - bignum.c, [646](#)
  - bignum.h, [525](#)
- mpi\_get\_digit
  - bignum.c, [646](#)
- mpi\_grow
  - bignum.c, [647](#)
  - bignum.h, [525](#)
- mpi\_init
  - bignum.c, [647](#)
  - bignum.h, [525](#)
- mpi\_inv\_mod
  - bignum.c, [647](#)
  - bignum.h, [525](#)
- mpi\_is\_prime
  - bignum.c, [647](#)
  - bignum.h, [526](#)
- mpi\_lsb
  - bignum.c, [648](#)
  - bignum.h, [526](#)
- mpi\_lset
  - bignum.c, [648](#)
  - bignum.h, [526](#)
- mpi\_mod\_int
  - bignum.c, [648](#)
  - bignum.h, [526](#)
- mpi\_mod\_mpi
  - bignum.c, [648](#)
  - bignum.h, [527](#)
- mpi\_montg\_init
  - bignum.c, [648](#)
- mpi\_montmul
  - bignum.c, [649](#)
- mpi\_montred
  - bignum.c, [649](#)
- mpi\_msb
  - bignum.c, [649](#)
  - bignum.h, [527](#)
- mpi\_mul\_hlp
  - bignum.c, [649](#)
- mpi\_mul\_int
  - bignum.c, [649](#)
  - bignum.h, [527](#)
- mpi\_mul\_mpi
  - bignum.c, [649](#)
  - bignum.h, [527](#)
- mpi\_read\_binary
  - bignum.c, [650](#)
  - bignum.h, [528](#)
- mpi\_read\_file
  - bignum.c, [650](#)
  - bignum.h, [528](#)
- mpi\_read\_string

- bignum.c, 650
- bignum.h, 528
- mpi\_self\_test
  - bignum.c, 651
  - bignum.h, 529
- mpi\_shift\_l
  - bignum.c, 651
  - bignum.h, 529
- mpi\_shift\_r
  - bignum.c, 651
  - bignum.h, 529
- mpi\_sub\_abs
  - bignum.c, 652
  - bignum.h, 530
- mpi\_sub\_hlp
  - bignum.c, 652
- mpi\_sub\_int
  - bignum.c, 652
  - bignum.h, 530
- mpi\_sub\_mpi
  - bignum.c, 652
  - bignum.h, 530
- mpi\_swap
  - bignum.c, 652
  - bignum.h, 530
- mpi\_write\_binary
  - bignum.c, 653
  - bignum.h, 530
- mpi\_write\_file
  - bignum.c, 653
  - bignum.h, 531
- mpi\_write\_hlp
  - bignum.c, 653
- mpi\_write\_string
  - bignum.c, 654
  - bignum.h, 531
- mtp\_http.h
  - ACCEPTED, 378
  - BAD\_REQUEST, 378
  - CONFLICT, 378
  - CONTINUE, 378
  - CREATED, 378
  - EXPECTATION\_FAILED, 379
  - FAILED\_DEPENDANCY, 379
  - FORBIDDEN, 378
  - GONE, 378
  - HTTP\_CONNECT, 378
  - HTTP\_DELETE, 378
  - HTTP\_GET, 377
  - HTTP\_HEAD, 377
  - HTTP\_NUM\_PERFORMATIVES, 378
  - HTTP\_OPTIONS, 378
  - HTTP\_PERFORMATIVE\_UNDEF, 377
  - HTTP\_PERFORMATIVE\_ZERO, 377
  - HTTP\_POST, 377
  - HTTP\_PUT, 377
  - HTTP\_RESPONSE, 378
  - HTTP\_TRACE, 378
  - LENGTH\_REQUIRED, 378
  - LOCKED, 379
  - METHOD\_NOT\_ALLOWED, 378
  - MULTI\_STATUS, 378
  - NO\_CONTENT, 378
  - NON\_AUTHORITATIVE\_INFORMATION, 378
  - NOT\_ACCEPTABLE, 378
  - NOT\_FOUND, 378
  - OK, 378
  - PARTIAL\_CONTENT, 378
  - PAYMENT\_REQUIRED, 378
  - PRECONDITION\_FAILED, 378
  - PROCESSING, 378
  - PROXY\_AUTHENTICATION\_REQUIRED, 378
  - REQUEST\_TIMEOUT, 378
  - REQUEST\_URI\_TOO\_LONG, 378
  - REQUESTED\_RANGE\_NOT\_SATISFIABLE, 378
  - REQUST\_ENTITY\_TOO\_LARGE, 378
  - RESET\_CONTENT, 378
  - RETRY\_WITH, 379
  - SWITCHING\_PROTOCOLS, 378
  - UNAUTHORIZED, 378
  - UNORDERED\_COLLECTION, 379
  - UNPROCESSABLE\_ENTITY, 379
  - UNSUPPORTED\_MEDIA\_TYPE, 378
  - UPGRADE\_REQUIRED, 379
- mtp\_http.c
  - http\_GetExpression, 450
  - http\_GetToken, 450
  - HTTP\_PARSE\_EXPR, 449
  - http\_ParseExpression, 450
  - http\_ParseHeader, 451
  - mtp\_http\_ComposeMessage, 451
  - mtp\_http\_CreateMessage, 451
  - mtp\_http\_Destroy, 451
  - mtp\_http\_InitializeFromConnection, 452
  - mtp\_http\_New, 452
  - mtp\_http\_Parse, 452
  - mtp\_http\_ParseResponse, 452
  - SAFE\_FREE, 450
- mtp\_http.h
  - http\_GetExpression, 379
  - http\_GetToken, 379
  - http\_ParseExpression, 379
  - http\_ParseHeader, 380
  - http\_performative\_e, 377
  - http\_status\_code\_e, 378

- mtp\_http\_ComposeMessage, 380
- mtp\_http\_content\_t, 377
- mtp\_http\_CreateMessage, 380
- mtp\_http\_Destroy, 380
- mtp\_http\_InitializeFromConnection, 380
- mtp\_http\_New, 380
- mtp\_http\_p, 377
- mtp\_http\_Parse, 381
- mtp\_http\_t, 377
- SOCKET\_INPUT\_SIZE, 377
- mtp\_http\_ComposeMessage
  - mtp\_http.c, 451
  - mtp\_http.h, 380
- mtp\_http\_content\_s, 159
  - content\_type, 159
  - data, 159
- mtp\_http\_content\_t
  - mtp\_http.h, 377
- mtp\_http\_CreateMessage
  - mtp\_http.c, 451
  - mtp\_http.h, 380
- mtp\_http\_Destroy
  - mtp\_http.c, 451
  - mtp\_http.h, 380
- mtp\_http\_InitializeFromConnection
  - mtp\_http.c, 452
  - mtp\_http.h, 380
- mtp\_http\_New
  - mtp\_http.c, 452
  - mtp\_http.h, 380
- mtp\_http\_p
  - mtp\_http.h, 377
- mtp\_http\_Parse
  - mtp\_http.c, 452
  - mtp\_http.h, 381
- mtp\_http\_ParseResponse
  - mtp\_http.c, 452
- mtp\_http\_s, 160
  - accept\_ranges, 161
  - boundary, 162
  - cache\_control, 162
  - connection, 161
  - content, 162
  - content\_length, 161
  - content\_type, 161
  - date, 161
  - host, 160
  - http\_performative, 160
  - http\_status\_code, 160
  - http\_version, 160
  - message\_parts, 162
  - mime\_version, 162
  - response\_code, 162
  - response\_string, 162
  - return\_code, 161
  - server, 161
  - target, 161
  - user\_agent, 162
- mtp\_http\_t
  - mtp\_http.h, 377
- MUL
  - aes.c, 630
- MULADDC\_CORE
  - bn\_asm.h, 533
- MULADDC\_INIT
  - bn\_asm.h, 533
- MULADDC\_STOP
  - bn\_asm.h, 533
- MULTI\_STATUS
  - mtp\_http.h, 378
- MUTEX\_DESTROY
  - macros.h, 361
- MUTEX\_INIT
  - macros.h, 362
- MUTEX\_LOCK
  - macros.h, 362
- MUTEX\_NEW
  - macros.h, 362
- MUTEX\_T
  - macros.h, 362
- MUTEX\_UNLOCK
  - macros.h, 363
- MutexLock
  - LibMC::MCAgency, 117
- MutexUnlock
  - LibMC::MCAgency, 117
- mxml-attr.c
  - mxmlElementGetAttr, 457
  - mxmlElementSetAttr, 457
- mxml-entity.c
  - callbacks, 459
  - default\_callback, 458
  - mxmlEntityAddCallback, 458
  - mxmlEntityGetName, 458
  - mxmlEntityGetValue, 458
  - mxmlEntityRemoveCallback, 458
  - num\_callbacks, 459
- mxml-file.c
  - ENCODE\_UTF16BE, 461
  - ENCODE\_UTF16LE, 461
  - ENCODE\_UTF8, 461
  - mxml\_add\_char, 461
  - mxml\_bad\_char, 461
  - mxml\_custom\_load\_cb, 466
  - mxml\_custom\_save\_cb, 466
  - mxml\_error\_cb, 466
  - mxml\_fd\_getc, 462
  - mxml\_fd\_putc, 462



- mxml\_fd\_read, 462
- mxml\_fd\_write, 462
- mxml\_fdbuf\_t, 461
- mxml\_file\_getc, 462
- mxml\_file\_putc, 462
- mxml\_get\_entity, 462
- mxml\_load\_data, 463
- mxml\_parse\_element, 463
- mxml\_string\_getc, 463
- mxml\_string\_putc, 463
- mxml\_write\_name, 463
- mxml\_write\_node, 463
- mxml\_write\_string, 464
- mxml\_write\_ws, 464
- mxmlLoadFd, 464
- mxmlLoadFile, 464
- mxmlLoadString, 464
- mxmlSaveAllocString, 465
- mxmlSaveFd, 465
- mxmlSaveFile, 465
- mxmlSaveString, 465
- mxmlSetCustomHandlers, 465
- mxmlSetErrorCallback, 465
- mxml-index.c
  - index\_compare, 467
  - index\_find, 467
  - index\_sort, 467
  - mxmlIndexDelete, 467
  - mxmlIndexEnum, 468
  - mxmlIndexFind, 468
  - mxmlIndexNew, 468
  - mxmlIndexReset, 468
- mxml-node.c
  - mxml\_new, 469
  - mxmlAdd, 469
  - mxmlDelete, 469
  - mxmlNewCustom, 470
  - mxmlNewElement, 470
  - mxmlNewInteger, 470
  - mxmlNewOpaque, 470
  - mxmlNewReal, 470
  - mxmlNewText, 471
  - mxmlNewTextf, 471
  - mxmlRemove, 471
- mxml-private.c
  - mxml\_error, 472
  - mxml\_error\_cb, 473
  - mxml\_integer\_cb, 472
  - mxml\_opaque\_cb, 472
  - mxml\_real\_cb, 472
- mxml-search.c
  - mxmlFindElement, 474
  - mxmlWalkNext, 474
  - mxmlWalkPrev, 474
- mxml-set.c
  - mxmlSetCustom, 475
  - mxmlSetElement, 475
  - mxmlSetInteger, 475
  - mxmlSetOpaque, 475
  - mxmlSetReal, 475
  - mxmlSetText, 475
  - mxmlSetTextf, 476
- mxml-string.c
  - mxml\_strdupf, 477
  - mxml\_vsnprintf, 477
- mxml.h
  - MXML\_CUSTOM, 483
  - MXML\_ELEMENT, 483
  - MXML\_INTEGER, 483
  - MXML\_OPAQUE, 483
  - MXML\_REAL, 483
  - MXML\_TEXT, 483
  - MXML\_ADD\_AFTER, 480
  - MXML\_ADD\_BEFORE, 480
  - MXML\_ADD\_TO\_PARENT, 480
  - mxml\_attr\_t, 482
  - mxml\_custom\_load\_cb\_t, 482
  - mxml\_custom\_save\_cb\_t, 482
  - mxml\_custom\_t, 482
  - MXML\_DESCEND, 480
  - MXML\_DESCEND\_FIRST, 480
  - mxml\_element\_t, 483
  - mxml\_error, 483
  - mxml\_index\_t, 483
  - MXML\_INTEGER\_CALLBACK, 481
  - mxml\_integer\_cb, 483
  - MXML\_NO\_CALLBACK, 481
  - MXML\_NO\_DESCEND, 481
  - MXML\_NO\_PARENT, 481
  - mxml\_node\_t, 483
  - MXML\_OPAQUE\_CALLBACK, 481
  - mxml\_opaque\_cb, 483
  - MXML\_REAL\_CALLBACK, 481
  - mxml\_real\_cb, 483
  - MXML\_TAB, 481
  - MXML\_TEXT\_CALLBACK, 482
  - mxml\_text\_t, 483
  - mxml\_type\_e, 483
  - mxml\_type\_t, 483
  - mxml\_value\_t, 483
  - MXML\_WRAP, 482
  - MXML\_WS\_AFTER\_CLOSE, 482
  - MXML\_WS\_AFTER\_OPEN, 482
  - MXML\_WS\_BEFORE\_CLOSE, 482
  - MXML\_WS\_BEFORE\_OPEN, 482
  - mxmlAdd, 484
  - mxmlDelete, 484
  - mxmlElementGetAttr, 484

- mxmlelementsetattr, 484
- mxmlentityaddcallback, 485
- mxmlentitygetname, 485
- mxmlentitygetvalue, 485
- mxmlentityremovecallback, 485
- mxmlelementfind, 485
- mxmlelementdelete, 485
- mxmlelementenum, 485
- mxmlelementfind, 486
- mxmlelementnew, 486
- mxmlelementreset, 486
- mxmlelementloadfd, 486
- mxmlelementloadfile, 486
- mxmlelementloadstring, 486
- mxmlelementnewcustom, 486
- mxmlelementnewelement, 486
- mxmlelementnewinteger, 487
- mxmlelementnewopaque, 487
- mxmlelementnewreal, 487
- mxmlelementnewtext, 487
- mxmlelementnewtextf, 487
- mxmlelementremove, 488
- mxmlelementsaveallocstring, 488
- mxmlelementsavefd, 488
- mxmlelementsavefile, 488
- mxmlelementsavestring, 488
- mxmlelementsetcustom, 488
- mxmlelementsetcustomhandlers, 488
- mxmlelementsetelement, 488
- mxmlelementseterrorcallback, 488
- mxmlelementsetinteger, 488
- mxmlelementsetopaque, 489
- mxmlelementsetreal, 489
- mxmlelementsettext, 489
- mxmlelementsettextf, 489
- mxmlelementwalknext, 489
- mxmlelementwalkprev, 489
- MXML\_CUSTOM
  - mxml.h, 483
- MXML\_ELEMENT
  - mxml.h, 483
- MXML\_INTEGER
  - mxml.h, 483
- MXML\_OPAQUE
  - mxml.h, 483
- MXML\_REAL
  - mxml.h, 483
- MXML\_TEXT
  - mxml.h, 483
- MXML\_ADD\_AFTER
  - mxml.h, 480
- MXML\_ADD\_BEFORE
  - mxml.h, 480
- mxml\_add\_char
  - mxml-file.c, 461
- MXML\_ADD\_TO\_PARENT
  - mxml.h, 480
- mxml\_attr\_s, 163
  - name, 163
  - value, 163
- mxml\_attr\_t
  - mxml.h, 482
- mxml\_bad\_char
  - mxml-file.c, 461
- mxml\_custom\_load\_cb
  - mxml-file.c, 466
- mxml\_custom\_load\_cb\_t
  - mxml.h, 482
- mxml\_custom\_s, 164
  - data, 164
  - destroy, 164
- mxml\_custom\_save\_cb
  - mxml-file.c, 466
- mxml\_custom\_save\_cb\_t
  - mxml.h, 482
- mxml\_custom\_t
  - mxml.h, 482
- MXML\_DESCEND
  - mxml.h, 480
- MXML\_DESCEND\_FIRST
  - mxml.h, 480
- mxml\_element\_t
  - mxml.h, 483
- mxml\_error
  - mxml-private.c, 472
  - mxml.h, 483
- mxml\_error\_cb
  - mxml-file.c, 466
  - mxml-private.c, 473
- mxml\_fd\_getc
  - mxml-file.c, 462
- mxml\_fd\_putc
  - mxml-file.c, 462
- mxml\_fd\_read
  - mxml-file.c, 462
- mxml\_fd\_write
  - mxml-file.c, 462
- mxml\_fdbuf\_s, 165
  - buffer, 165
  - current, 165
  - end, 165
  - fd, 165
- mxml\_fdbuf\_t
  - mxml-file.c, 461
- mxml\_file\_getc
  - mxml-file.c, 462
- mxml\_file\_putc
  - mxml-file.c, 462

- mxml\_get\_entity
  - mxml-file.c, [462](#)
- mxml\_index\_s, [166](#)
  - alloc\_nodes, [166](#)
  - attr, [166](#)
  - cur\_node, [166](#)
  - nodes, [166](#)
  - num\_nodes, [166](#)
- mxml\_index\_t
  - mxml.h, [483](#)
- MXML\_INTEGER\_CALLBACK
  - mxml.h, [481](#)
- mxml\_integer\_cb
  - mxml-private.c, [472](#)
  - mxml.h, [483](#)
- mxml\_load\_data
  - mxml-file.c, [463](#)
- mxml\_new
  - mxml-node.c, [469](#)
- MXML\_NO\_CALLBACK
  - mxml.h, [481](#)
- MXML\_NO\_DESCEND
  - mxml.h, [481](#)
- MXML\_NO\_PARENT
  - mxml.h, [481](#)
- mxml\_node\_s, [168](#)
  - child, [168](#)
  - last\_child, [169](#)
  - next, [168](#)
  - parent, [168](#)
  - prev, [168](#)
  - type, [168](#)
  - value, [169](#)
- mxml\_node\_t
  - mxml.h, [483](#)
- MXML\_OPAQUE\_CALLBACK
  - mxml.h, [481](#)
- mxml\_opaque\_cb
  - mxml-private.c, [472](#)
  - mxml.h, [483](#)
- mxml\_parse\_element
  - mxml-file.c, [463](#)
- MXML\_REAL\_CALLBACK
  - mxml.h, [481](#)
- mxml\_real\_cb
  - mxml-private.c, [472](#)
  - mxml.h, [483](#)
- mxml\_strdupf
  - config.h, [453](#)
  - mxml-string.c, [477](#)
  - vcnet/config.h, [454](#)
  - vcnet2005/config.h, [455](#)
- mxml\_string\_getc
  - mxml-file.c, [463](#)
- mxml\_string\_putc
  - mxml-file.c, [463](#)
- MXML\_TAB
  - mxml.h, [481](#)
- MXML\_TEXT\_CALLBACK
  - mxml.h, [482](#)
- mxml\_text\_s, [170](#)
  - string, [170](#)
  - whitespace, [170](#)
- mxml\_text\_t
  - mxml.h, [483](#)
- mxml\_type\_e
  - mxml.h, [483](#)
- mxml\_type\_t
  - mxml.h, [483](#)
- mxml\_value\_s, [171](#)
  - attrs, [171](#)
  - name, [171](#)
  - num\_attrs, [171](#)
- mxml\_value\_t
  - mxml.h, [483](#)
- mxml\_value\_u, [172](#)
  - custom, [173](#)
  - element, [172](#)
  - integer, [172](#)
  - opaque, [172](#)
  - real, [172](#)
  - text, [172](#)
- MXML\_VERSION
  - config.h, [453](#)
  - vcnet/config.h, [454](#)
  - vcnet2005/config.h, [455](#)
- mxml\_vsnprintf
  - mxml-string.c, [477](#)
  - vcnet/config.h, [454](#)
  - vcnet2005/config.h, [455](#)
- MXML\_WRAP
  - mxml.h, [482](#)
- mxml\_write\_name
  - mxml-file.c, [463](#)
- mxml\_write\_node
  - mxml-file.c, [463](#)
- mxml\_write\_string
  - mxml-file.c, [464](#)
- mxml\_write\_ws
  - mxml-file.c, [464](#)
- MXML\_WS\_AFTER\_CLOSE
  - mxml.h, [482](#)
- MXML\_WS\_AFTER\_OPEN
  - mxml.h, [482](#)
- MXML\_WS\_BEFORE\_CLOSE
  - mxml.h, [482](#)
- MXML\_WS\_BEFORE\_OPEN
  - mxml.h, [482](#)

- mxxmlAdd
  - mxxml-node.c, 469
  - mxxml.h, 484
- mxxmlDelete
  - mxxml-node.c, 469
  - mxxml.h, 484
- mxxmlDoc.c
  - add\_variable, 491
  - main, 491
  - safe\_strcpy, 491
  - scan\_file, 491
  - sort\_node, 492
  - STATE\_C\_COMMENT, 490
  - STATE\_CHARACTER, 490
  - STATE\_CXX\_COMMENT, 490
  - STATE\_IDENTIFIER, 490
  - STATE\_NONE, 491
  - STATE\_PREPROCESSOR, 491
  - STATE\_STRING, 491
  - update\_comment, 492
  - write\_documentation, 492
  - write\_element, 492
  - write\_string, 492
  - ws\_cb, 493
- mxxmlElementGetAttr
  - mxxml-attr.c, 457
  - mxxml.h, 484
- mxxmlElementSetAttr
  - mxxml-attr.c, 457
  - mxxml.h, 484
- mxxmlEntityAddCallback
  - mxxml-entity.c, 458
  - mxxml.h, 485
- mxxmlEntityGetName
  - mxxml-entity.c, 458
  - mxxml.h, 485
- mxxmlEntityGetValue
  - mxxml-entity.c, 458
  - mxxml.h, 485
- mxxmlEntityRemoveCallback
  - mxxml-entity.c, 458
  - mxxml.h, 485
- mxxmlFindElement
  - mxxml-search.c, 474
  - mxxml.h, 485
- mxxmlIndexDelete
  - mxxml-index.c, 467
  - mxxml.h, 485
- mxxmlIndexEnum
  - mxxml-index.c, 468
  - mxxml.h, 485
- mxxmlIndexFind
  - mxxml-index.c, 468
  - mxxml.h, 486
- mxxmlIndexNew
  - mxxml-index.c, 468
  - mxxml.h, 486
- mxxmlIndexReset
  - mxxml-index.c, 468
  - mxxml.h, 486
- mxxmlLoadFd
  - mxxml-file.c, 464
  - mxxml.h, 486
- mxxmlLoadFile
  - mxxml-file.c, 464
  - mxxml.h, 486
- mxxmlLoadString
  - mxxml-file.c, 464
  - mxxml.h, 486
- mxxmlNewCustom
  - mxxml-node.c, 470
  - mxxml.h, 486
- mxxmlNewElement
  - mxxml-node.c, 470
  - mxxml.h, 486
- mxxmlNewInteger
  - mxxml-node.c, 470
  - mxxml.h, 487
- mxxmlNewOpaque
  - mxxml-node.c, 470
  - mxxml.h, 487
- mxxmlNewReal
  - mxxml-node.c, 470
  - mxxml.h, 487
- mxxmlNewText
  - mxxml-node.c, 471
  - mxxml.h, 487
- mxxmlNewTextf
  - mxxml-node.c, 471
  - mxxml.h, 487
- mxxmlRemove
  - mxxml-node.c, 471
  - mxxml.h, 488
- mxxmlSaveAllocString
  - mxxml-file.c, 465
  - mxxml.h, 488
- mxxmlSaveFd
  - mxxml-file.c, 465
  - mxxml.h, 488
- mxxmlSaveFile
  - mxxml-file.c, 465
  - mxxml.h, 488
- mxxmlSaveString
  - mxxml-file.c, 465
  - mxxml.h, 488
- mxxmlSetCustom
  - mxxml-set.c, 475
  - mxxml.h, 488

- mxmmlSetCustomHandlers
  - mxmml-file.c, 465
  - mxmml.h, 488
- mxmmlSetElement
  - mxmml-set.c, 475
  - mxmml.h, 488
- mxmmlSetErrorCallback
  - mxmml-file.c, 465
  - mxmml.h, 488
- mxmmlSetInteger
  - mxmml-set.c, 475
  - mxmml.h, 488
- mxmmlSetOpaque
  - mxmml-set.c, 475
  - mxmml.h, 489
- mxmmlSetReal
  - mxmml-set.c, 475
  - mxmml.h, 489
- mxmmlSetText
  - mxmml-set.c, 475
  - mxmml.h, 489
- mxmmlSetTextf
  - mxmml-set.c, 476
  - mxmml.h, 489
- mxmmlWalkNext
  - mxmml-search.c, 474
  - mxmml.h, 489
- mxmmlWalkPrev
  - mxmml-search.c, 474
  - mxmml.h, 489
- my\_preferred\_ciphers
  - ssl\_server.c, 784
- myrand
  - benchmark.c, 787
- N
  - rsa\_context, 174
- n
  - mpi, 157
- N\_UNDRSTD
  - message.h, 374
- name
  - agent\_s, 38
  - fipa\_agent\_identifier\_s, 66
  - interpreter\_variable\_data\_s, 87
  - LibMC::MCAgent, 145
  - mxmml\_attr\_s, 163
  - mxmml\_value\_s, 171
- nb\_zero
  - ssl\_context, 188
- net.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, 701
  - net\_accept, 701
  - net\_bind, 701
  - net\_close, 701
  - net\_connect, 701
  - net\_is\_blocking, 702
  - net\_recv, 702
  - net\_send, 702
  - net\_set\_block, 702
  - net\_set\_nonblock, 702
  - net\_usleep, 703
- net.h
  - ERR\_NET\_ACCEPT\_FAILED, 563
  - ERR\_NET\_BIND\_FAILED, 563
  - ERR\_NET\_CONN\_RESET, 563
  - ERR\_NET\_CONNECT\_FAILED, 563
  - ERR\_NET\_LISTEN\_FAILED, 563
  - ERR\_NET\_RECV\_FAILED, 563
  - ERR\_NET\_SEND\_FAILED, 563
  - ERR\_NET\_SOCKET\_FAILED, 563
  - ERR\_NET\_UNKNOWN\_HOST, 563
  - ERR\_NET\_WOULD\_BLOCK, 564
  - net\_accept, 564
  - net\_bind, 564
  - net\_close, 564
  - net\_connect, 564
  - net\_recv, 565
  - net\_send, 565
  - net\_set\_nonblock, 565
  - net\_usleep, 565
- net\_accept
  - net.c, 701
  - net.h, 564
- net\_bind
  - net.c, 701
  - net.h, 564
- net\_close
  - net.c, 701
  - net.h, 564
- net\_connect
  - net.c, 701
  - net.h, 564
- net\_is\_blocking
  - net.c, 702
- net\_recv
  - net.c, 702
  - net.h, 565
- net\_send
  - net.c, 702
  - net.h, 565
- net\_set\_block
  - net.c, 702
- net\_set\_nonblock
  - net.c, 702
  - net.h, 565
- net\_usleep
  - net.c, 703

- net.h, 565
- New
  - LibMC::MCAclMessage, 100
- next
  - \_x509\_cert, 25
  - \_x509\_name, 27
  - listNode\_s, 90
  - mxml\_node\_s, 168
- NO\_CONTENT
  - mtp\_http.h, 378
- node
  - df.h, 290
- node\_data
  - listNode\_s, 90
- nodes
  - mxml\_index\_s, 166
- NON\_AUTHORITATIVE\_INFORMATION
  - mtp\_http.h, 378
- NoState
  - LibMC::MCAgency, 109
- NOT\_ACCEPTABLE
  - mtp\_http.h, 378
- NOT\_FOUND
  - mtp\_http.h, 378
- nr
  - aes\_context, 30
- num
  - fipa\_agent\_identifier\_set\_s, 67
  - fipa\_url\_sequence\_s, 76
- NUM\_MESSAGE\_TYPE
  - message.h, 374
- num\_args
  - command\_s, 50
- num\_attrs
  - mxml\_value\_s, 171
- num\_callbacks
  - mxml-entity.c, 459
- num\_entries
  - df.h, 290
- num\_nodes
  - mxml\_index\_s, 166
- num\_params
  - fipa\_acl\_envelope\_s, 60
- num\_readers
  - mc\_rwlock\_s, 96
- num\_registered
  - barrier\_node\_s, 47
- num\_saved\_variables
  - agent\_task\_s, 44
- num\_waiting
  - barrier\_node\_s, 47
- number
  - fipa\_expression\_s::content\_u, 71
- number\_of\_elements
  - agent\_task\_s, 43
- number\_of\_tasks
  - agent\_datastate\_s, 35
- numTasks
  - LibMC::MCAgent, 145
- O\_BINARY
  - testmxml.c, 498
- offset
  - havege\_state, 83
- oid
  - \_x509\_name, 27
- OID\_PKCS1
  - x509.h, 623
- OID\_PKCS1\_RSA
  - x509.h, 623
- OID\_PKCS9
  - x509.h, 623
- OID\_X520
  - x509.h, 623
- OK
  - mtp\_http.h, 378
- ONE\_ITERATION
  - havege.c, 680
- ontology
  - fipa\_acl\_message\_s, 62
- opad
  - md2\_context, 148
  - md4\_context, 150
  - md5\_context, 152
  - sha1\_context, 179
  - sha2\_context, 181
  - sha4\_context, 183
- opaque
  - hr\_time, 86
  - mxml\_value\_u, 172
- operator IntPtr
  - LibMC::MCAgent, 141
- operator MCAgent
  - LibMC::MCAgent, 141
- options
  - LibMC::MCAgency, 129
- orphan
  - agent\_s, 39
- out\_ctr
  - ssl\_context, 188
- out\_hdr
  - ssl\_context, 188
- out\_left
  - ssl\_context, 189
- out\_msg
  - ssl\_context, 188
- out\_msglen
  - ssl\_context, 189

- out\_msgtype
  - ssl\_context, 189
- out\_uoff
  - ssl\_context, 189
- own\_cert
  - ssl\_context, 190
- own\_key
  - ssl\_context, 189
- owner
  - agent\_s, 39
- P
  - dhm\_context, 55
  - md4.c, 689
  - md5.c, 695
  - rsa\_context, 175
  - sha1.c, 712
  - sha2.c, 719
  - sha4.c, 726
- p
  - \_x509\_buf, 22
  - mpi, 157
- PACKAGE\_STRING
  - winconfig.h, 796
- PACKAGE\_VERSION
  - winconfig.h, 796
- params
  - fipa\_acl\_envelope\_s, 60
- parent
  - mxml\_node\_s, 168
- parse
  - fipa\_message\_string\_s, 72
- PARTIAL\_CONTENT
  - mtp\_http.h, 378
- payload\_encoding
  - fipa\_acl\_Param\_s, 65
- payload\_length
  - fipa\_acl\_Param\_s, 64
- PAYMENT\_REQUIRED
  - mtp\_http.h, 378
- peer\_cert
  - ssl\_context, 190
- peer\_cn
  - ssl\_context, 190
- performative
  - fipa\_acl\_message\_s, 61
- persistent
  - agent\_datastate\_s, 35
  - agent\_task\_s, 43
- PI\_SUBST
  - md2.c, 687
- pk\_oid
  - \_x509\_cert, 24
- PKCS9\_EMAIL
  - x509.h, 623
- PLAINTEXT
  - dh\_server.c, 773
- pmslen
  - ssl\_context, 192
- pool
  - havege\_state, 83
- Port
  - LibMC::MCAgency, 130
- port
  - acc.h, 248
  - host\_id\_s, 85
  - LibMC::MCAgency, 130
  - mc\_platform\_s, 92
- portno
  - agency\_s, 31
  - LibMC::MCAgency::MCAgency\_t, 133
- portnoc
  - agency\_s, 32
  - LibMC::MCAgency::MCAgency\_t, 133
- PRECONDITION\_FAILED
  - mtp\_http.h, 378
- premaster
  - ssl\_context, 192
- prev
  - mxml\_node\_s, 168
- print\_connectlist
  - commands.x.h, 279
- PrintAgentCode
  - LibMC::MCAgent, 142
- printlist\_agents
  - commands.x.h, 279
- process\_command
  - cmd\_prompt.c, 217
  - cmd\_prompt.h, 276
- PROCESSING
  - mtp\_http.h, 378
- protocol
  - fipa\_acl\_message\_s, 62
- PROXY\_AUTHENTICATION\_REQUIRED
  - mtp\_http.h, 378
- PT1
  - havege\_state, 83
- PT2
  - havege\_state, 83
- PTHREAD\_STACK\_SIZE
  - macros.h, 363
- PUT\_UINT32\_BE
  - aes.c, 630
  - des.c, 661
  - sha1.c, 713
  - sha2.c, 719
- PUT\_UINT32\_LE
  - md4.c, 689

- md5.c, 695
- PUT\_UINT64\_BE
  - sha4.c, 726
- Q
  - rsa\_context, 175
- QP
  - rsa\_context, 176
- QUER\_IF
  - message.h, 374
- QUER\_REF
  - message.h, 374
- QUEUE\_TEMPLATE
  - list.h, 433
- quit
  - mc\_platform\_s, 95
- quit\_lock
  - mc\_platform\_s, 95
- R
  - sha1.c, 713
  - sha2.c, 720
- randbytes
  - ssl\_context, 192
- raw
  - \_x509\_cert, 23
- RCON
  - aes.c, 634
- read\_fd
  - ssl\_context, 187
- real
  - mxml\_value\_u, 172
- received
  - fipa\_acl\_Param\_s, 65
- received\_by
  - fipa\_acl\_envelope\_Received\_s, 58
- received\_date
  - fipa\_acl\_envelope\_Received\_s, 58
- received\_from
  - fipa\_acl\_envelope\_Received\_s, 58
- received\_id
  - fipa\_acl\_envelope\_Received\_s, 58
- received\_via
  - fipa\_acl\_envelope\_Received\_s, 58
- receiver
  - fipa\_acl\_message\_s, 61
- receiver\_num
  - fipa\_acl\_message\_s, 61
- register
  - df\_request.x.h, 292
- RegisterService
  - LibMC::MCAgency, 121
- RELAY
  - message.h, 374
- remote\_hostname
  - connection\_s, 51
- Remove
  - data\_structures.h, 285
- RemoveID
  - data\_structures.h, 285
- RemoveName
  - data\_structures.h, 285
- Reply
  - LibMC::MCAclMessage, 100
- reply\_by
  - fipa\_acl\_message\_s, 63
- reply\_to
  - fipa\_acl\_message\_s, 62
- reply\_with
  - fipa\_acl\_message\_s, 63
- REQUEST
  - df.c, 221
  - df.h, 287
  - df\_request.x.h, 292
  - message.h, 374
- REQUEST\_ENCRYPTION\_INITIALIZE
  - message.h, 374
- REQUEST\_TIMEOUT
  - mtp\_http.h, 378
- REQUEST\_URI\_TOO\_LONG
  - mtp\_http.h, 378
- request\_handler\_DEREGISTER
  - df.c, 224
- request\_handler\_REGISTER
  - df.c, 224
- request\_handler\_SEARCH
  - df.c, 224
- request\_handler\_SUBSCRIBE
  - df.c, 224
- request\_list
  - df.h, 290
- REQUESTED\_RANGE\_NOT\_SATISFIABLE
  - mtp\_http.h, 378
- REQUEST\_ENTITY\_TOO\_LARGE
  - mtp\_http.h, 378
- RESET\_CONTENT
  - mtp\_http.h, 378
- ResetSignal
  - LibMC::MCAgency, 118
- resolvers
  - fipa\_agent\_identifier\_s, 66
- response\_code
  - mtp\_http\_s, 162
- response\_string
  - mtp\_http\_s, 162
- ResumeAgency
  - LibMC::MCAgency, 113
- resumed



- ssl\_context, 191
- retain\_order
  - fipa\_agent\_identifier\_set\_s, 67
- RetrieveAgent
  - LibMC::MCAgency, 123
- RetrieveAgentCode
  - LibMC::MCAgent, 142
- RETRY\_WITH
  - mtp\_http.h, 379
- RETURN\_MSG
  - message.h, 374
- return\_code
  - mtp\_http\_s, 161
- return\_data
  - agent\_datastate\_s, 35
  - agent\_s, 40
- RHs
  - des.c, 666
- RN
  - rsa\_context, 176
- rng\_d
  - ssl\_context, 192
- rng\_f
  - ssl\_context, 192
- ROTR
  - sha2.c, 720
  - sha4.c, 727
- ROTR8
  - aes.c, 630
- RP
  - dhm\_context, 56
  - rsa\_context, 176
- RQ
  - rsa\_context, 176
- rsa
  - \_x509\_cert, 24
- rsa.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, 705
  - rsa\_check\_privkey, 705
  - rsa\_check\_pubkey, 705
  - rsa\_free, 705
  - rsa\_gen\_key, 706
  - rsa\_pkcs1\_decrypt, 706
  - rsa\_pkcs1\_encrypt, 706
  - rsa\_pkcs1\_sign, 707
  - rsa\_pkcs1\_verify, 707
  - rsa\_private, 708
  - rsa\_public, 708
  - rsa\_read\_private, 709
  - rsa\_read\_public, 709
  - rsa\_self\_test, 709
  - rsa\_write\_private, 710
  - rsa\_write\_public, 710
- rsa.h
  - ASN1\_HASH\_MDX, 569
  - ASN1\_HASH\_SHA1, 569
  - ERR\_RSA\_BAD\_INPUT\_DATA, 569
  - ERR\_RSA\_INVALID\_PADDING, 569
  - ERR\_RSA\_KEY\_CHK\_FAILED, 569
  - ERR\_RSA\_KEY\_GEN\_FAILED, 569
  - ERR\_RSA\_KEY\_RD\_FAILED, 569
  - ERR\_RSA\_KEY\_WR\_FAILED, 569
  - ERR\_RSA\_PRIVATE\_FAILED, 570
  - ERR\_RSA\_PUBLIC\_FAILED, 570
  - ERR\_RSA\_VERIFY\_FAILED, 570
  - KEY\_LEN, 570
  - rsa\_check\_privkey, 573
  - rsa\_check\_pubkey, 573
  - RSA\_CRYPT, 570
  - RSA\_D, 570
  - RSA\_DP, 570
  - RSA\_DQ, 571
  - RSA\_E, 571
  - rsa\_free, 573
  - rsa\_gen\_key, 573
  - RSA\_MD2, 571
  - RSA\_MD4, 571
  - RSA\_MD5, 571
  - RSA\_N, 571
  - RSA\_P, 572
  - rsa\_pkcs1\_decrypt, 574
  - rsa\_pkcs1\_encrypt, 574
  - rsa\_pkcs1\_sign, 575
  - rsa\_pkcs1\_verify, 575
  - rsa\_private, 576
  - rsa\_public, 576
  - RSA\_Q, 572
  - RSA\_QP, 572
  - RSA\_RAW, 572
  - rsa\_read\_private, 576
  - rsa\_read\_public, 577
  - rsa\_self\_test, 577
  - RSA\_SHA1, 572
  - RSA\_SIGN, 573
  - rsa\_write\_private, 577
  - rsa\_write\_public, 578
- rsa\_check\_privkey
  - dsa.c, 676
  - rsa.c, 705
  - rsa.h, 573
- rsa\_check\_pubkey
  - dsa.c, 676
  - rsa.c, 705
  - rsa.h, 573
- rsa\_context, 174
  - D, 175
  - DP, 175
  - DQ, 175

- E, [175](#)
- len, [174](#)
- N, [174](#)
- P, [175](#)
- Q, [175](#)
- QP, [176](#)
- RN, [176](#)
- RP, [176](#)
- RQ, [176](#)
- ver, [174](#)
- RSA\_CRYPT
  - rsa.h, [570](#)
- RSA\_D
  - rsa.h, [570](#)
- RSA\_DP
  - rsa.h, [570](#)
- RSA\_DQ
  - rsa.h, [571](#)
- RSA\_E
  - rsa.h, [571](#)
- rsa\_free
  - dsa.c, [676](#)
  - rsa.c, [705](#)
  - rsa.h, [573](#)
- rsa\_gen\_key
  - dsa.c, [676](#)
  - rsa.c, [706](#)
  - rsa.h, [573](#)
- rsa\_genkey.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, [776](#)
  - EXPONENT, [776](#)
  - KEY\_SIZE, [776](#)
  - main, [776](#)
- RSA\_MD2
  - rsa.h, [571](#)
- RSA\_MD4
  - rsa.h, [571](#)
- RSA\_MD5
  - rsa.h, [571](#)
- RSA\_N
  - rsa.h, [571](#)
- RSA\_P
  - rsa.h, [572](#)
- rsa\_pkcs1\_decrypt
  - dsa.c, [677](#)
  - rsa.c, [706](#)
  - rsa.h, [574](#)
- rsa\_pkcs1\_encrypt
  - dsa.c, [677](#)
  - rsa.c, [706](#)
  - rsa.h, [574](#)
- rsa\_pkcs1\_sign
  - dsa.c, [677](#)
  - rsa.c, [707](#)
- rsa.h, [575](#)
- rsa\_pkcs1\_verify
  - dsa.c, [678](#)
  - rsa.c, [707](#)
  - rsa.h, [575](#)
- rsa\_private
  - dsa.c, [678](#)
  - rsa.c, [708](#)
  - rsa.h, [576](#)
- rsa\_public
  - dsa.c, [679](#)
  - rsa.c, [708](#)
  - rsa.h, [576](#)
- RSA\_Q
  - rsa.h, [572](#)
- RSA\_QP
  - rsa.h, [572](#)
- RSA\_RAW
  - rsa.h, [572](#)
- rsa\_read\_private
  - rsa.c, [709](#)
  - rsa.h, [576](#)
- rsa\_read\_public
  - rsa.c, [709](#)
  - rsa.h, [577](#)
- rsa\_self\_test
  - dsa.c, [679](#)
  - rsa.c, [709](#)
  - rsa.h, [577](#)
- RSA\_SHA1
  - rsa.h, [572](#)
- RSA\_SIGN
  - rsa.h, [573](#)
- rsa\_sign.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, [777](#)
  - main, [777](#)
- rsa\_verify.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, [778](#)
  - main, [778](#)
- rsa\_write\_private
  - rsa.c, [710](#)
  - rsa.h, [577](#)
- rsa\_write\_public
  - rsa.c, [710](#)
  - rsa.h, [578](#)
- RSb
  - aes.c, [634](#)
- RT0
  - aes.c, [634](#)
- RT1
  - aes.c, [634](#)
- RT2
  - aes.c, [634](#)
- RT3

- aes.c, [634](#)
- run\_lock
  - agent\_s, [40](#)
- Running
  - LibMC::MCAgency, [109](#)
- RWLOCK\_DESTROY
  - macros.h, [363](#)
- RWLOCK\_INIT
  - macros.h, [363](#)
- RWLOCK\_RDLOCK
  - macros.h, [363](#)
- RWLOCK\_RDUNLOCK
  - macros.h, [363](#)
- RWLOCK\_T
  - macros.h, [363](#)
- RWLOCK\_WRLOCK
  - macros.h, [364](#)
- RWLOCK\_WRUNLOCK
  - macros.h, [364](#)
- S
  - md4.c, [690](#)
  - md5.c, [696](#)
  - sha1.c, [713](#)
- s
  - mpi, [157](#)
- S0
  - sha2.c, [720](#)
  - sha4.c, [727](#)
- S1
  - sha2.c, [720](#)
  - sha4.c, [727](#)
- S2
  - sha2.c, [720](#)
  - sha4.c, [727](#)
- S3
  - sha2.c, [720](#)
  - sha4.c, [727](#)
- SAFE\_FREE
  - mtp\_http.c, [450](#)
- safe\_strcpy
  - mxmldoc.c, [491](#)
- saved\_variables
  - agent\_task\_s, [44](#)
- SB1
  - des.c, [666](#)
- SB2
  - des.c, [667](#)
- SB3
  - des.c, [667](#)
- SB4
  - des.c, [668](#)
- SB5
  - des.c, [668](#)
- SB6
  - des.c, [668](#)
- SB7
  - des.c, [669](#)
- SB8
  - des.c, [669](#)
- scan\_file
  - mxmldoc.c, [491](#)
- scanf\_argv
  - aescript2.c, [762](#)
- Search
  - data\_structures.h, [285](#)
- search
  - df\_request.x.h, [292](#)
- search\_results
  - df.h, [291](#)
- search\_string
  - df.h, [291](#)
- SearchForService
  - LibMC::MCAgency, [121](#)
- SearchName
  - data\_structures.h, [285](#)
- sec
  - \_x509\_time, [28](#)
- second
  - fipa\_DateTime\_s, [68](#)
- selftest.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, [788](#)
  - main, [788](#)
- sem
  - syncListNode\_s, [197](#)
- SEMAPHORE\_DESTROY
  - macros.h, [364](#)
- SEMAPHORE\_INIT
  - macros.h, [364](#)
- SEMAPHORE\_POST
  - macros.h, [364](#)
- SEMAPHORE\_T
  - macros.h, [364](#)
- SEMAPHORE\_WAIT
  - macros.h, [364](#)
- SemaphorePost
  - LibMC::MCAgency, [117](#)
- SemaphoreWait
  - LibMC::MCAgency, [118](#)
- send
  - commands.x.h, [280](#)
- SendAgentMigrationMessage
  - LibMC::MCAgency, [115](#)
- SendAgentMigrationMessageFile
  - LibMC::MCAgency, [114](#)
- sender
  - fipa\_acl\_message\_s, [61](#)
- serial

- [\\_x509\\_cert](#), [23](#)
- server
  - [agency\\_s](#), [31](#)
  - [LibMC::MCAgency::MCAgency\\_t](#), [133](#)
  - [mtp\\_http\\_s](#), [161](#)
- server\_fd
  - [acc.h](#), [248](#)
- SERVER\_NAME
  - [dh\\_client.c](#), [770](#)
  - [ssl\\_client1.c](#), [779](#)
  - [ssl\\_client2.c](#), [781](#)
- server\_name
  - [agent\\_task\\_s](#), [44](#)
- SERVER\_PORT
  - [dh\\_client.c](#), [770](#)
  - [dh\\_server.c](#), [773](#)
  - [ssl\\_client1.c](#), [779](#)
  - [ssl\\_client2.c](#), [781](#)
- serverfd
  - [connection\\_s](#), [51](#)
- service\_list
  - [df.h](#), [291](#)
- service\_types\_e
  - [df.h](#), [287](#)
- ssid
  - [ssl\\_context](#), [191](#)
- session\_table
  - [ssl\\_server.c](#), [784](#)
- set\_alarm
  - [timing.c](#), [750](#)
  - [timing.h](#), [613](#)
- set\_bar
  - [foo\\_c](#), [79](#)
  - [foo\\_s](#), [81](#)
- set\_foo
  - [foo\\_c](#), [79](#)
  - [foo\\_s](#), [82](#)
- set\_foobar
  - [foo\\_c](#), [79](#)
- SET\_THREAD\_OFF
  - [macros.h](#), [364](#)
- SET\_THREAD\_ON
  - [macros.h](#), [364](#)
- set\_timer
  - [timing.c](#), [750](#)
  - [timing.h](#), [614](#)
- SetContent
  - [LibMC::MCAclMessage](#), [102](#)
- SetDefaultAgentStatus
  - [LibMC::MCAgency](#), [114](#)
- SetPerformative
  - [LibMC::MCAclMessage](#), [100](#)
- SetSender
  - [LibMC::MCAclMessage](#), [101](#)
- SetThreadOff
  - [LibMC::MCAgency](#), [113](#)
- SetThreadOn
  - [LibMC::MCAgency](#), [112](#)
- SetThreadsAllOff
  - [LibMC::MCAgency](#), [112](#)
- SetThreadsAllOn
  - [LibMC::MCAgency](#), [112](#)
- SettingChangingEventHandler
  - [LibMC::Properties::Settings](#), [177](#)
- Settings
  - [LibMC::Properties::Settings](#), [177](#)
- SettingsSavingEventHandler
  - [LibMC::Properties::Settings](#), [177](#)
- sha1
  - [sha1.c](#), [713](#)
  - [sha1.h](#), [580](#)
- sha1.c
  - [\\_CRT\\_SECURE\\_NO\\_DEPRECATED](#), [712](#)
  - [\\_sha1\\_src](#), [716](#)
  - [F](#), [712](#)
  - [GET\\_UINT32\\_BE](#), [712](#)
  - [K](#), [712](#)
  - [P](#), [712](#)
  - [PUT\\_UINT32\\_BE](#), [713](#)
  - [R](#), [713](#)
  - [S](#), [713](#)
  - [sha1](#), [713](#)
  - [sha1\\_file](#), [713](#)
  - [sha1\\_finish](#), [714](#)
  - [sha1\\_hmac](#), [714](#)
  - [sha1\\_hmac\\_finish](#), [714](#)
  - [sha1\\_hmac\\_starts](#), [715](#)
  - [sha1\\_hmac\\_update](#), [715](#)
  - [sha1\\_padding](#), [716](#)
  - [sha1\\_process](#), [715](#)
  - [sha1\\_self\\_test](#), [715](#)
  - [sha1\\_starts](#), [716](#)
  - [sha1\\_update](#), [716](#)
- sha1.h
  - [sha1](#), [580](#)
  - [sha1\\_file](#), [580](#)
  - [sha1\\_finish](#), [580](#)
  - [sha1\\_hmac](#), [580](#)
  - [sha1\\_hmac\\_finish](#), [581](#)
  - [sha1\\_hmac\\_starts](#), [581](#)
  - [sha1\\_hmac\\_update](#), [581](#)
  - [sha1\\_self\\_test](#), [582](#)
  - [sha1\\_starts](#), [582](#)
  - [sha1\\_update](#), [582](#)
- sha1\_check
  - [sha1sum.c](#), [766](#)
- sha1\_context, [179](#)
  - [buffer](#), [179](#)

- ipad, 179
- opad, 179
- state, 179
- total, 179
- sha1\_file
  - sha1.c, 713
  - sha1.h, 580
- sha1\_finish
  - sha1.c, 714
  - sha1.h, 580
- sha1\_hmac
  - sha1.c, 714
  - sha1.h, 580
- sha1\_hmac\_finish
  - sha1.c, 714
  - sha1.h, 581
- sha1\_hmac\_starts
  - sha1.c, 715
  - sha1.h, 581
- sha1\_hmac\_update
  - sha1.c, 715
  - sha1.h, 581
- sha1\_padding
  - sha1.c, 716
- sha1\_print
  - sha1sum.c, 766
- sha1\_process
  - sha1.c, 715
- sha1\_self\_test
  - sha1.c, 715
  - sha1.h, 582
- sha1\_starts
  - sha1.c, 716
  - sha1.h, 582
- sha1\_update
  - sha1.c, 716
  - sha1.h, 582
- sha1\_wrapper
  - sha1sum.c, 766
- sha1sum.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, 766
  - main, 766
  - sha1\_check, 766
  - sha1\_print, 766
  - sha1\_wrapper, 766
- sha2
  - sha2.c, 720
  - sha2.h, 585
- sha2.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, 719
  - \_sha2\_src, 724
  - F0, 719
  - F1, 719
  - GET\_UINT32\_BE, 719
  - P, 719
  - PUT\_UINT32\_BE, 719
  - R, 720
  - ROTR, 720
  - S0, 720
  - S1, 720
  - S2, 720
  - S3, 720
  - sha2, 720
  - sha2\_file, 721
  - sha2\_finish, 721
  - sha2\_hmac, 721
  - sha2\_hmac\_finish, 722
  - sha2\_hmac\_starts, 722
  - sha2\_hmac\_update, 722
  - sha2\_padding, 724
  - sha2\_process, 722
  - sha2\_self\_test, 723
  - sha2\_starts, 723
  - sha2\_update, 723
  - SHR, 720
- sha2.h
  - sha2, 585
  - sha2\_file, 585
  - sha2\_finish, 585
  - sha2\_hmac, 585
  - sha2\_hmac\_finish, 586
  - sha2\_hmac\_starts, 586
  - sha2\_hmac\_update, 586
  - sha2\_self\_test, 587
  - sha2\_starts, 587
  - sha2\_update, 587
- sha2\_check
  - sha2sum.c, 768
- sha2\_context, 181
  - buffer, 181
  - ipad, 181
  - is224, 182
  - opad, 181
  - state, 181
  - total, 181
- sha2\_file
  - sha2.c, 721
  - sha2.h, 585
- sha2\_finish
  - sha2.c, 721
  - sha2.h, 585
- sha2\_hmac
  - sha2.c, 721
  - sha2.h, 585
- sha2\_hmac\_finish
  - sha2.c, 722
  - sha2.h, 586
- sha2\_hmac\_starts

- sha2.c, [722](#)
  - sha2.h, [586](#)
- sha2\_hmac\_update
  - sha2.c, [722](#)
  - sha2.h, [586](#)
- sha2\_padding
  - sha2.c, [724](#)
- sha2\_print
  - sha2sum.c, [768](#)
- sha2\_process
  - sha2.c, [722](#)
- sha2\_self\_test
  - sha2.c, [723](#)
  - sha2.h, [587](#)
- sha2\_starts
  - sha2.c, [723](#)
  - sha2.h, [587](#)
- sha2\_update
  - sha2.c, [723](#)
  - sha2.h, [587](#)
- sha2\_wrapper
  - sha2sum.c, [768](#)
- sha2sum.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, [768](#)
  - main, [768](#)
  - sha2\_check, [768](#)
  - sha2\_print, [768](#)
  - sha2\_wrapper, [768](#)
- sha4
  - sha4.c, [727](#)
  - sha4.h, [590](#)
- sha4.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, [726](#)
  - \_sha4\_src, [730](#)
  - F0, [726](#)
  - F1, [726](#)
  - GET\_UINT64\_BE, [726](#)
  - K, [730](#)
  - P, [726](#)
  - PUT\_UINT64\_BE, [726](#)
  - ROTR, [727](#)
  - S0, [727](#)
  - S1, [727](#)
  - S2, [727](#)
  - S3, [727](#)
  - sha4, [727](#)
  - sha4\_file, [727](#)
  - sha4\_finish, [728](#)
  - sha4\_hmac, [728](#)
  - sha4\_hmac\_finish, [728](#)
  - sha4\_hmac\_starts, [729](#)
  - sha4\_hmac\_update, [729](#)
  - sha4\_padding, [730](#)
  - sha4\_process, [729](#)
  - sha4\_self\_test, [729](#)
  - sha4\_starts, [730](#)
  - sha4\_update, [730](#)
  - sha4\_update, [730](#)
  - SHR, [727](#)
- sha4.h
  - sha4, [590](#)
  - sha4\_file, [590](#)
  - sha4\_finish, [590](#)
  - sha4\_hmac, [591](#)
  - sha4\_hmac\_finish, [591](#)
  - sha4\_hmac\_starts, [591](#)
  - sha4\_hmac\_update, [592](#)
  - sha4\_self\_test, [592](#)
  - sha4\_starts, [592](#)
  - sha4\_update, [592](#)
  - uint64, [590](#)
  - UL64, [590](#)
- sha4\_context, [183](#)
  - buffer, [183](#)
  - ipad, [183](#)
  - is384, [184](#)
  - opad, [183](#)
  - state, [183](#)
  - total, [183](#)
- sha4\_file
  - sha4.c, [727](#)
  - sha4.h, [590](#)
- sha4\_finish
  - sha4.c, [728](#)
  - sha4.h, [590](#)
- sha4\_hmac
  - sha4.c, [728](#)
  - sha4.h, [591](#)
- sha4\_hmac\_finish
  - sha4.c, [728](#)
  - sha4.h, [591](#)
- sha4\_hmac\_starts
  - sha4.c, [729](#)
  - sha4.h, [591](#)
- sha4\_hmac\_update
  - sha4.c, [729](#)
  - sha4.h, [592](#)
- sha4\_padding
  - sha4.c, [730](#)
- sha4\_process
  - sha4.c, [729](#)
- sha4\_self\_test
  - sha4.c, [729](#)
  - sha4.h, [592](#)
- sha4\_starts
  - sha4.c, [730](#)
  - sha4.h, [592](#)
- sha4\_update
  - sha4.c, [730](#)

- sha4.h, [592](#)
- shelltype
  - LibMC::MCAgency::ChOptions\_t, [132](#)
- SHR
  - sha2.c, [720](#)
  - sha4.c, [727](#)
- sidlen
  - ssl\_context, [191](#)
- sidtable
  - ssl\_context, [191](#)
- sig
  - \_x509\_cert, [25](#)
- sig\_oid1
  - \_x509\_cert, [24](#)
- sig\_oid2
  - \_x509\_cert, [25](#)
- sighandler
  - timing.c, [750](#)
- sign
  - fipa\_DateTime\_s, [69](#)
- SIGNAL
  - macros.h, [365](#)
- signalled
  - syncListNode\_s, [197](#)
- size
  - barrier\_queue\_s, [48](#)
  - df.h, [291](#)
  - dynstring\_s, [57](#)
  - interpreter\_variable\_data\_s, [87](#)
  - list\_s, [89](#)
  - syncList\_s, [196](#)
- size\_of\_element\_array
  - agent\_task\_s, [43](#)
- SLEEP\_QUEUE
  - macros.h, [365](#)
- SLEEP\_RESET
  - macros.h, [365](#)
- small\_prime
  - bignum.c, [654](#)
- SOCKET\_INPUT\_SIZE
  - message.c, [446](#)
  - mtp\_http.h, [377](#)
- SOME
  - df.h, [287](#)
- sort\_node
  - mxmldoc.c, [492](#)
- split\_string
  - cmd\_prompt.c, [217](#)
  - cmd\_prompt.h, [276](#)
- ssl.h
  - ERR\_SSL\_BAD\_HS\_CERTIFICATE, [597](#)
  - ERR\_SSL\_BAD\_HS\_CERTIFICATE\_-  
REQUEST, [597](#)
  - ERR\_SSL\_BAD\_HS\_CERTIFICATE\_-  
VERIFY, [597](#)
  - ERR\_SSL\_BAD\_HS\_CHANGE\_CIPHER\_-  
SPEC, [597](#)
  - ERR\_SSL\_BAD\_HS\_CLIENT\_HELLO, [597](#)
  - ERR\_SSL\_BAD\_HS\_CLIENT\_KEY\_-  
EXCHANGE, [598](#)
  - ERR\_SSL\_BAD\_HS\_FINISHED, [598](#)
  - ERR\_SSL\_BAD\_HS\_SERVER\_HELLO, [598](#)
  - ERR\_SSL\_BAD\_HS\_SERVER\_HELLO\_-  
DONE, [598](#)
  - ERR\_SSL\_BAD\_HS\_SERVER\_KEY\_-  
EXCHANGE, [598](#)
  - ERR\_SSL\_CA\_CHAIN\_REQUIRED, [598](#)
  - ERR\_SSL\_CERTIFICATE\_REQUIRED, [598](#)
  - ERR\_SSL\_CERTIFICATE\_TOO\_LARGE,  
[598](#)
  - ERR\_SSL\_FATAL\_ALERT\_MESSAGE, [598](#)
  - ERR\_SSL\_FEATURE\_UNAVAILABLE, [599](#)
  - ERR\_SSL\_INVALID\_MAC, [599](#)
  - ERR\_SSL\_INVALID\_MODULUS\_SIZE, [599](#)
  - ERR\_SSL\_INVALID\_RECORD, [599](#)
  - ERR\_SSL\_NO\_CIPHER\_CHOSEN, [599](#)
  - ERR\_SSL\_NO\_CLIENT\_CERTIFICATE,  
[599](#)
  - ERR\_SSL\_NO\_SESSION\_FOUND, [599](#)
  - ERR\_SSL\_PEER\_CLOSE\_NOTIFY, [599](#)
  - ERR\_SSL\_PEER\_VERIFY\_FAILED, [599](#)
  - ERR\_SSL\_PRIVATE\_KEY\_REQUIRED, [600](#)
  - ERR\_SSL\_UNEXPECTED\_MESSAGE, [600](#)
  - ERR\_SSL\_UNKNOWN\_CIPHER, [600](#)
  - SSL3\_EDH\_RSA\_DES\_168\_SHA, [600](#)
  - SSL3\_RSA\_DES\_168\_SHA, [600](#)
  - SSL3\_RSA\_RC4\_128\_MD5, [600](#)
  - SSL3\_RSA\_RC4\_128\_SHA, [600](#)
  - SSL\_CERTIFICATE\_REQUEST, [605](#)
  - SSL\_CERTIFICATE\_VERIFY, [605](#)
  - SSL\_CLIENT\_CERTIFICATE, [605](#)
  - SSL\_CLIENT\_CHANGE\_CIPHER\_SPEC,  
[605](#)
  - SSL\_CLIENT\_FINISHED, [605](#)
  - SSL\_CLIENT\_HELLO, [604](#)
  - SSL\_CLIENT\_KEY\_EXCHANGE, [605](#)
  - SSL\_HANDSHAKE\_OVER, [605](#)
  - SSL\_HELLO\_REQUEST, [604](#)
  - SSL\_SERVER\_CERTIFICATE, [604](#)
  - SSL\_SERVER\_CHANGE\_CIPHER\_SPEC,  
[605](#)
  - SSL\_SERVER\_FINISHED, [605](#)
  - SSL\_SERVER\_HELLO, [604](#)
  - SSL\_SERVER\_HELLO\_DONE, [605](#)
  - SSL\_SERVER\_KEY\_EXCHANGE, [605](#)
  - SSL\_ALERT\_CLOSE\_NOTIFY, [600](#)
  - SSL\_ALERT\_FATAL, [600](#)

- SSL\_ALERT\_NO\_CERTIFICATE, 601
- SSL\_ALERT\_WARNING, 601
- SSL\_BUFFER\_LEN, 601
- ssl\_calc\_verify, 605
- ssl\_client\_start, 605
- ssl\_close\_notify, 605
- SSL\_COMPRESS\_NULL, 601
- ssl\_default\_ciphers, 611
- ssl\_derive\_keys, 606
- SSL\_EXPIRATION\_TIME, 601
- ssl\_flush\_output, 606
- ssl\_free, 606
- ssl\_get\_cipher\_name, 606
- ssl\_get\_verify\_result, 606
- ssl\_handshake, 606
- SSL\_HS\_CERTIFICATE, 601
- SSL\_HS\_CERTIFICATE\_REQUEST, 601
- SSL\_HS\_CERTIFICATE\_VERIFY, 601
- SSL\_HS\_CLIENT\_HELLO, 601
- SSL\_HS\_CLIENT\_KEY\_EXCHANGE, 602
- SSL\_HS\_FINISHED, 602
- SSL\_HS\_HELLO\_REQUEST, 602
- SSL\_HS\_SERVER\_HELLO, 602
- SSL\_HS\_SERVER\_HELLO\_DONE, 602
- SSL\_HS\_SERVER\_KEY\_EXCHANGE, 602
- ssl\_init, 607
- SSL\_IS\_CLIENT, 602
- SSL\_IS\_SERVER, 602
- SSL\_MAX\_CONTENT\_LEN, 602
- SSL\_MSG\_ALERT, 603
- SSL\_MSG\_APPLICATION\_DATA, 603
- SSL\_MSG\_CHANGE\_CIPHER\_SPEC, 603
- SSL\_MSG\_HANDSHAKE, 603
- ssl\_parse\_certificate, 607
- ssl\_parse\_change\_cipher\_spec, 607
- ssl\_parse\_finished, 607
- ssl\_read, 607
- ssl\_read\_record, 608
- ssl\_server\_start, 608
- SSL\_SESSION\_TBL\_LEN, 603
- ssl\_set\_authmode, 608
- ssl\_set\_ca\_chain, 609
- ssl\_set\_ciphlist, 609
- ssl\_set\_dhm\_vals, 609
- ssl\_set\_endpoint, 609
- ssl\_set\_io\_files, 609
- ssl\_set\_rng\_func, 609
- ssl\_set\_rsa\_cert, 609
- ssl\_set\_sidtable, 610
- ssl\_states, 604
- SSL\_VERIFY\_NONE, 603
- SSL\_VERIFY\_OPTIONAL, 603
- SSL\_VERIFY\_REQUIRED, 603
- ssl\_write, 610
- ssl\_write\_certificate, 610
- ssl\_write\_change\_cipher\_spec, 610
- ssl\_write\_finished, 610
- ssl\_write\_record, 611
- SSLV3\_MAJOR\_VERSION, 604
- SSLV3\_MINOR\_VERSION, 604
- TLS10\_MINOR\_VERSION, 604
- TLS11\_MINOR\_VERSION, 604
- TLS1\_EDH\_RSA\_AES\_256\_SHA, 604
- TLS1\_RSA\_AES\_256\_SHA, 604
- SSL3\_EDH\_RSA\_DES\_168\_SHA
- ssl.h, 600
- SSL3\_RSA\_DES\_168\_SHA
- ssl.h, 600
- SSL3\_RSA\_RC4\_128\_MD5
- ssl.h, 600
- SSL3\_RSA\_RC4\_128\_SHA
- ssl.h, 600
- SSL\_CERTIFICATE\_REQUEST
- ssl.h, 605
- SSL\_CERTIFICATE\_VERIFY
- ssl.h, 605
- SSL\_CLIENT\_CERTIFICATE
- ssl.h, 605
- SSL\_CLIENT\_CHANGE\_CIPHER\_SPEC
- ssl.h, 605
- SSL\_CLIENT\_FINISHED
- ssl.h, 605
- SSL\_CLIENT\_HELLO
- ssl.h, 604
- SSL\_CLIENT\_KEY\_EXCHANGE
- ssl.h, 605
- SSL\_HANDSHAKE\_OVER
- ssl.h, 605
- SSL\_HELLO\_REQUEST
- ssl.h, 604
- SSL\_SERVER\_CERTIFICATE
- ssl.h, 604
- SSL\_SERVER\_CHANGE\_CIPHER\_SPEC
- ssl.h, 605
- SSL\_SERVER\_FINISHED
- ssl.h, 605
- SSL\_SERVER\_HELLO
- ssl.h, 604
- SSL\_SERVER\_HELLO\_DONE
- ssl.h, 605
- SSL\_SERVER\_KEY\_EXCHANGE
- ssl.h, 605
- SSL\_ALERT\_CLOSE\_NOTIFY
- ssl.h, 600
- SSL\_ALERT\_FATAL
- ssl.h, 600
- SSL\_ALERT\_NO\_CERTIFICATE
- ssl.h, 601



- SSL\_ALERT\_WARNING
  - ssl.h, 601
- SSL\_BUFFER\_LEN
  - ssl.h, 601
- ssl\_calc\_finished
  - ssl\_tls.c, 741
- ssl\_calc\_verify
  - ssl.h, 605
  - ssl\_tls.c, 741
- ssl\_cli.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, 732
  - \_ssl\_cli\_src, 734
  - ssl\_client\_start, 732
  - ssl\_parse\_certificate\_request, 733
  - ssl\_parse\_server\_hello, 733
  - ssl\_parse\_server\_hello\_done, 733
  - ssl\_parse\_server\_key\_exchange, 733
  - ssl\_write\_certificate\_verify, 733
  - ssl\_write\_client\_hello, 734
  - ssl\_write\_client\_key\_exchange, 734
- ssl\_client1.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, 779
  - GET\_REQUEST, 779
  - main, 780
  - SERVER\_NAME, 779
  - SERVER\_PORT, 779
- ssl\_client2.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, 781
  - GET\_REQUEST, 781
  - main, 782
  - SERVER\_NAME, 781
  - SERVER\_PORT, 781
- ssl\_client\_start
  - ssl.h, 605
  - ssl\_cli.c, 732
- ssl\_close\_notify
  - ssl.h, 605
  - ssl\_tls.c, 741
- SSL\_COMPRESS\_NULL
  - ssl.h, 601
- ssl\_context, 185
  - authmode, 190
  - ca\_chain, 190
  - cipher, 193
  - cipherlist, 193
  - client\_auth, 190
  - ctx\_dec, 194
  - ctx\_enc, 194
  - ctxlen, 193
  - dhm\_ctx, 192
  - endpoint, 190
  - hs\_md5, 191
  - hs\_sha1, 192
  - in\_ctr, 187
  - in\_hdr, 187
  - in\_hslen, 188
  - in\_left, 188
  - in\_msg, 187
  - in\_msglen, 188
  - in\_msgtype, 187
  - in\_offt, 187
  - iv\_dec, 194
  - iv\_enc, 194
  - ivlen, 194
  - keylen, 193
  - mac\_dec, 195
  - mac\_enc, 194
  - maclen, 194
  - major\_ver, 186
  - master, 193
  - max\_ver, 186
  - minlen, 193
  - minor\_ver, 186
  - nb\_zero, 188
  - out\_ctr, 188
  - out\_hdr, 188
  - out\_left, 189
  - out\_msg, 188
  - out\_msglen, 189
  - out\_msgtype, 189
  - out\_uoff, 189
  - own\_cert, 190
  - own\_key, 189
  - peer\_cert, 190
  - peer\_cn, 190
  - pmslen, 192
  - premaster, 192
  - randbytes, 192
  - read\_fd, 187
  - resumed, 191
  - rng\_d, 192
  - rng\_f, 192
  - sessid, 191
  - sidlen, 191
  - sidtable, 191
  - state, 186
  - verify\_result, 191
  - write\_fd, 189
- ssl\_decrypt\_buf
  - ssl\_tls.c, 741
- ssl\_default\_ciphers
  - ssl.h, 611
  - ssl\_tls.c, 747
- ssl\_derive\_keys
  - ssl.h, 606
  - ssl\_tls.c, 741
- ssl\_encrypt\_buf
  - ssl\_tls.c, 742

- SSL\_EXPIRATION\_TIME
  - ssl.h, 601
- ssl\_flush\_output
  - ssl.h, 606
  - ssl\_tls.c, 742
- ssl\_free
  - ssl.h, 606
  - ssl\_tls.c, 742
- ssl\_get\_cipher\_name
  - ssl.h, 606
  - ssl\_tls.c, 742
- ssl\_get\_session
  - ssl\_srv.c, 735
- ssl\_get\_verify\_result
  - ssl.h, 606
  - ssl\_tls.c, 742
- ssl\_handshake
  - ssl.h, 606
  - ssl\_tls.c, 743
- SSL\_HS\_CERTIFICATE
  - ssl.h, 601
- SSL\_HS\_CERTIFICATE\_REQUEST
  - ssl.h, 601
- SSL\_HS\_CERTIFICATE\_VERIFY
  - ssl.h, 601
- SSL\_HS\_CLIENT\_HELLO
  - ssl.h, 601
- SSL\_HS\_CLIENT\_KEY\_EXCHANGE
  - ssl.h, 602
- SSL\_HS\_FINISHED
  - ssl.h, 602
- SSL\_HS\_HELLO\_REQUEST
  - ssl.h, 602
- SSL\_HS\_SERVER\_HELLO
  - ssl.h, 602
- SSL\_HS\_SERVER\_HELLO\_DONE
  - ssl.h, 602
- SSL\_HS\_SERVER\_KEY\_EXCHANGE
  - ssl.h, 602
- ssl\_init
  - ssl.h, 607
  - ssl\_tls.c, 743
- SSL\_IS\_CLIENT
  - ssl.h, 602
- SSL\_IS\_SERVER
  - ssl.h, 602
- ssl\_mac\_md5
  - ssl\_tls.c, 743
- ssl\_mac\_shal
  - ssl\_tls.c, 743
- SSL\_MAX\_CONTENT\_LEN
  - ssl.h, 602
- SSL\_MSG\_ALERT
  - ssl.h, 603
- SSL\_MSG\_APPLICATION\_DATA
  - ssl.h, 603
- SSL\_MSG\_CHANGE\_CIPHER\_SPEC
  - ssl.h, 603
- SSL\_MSG\_HANDSHAKE
  - ssl.h, 603
- ssl\_parse\_certificate
  - ssl.h, 607
  - ssl\_tls.c, 743
- ssl\_parse\_certificate\_request
  - ssl\_cli.c, 733
- ssl\_parse\_certificate\_verify
  - ssl\_srv.c, 735
- ssl\_parse\_change\_cipher\_spec
  - ssl.h, 607
  - ssl\_tls.c, 744
- ssl\_parse\_client\_hello
  - ssl\_srv.c, 736
- ssl\_parse\_client\_key\_exchange
  - ssl\_srv.c, 736
- ssl\_parse\_finished
  - ssl.h, 607
  - ssl\_tls.c, 744
- ssl\_parse\_server\_hello
  - ssl\_cli.c, 733
- ssl\_parse\_server\_hello\_done
  - ssl\_cli.c, 733
- ssl\_parse\_server\_key\_exchange
  - ssl\_cli.c, 733
- ssl\_read
  - ssl.h, 607
  - ssl\_tls.c, 744
- ssl\_read\_record
  - ssl.h, 608
  - ssl\_tls.c, 744
- ssl\_server.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, 783
  - dhm\_G, 784
  - dhm\_P, 784
  - HTTP\_RESPONSE, 783
  - main, 784
  - my\_preferred\_ciphers, 784
  - session\_table, 784
- ssl\_server\_start
  - ssl.h, 608
  - ssl\_srv.c, 736
- SSL\_SESSION\_TBL\_LEN
  - ssl.h, 603
- ssl\_set\_authmode
  - ssl.h, 608
  - ssl\_tls.c, 744
- ssl\_set\_ca\_chain
  - ssl.h, 609
  - ssl\_tls.c, 745

- ssl\_set\_ciphlist
  - ssl.h, 609
  - ssl\_tls.c, 745
- ssl\_set\_dhm\_vals
  - ssl.h, 609
  - ssl\_tls.c, 745
- ssl\_set\_endpoint
  - ssl.h, 609
  - ssl\_tls.c, 745
- ssl\_set\_io\_files
  - ssl.h, 609
  - ssl\_tls.c, 745
- ssl\_set\_rng\_func
  - ssl.h, 609
  - ssl\_tls.c, 745
- ssl\_set\_rsa\_cert
  - ssl.h, 609
  - ssl\_tls.c, 746
- ssl\_set\_session
  - ssl\_srv.c, 736
- ssl\_set\_sidtable
  - ssl.h, 610
  - ssl\_tls.c, 746
- ssl\_srv.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, 735
  - \_ssl\_srv\_src, 738
  - ssl\_get\_session, 735
  - ssl\_parse\_certificate\_verify, 735
  - ssl\_parse\_client\_hello, 736
  - ssl\_parse\_client\_key\_exchange, 736
  - ssl\_server\_start, 736
  - ssl\_set\_session, 736
  - ssl\_write\_certificate\_request, 737
  - ssl\_write\_server\_hello, 737
  - ssl\_write\_server\_hello\_done, 737
  - ssl\_write\_server\_key\_exchange, 737
- ssl\_states
  - ssl.h, 604
- ssl\_tls.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, 741
  - \_ssl\_tls\_src, 747
  - ssl\_calc\_finished, 741
  - ssl\_calc\_verify, 741
  - ssl\_close\_notify, 741
  - ssl\_decrypt\_buf, 741
  - ssl\_default\_ciphers, 747
  - ssl\_derive\_keys, 741
  - ssl\_encrypt\_buf, 742
  - ssl\_flush\_output, 742
  - ssl\_free, 742
  - ssl\_get\_cipher\_name, 742
  - ssl\_get\_verify\_result, 742
  - ssl\_handshake, 743
  - ssl\_init, 743
  - ssl\_mac\_md5, 743
  - ssl\_mac\_sha1, 743
  - ssl\_parse\_certificate, 743
  - ssl\_parse\_change\_cipher\_spec, 744
  - ssl\_parse\_finished, 744
  - ssl\_read, 744
  - ssl\_read\_record, 744
  - ssl\_set\_authmode, 744
  - ssl\_set\_ca\_chain, 745
  - ssl\_set\_ciphlist, 745
  - ssl\_set\_dhm\_vals, 745
  - ssl\_set\_endpoint, 745
  - ssl\_set\_io\_files, 745
  - ssl\_set\_rng\_func, 745
  - ssl\_set\_rsa\_cert, 746
  - ssl\_set\_sidtable, 746
  - ssl\_write, 746
  - ssl\_write\_certificate, 746
  - ssl\_write\_change\_cipher\_spec, 746
  - ssl\_write\_finished, 747
  - ssl\_write\_record, 747
  - tls1\_prf, 747
- SSL\_VERIFY\_NONE
  - ssl.h, 603
- SSL\_VERIFY\_OPTIONAL
  - ssl.h, 603
- SSL\_VERIFY\_REQUIRED
  - ssl.h, 603
- ssl\_write
  - ssl.h, 610
  - ssl\_tls.c, 746
- ssl\_write\_certificate
  - ssl.h, 610
  - ssl\_tls.c, 746
- ssl\_write\_certificate\_request
  - ssl\_srv.c, 737
- ssl\_write\_certificate\_verify
  - ssl\_cli.c, 733
- ssl\_write\_change\_cipher\_spec
  - ssl.h, 610
  - ssl\_tls.c, 746
- ssl\_write\_client\_hello
  - ssl\_cli.c, 734
- ssl\_write\_client\_key\_exchange
  - ssl\_cli.c, 734
- ssl\_write\_finished
  - ssl.h, 610
  - ssl\_tls.c, 747
- ssl\_write\_record
  - ssl.h, 611
  - ssl\_tls.c, 747
- ssl\_write\_server\_hello
  - ssl\_srv.c, 737
- ssl\_write\_server\_hello\_done

- ssl\_srv.c, [737](#)
- ssl\_write\_server\_key\_exchange
  - ssl\_srv.c, [737](#)
- SSLV3\_MAJOR\_VERSION
  - ssl.h, [604](#)
- SSLV3\_MINOR\_VERSION
  - ssl.h, [604](#)
- stack\_size
  - agency\_s, [32](#)
  - LibMC::MCAgency::MCAgency\_t, [134](#)
  - LibMC::MCAgency::MCAgencyOptions\_t, [135](#)
  - mc\_platform\_s, [93](#)
  - MCAgencyOptions\_s, [136](#)
- start
  - \_hr\_time, [21](#)
- State
  - LibMC::MCAgency, [131](#)
- state
  - LibMC::MCAgency, [130](#)
  - md2\_context, [148](#)
  - md4\_context, [150](#)
  - md5\_context, [152](#)
  - sha1\_context, [179](#)
  - sha2\_context, [181](#)
  - sha4\_context, [183](#)
  - ssl\_context, [186](#)
- STATE\_C\_COMMENT
  - mxmldoc.c, [490](#)
- STATE\_CHARACTER
  - mxmldoc.c, [490](#)
- STATE\_CXX\_COMMENT
  - mxmldoc.c, [490](#)
- STATE\_IDENTIFIER
  - mxmldoc.c, [490](#)
- STATE\_NONE
  - mxmldoc.c, [491](#)
- STATE\_PREPROCESSOR
  - mxmldoc.c, [491](#)
- STATE\_STRING
  - mxmldoc.c, [491](#)
- status
  - LibMC::MCAgent, [146](#)
- Steer
  - LibMC::MCAgency, [120](#)
- SteerControl
  - LibMC::MCAgency, [120](#)
- str
  - fipa\_number\_s, [73](#)
  - fipa\_url\_s, [75](#)
- string
  - fipa\_expression\_s::content\_u, [71](#)
  - mxmldoc\_text\_s, [170](#)
- strtok\_r
  - winconfig.h, [796](#)
- STRUCT
  - acc.h, [247](#)
  - ams.h, [268](#)
  - df.h, [289](#), [290](#)
  - macros.h, [365](#)
  - xml\_parser.h, [390](#)
- struct.cxx
  - foo\_t, [497](#)
- subject
  - \_x509\_cert, [24](#)
- subject\_id
  - \_x509\_cert, [25](#)
- subject\_raw
  - \_x509\_cert, [24](#)
- SUBSCRIBE
  - message.h, [374](#)
- SWAP
  - havege.c, [680](#)
- SWITCHING\_PROTOCOLS
  - mtp\_http.h, [378](#)
- sync\_list.c
  - syncListAddNode, [440](#)
  - syncListDelete, [440](#)
  - syncListFind, [440](#)
  - syncListInit, [440](#)
  - syncListNew, [441](#)
  - syncListNodeDestroy, [441](#)
  - syncListNodeInit, [441](#)
  - syncListNodeNew, [441](#)
  - syncListRemove, [441](#)
- sync\_list.h
  - syncList\_p, [442](#)
  - syncList\_t, [442](#)
  - syncListAddNode, [443](#)
  - syncListDelete, [443](#)
  - syncListFind, [443](#)
  - syncListGet, [443](#)
  - syncListInit, [443](#)
  - syncListNew, [443](#)
  - syncListNode\_p, [442](#)
  - syncListNode\_t, [442](#)
  - syncListNodeDestroy, [443](#)
  - syncListNodeInit, [444](#)
  - syncListNodeNew, [444](#)
  - syncListRemove, [444](#)
- SyncDelete
  - LibMC::MCAgency, [118](#)
- SyncInit
  - LibMC::MCAgency, [119](#)
- syncList
  - mc\_platform\_s, [93](#)
- syncList\_p
  - sync\_list.h, [442](#)

- syncList\_s, 196
  - giant\_lock, 196
  - list, 196
  - lock, 196
  - size, 196
- syncList\_t
  - sync\_list.h, 442
- syncListAddNode
  - sync\_list.c, 440
  - sync\_list.h, 443
- syncListDelete
  - sync\_list.c, 440
  - sync\_list.h, 443
- syncListFind
  - sync\_list.c, 440
  - sync\_list.h, 443
- syncListGet
  - sync\_list.h, 443
- syncListInit
  - sync\_list.c, 440
  - sync\_list.h, 443
- syncListNew
  - sync\_list.c, 441
  - sync\_list.h, 443
- syncListNode\_p
  - sync\_list.h, 442
- syncListNode\_s, 197
  - cond, 197
  - id, 197
  - lock, 197
  - sem, 197
  - signalled, 197
- syncListNode\_t
  - sync\_list.h, 442
- syncListNodeDestroy
  - sync\_list.c, 441
  - sync\_list.h, 443
- syncListNodeInit
  - sync\_list.c, 441
  - sync\_list.h, 444
- syncListNodeNew
  - sync\_list.c, 441
  - sync\_list.h, 444
- syncListRemove
  - sync\_list.c, 441
  - sync\_list.h, 444
- System, 15
- System::Collections::Generic, 16
- System::Reflection, 17
- System::Runtime::CompilerServices, 18
- System::Runtime::InteropServices, 19
- System::Text, 20
- t\_dbl
  - bignum.h, 521
- t\_int
  - bignum.h, 521
- tag
  - \_x509\_buf, 22
- target
  - message\_s, 156
  - mtp\_http\_s, 161
- task\_completed
  - agent\_task\_s, 43
- task\_progress
  - agent\_datastate\_s, 35
- tasks
  - agent\_datastate\_s, 34
- tbs
  - \_x509\_cert, 23
- TerminateAgent
  - LibMC::MCAgent, 142
- test\_ca\_crt
  - certs.c, 655
  - certs.h, 535
- test\_ca\_key
  - certs.c, 655
  - certs.h, 535
- test\_ca\_pwd
  - certs.c, 655
  - certs.h, 535
- test\_cli\_crt
  - certs.c, 656
  - certs.h, 535
- test\_cli\_key
  - certs.c, 656
  - certs.h, 535
- test\_srv\_crt
  - certs.c, 657
  - certs.h, 535
- test\_srv\_key
  - certs.c, 657
  - certs.h, 536
- testmxml.c
  - main, 498
  - O\_BINARY, 498
  - type\_cb, 498
  - whitespace\_cb, 499
- text
  - mxml\_value\_u, 172
- thread
  - cmd\_prompt\_s, 49
  - df.h, 291
- THREAD\_CANCEL
  - macros.h, 365
- THREAD\_CREATE
  - macros.h, 365
- THREAD\_JOIN

- macros.h, 366
- THREAD\_T
  - macros.h, 366
- threads
  - agency\_s, 32
  - LibMC::MCAgency::MCAgency\_t, 134
  - LibMC::MCAgency::MCAgencyOptions\_t, 135
  - MCAgencyOptions\_s, 136
- timing.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, 749
  - \_timing\_src, 750
  - alarmed, 750
  - hardclock, 750
  - hardclock\_init, 750
  - set\_alarm, 750
  - set\_timer, 750
  - sighandler, 750
  - tv\_init, 751
- timing.h
  - alarmed, 614
  - hardclock, 613
  - set\_alarm, 613
  - set\_timer, 614
- TLS10\_MINOR\_VERSION
  - ssl.h, 604
- TLS11\_MINOR\_VERSION
  - ssl.h, 604
- TLS1\_EDH\_RSA\_AES\_256\_SHA
  - ssl.h, 604
- tls1\_prf
  - ssl\_tls.c, 747
- TLS1\_RSA\_AES\_256\_SHA
  - ssl.h, 604
- to
  - fipa\_acl\_Param\_s, 64
- to\_address
  - message\_s, 156
- ToString
  - LibMC::MCAgent, 141
- total
  - md4\_context, 150
  - md5\_context, 152
  - sha1\_context, 179
  - sha2\_context, 181
  - sha4\_context, 183
- TST1\_ENTER
  - havege.c, 680
- TST1\_LEAVE
  - havege.c, 681
- TST2\_ENTER
  - havege.c, 681
- TST2\_LEAVE
  - havege.c, 681
- tv\_init
  - timing.c, 751
- type
  - fipa\_expression\_s, 70
  - LibMC::MCAgent, 146
  - mxml\_node\_s, 168
- type\_cb
  - testmxml.c, 498
- TYPES
  - df.h, 287
- uint32
  - aes.c, 631
- uint64
  - sha4.h, 590
- uint8
  - aes.c, 631
- UL64
  - sha4.h, 590
- UNAUTHORIZED
  - mtp\_http.h, 378
- UNORDERED\_COLLECTION
  - mtp\_http.h, 379
- UNPROCESSABLE\_ENTITY
  - mtp\_http.h, 379
- UNSUPPORTED\_MEDIA\_TYPE
  - mtp\_http.h, 378
- update\_comment
  - mxmldoc.c, 492
- update\_name
  - message\_s, 155
- update\_num
  - message\_s, 155
- UPGRADE\_REQUIRED
  - mtp\_http.h, 379
- urls
  - fipa\_url\_sequence\_s, 76
- USAGE
  - aescrypt2.c, 761
- user\_agent
  - mtp\_http\_s, 162
- v3\_ext
  - \_x509\_cert, 25
- val
  - \_x509\_name, 27
- valid\_from
  - \_x509\_cert, 24
- valid\_to
  - \_x509\_cert, 24
- value
  - mxml\_attr\_s, 163
  - mxml\_node\_s, 169
- var\_name

- agent\_task\_s, [44](#)
- vcnet/config.h
  - HAVE\_STRDUP, [454](#)
  - mxmldoc.c, [454](#)
  - MXML\_VERSION, [454](#)
  - mxmldoc.c, [454](#)
  - vsnprintf, [454](#)
- vcnet2005/config.h
  - HAVE\_STRDUP, [455](#)
  - mxmldoc.c, [455](#)
  - MXML\_VERSION, [455](#)
  - mxmldoc.c, [455](#)
  - vsnprintf, [455](#)
- ver
  - rsa\_context, [174](#)
- verify\_result
  - ssl\_context, [191](#)
- version
  - \_x509\_cert, [23](#)
- vsnprintf
  - vcnet/config.h, [454](#)
  - vcnet2005/config.h, [455](#)
- WaitAgent
  - LibMC::MCAgency, [114](#)
- waiting
  - df.h, [291](#)
- waiting\_cond
  - df.h, [291](#)
- waiting\_lock
  - df.h, [291](#)
- WaitRetrieveAgent
  - LibMC::MCAgency, [114](#)
- WaitSignal
  - LibMC::MCAgency, [119](#)
- WAKE\_QUEUE
  - macros.h, [366](#)
- WALK
  - havege\_state, [83](#)
- WARN
  - macros.h, [366](#)
- whitespace
  - mxmldoc.c, [170](#)
- whitespace\_cb
  - testmxml.c, [499](#)
  - xml\_helper.c, [801](#)
  - xml\_helper.h, [385](#)
- winconfig.h
  - PACKAGE\_STRING, [796](#)
  - PACKAGE\_VERSION, [796](#)
  - strtok\_r, [796](#)
- word
  - fipa\_expression\_s::content\_u, [71](#)
- write\_documentation
  - mxmldoc.c, [492](#)
- write\_element
  - mxmldoc.c, [492](#)
- write\_fd
  - ssl\_context, [189](#)
- write\_flag
  - mc\_rwlock\_s, [96](#)
- write\_request
  - mc\_rwlock\_s, [96](#)
- write\_string
  - mxmldoc.c, [492](#)
- ws\_cb
  - mxmldoc.c, [493](#)
- X
  - dhm\_context, [55](#)
- x
  - arc4\_context, [46](#)
- x509.h
  - ASN1\_BIT\_STRING, [617](#)
  - ASN1\_BMP\_STRING, [617](#)
  - ASN1\_BOOLEAN, [617](#)
  - ASN1\_CONSTRUCTED, [617](#)
  - ASN1\_CONTEXT\_SPECIFIC, [617](#)
  - ASN1\_IA5\_STRING, [618](#)
  - ASN1\_INTEGER, [618](#)
  - ASN1\_NULL, [618](#)
  - ASN1\_OCTET\_STRING, [618](#)
  - ASN1\_OID, [618](#)
  - ASN1\_PRIMITIVE, [618](#)
  - ASN1\_PRINTABLE\_STRING, [618](#)
  - ASN1\_SEQUENCE, [618](#)
  - ASN1\_SET, [619](#)
  - ASN1\_T61\_STRING, [619](#)
  - ASN1\_UNIVERSAL\_STRING, [619](#)
  - ASN1\_UTC\_TIME, [619](#)
  - ASN1\_UTF8\_STRING, [619](#)
  - BADCERT\_CN\_MISMATCH, [619](#)
  - BADCERT\_HAS\_EXPIRED, [619](#)
  - BADCERT\_NOT\_TRUSTED, [619](#)
  - ERR\_ASN1\_INVALID\_DATA, [619](#)
  - ERR\_ASN1\_INVALID\_LENGTH, [620](#)
  - ERR\_ASN1\_LENGTH\_MISMATCH, [620](#)
  - ERR\_ASN1\_OUT\_OF\_DATA, [620](#)
  - ERR\_ASN1\_UNEXPECTED\_TAG, [620](#)
  - ERR\_X509\_CERT\_INVALID\_ALG, [620](#)
  - ERR\_X509\_CERT\_INVALID\_DATE, [620](#)
  - ERR\_X509\_CERT\_INVALID\_-  
EXTENSIONS, [620](#)
  - ERR\_X509\_CERT\_INVALID\_FORMAT, [620](#)
  - ERR\_X509\_CERT\_INVALID\_NAME, [621](#)
  - ERR\_X509\_CERT\_INVALID\_PEM, [621](#)
  - ERR\_X509\_CERT\_INVALID\_PUBKEY, [621](#)
  - ERR\_X509\_CERT\_INVALID\_SERIAL, [621](#)

- ERR\_X509\_CERT\_INVALID\_SIGNATURE, 621
- ERR\_X509\_CERT\_INVALID\_VERSION, 621
- ERR\_X509\_CERT\_SIG\_MISMATCH, 621
- ERR\_X509\_CERT\_UNKNOWN\_PK\_ALG, 621
- ERR\_X509\_CERT\_UNKNOWN\_SIG\_ALG, 621
- ERR\_X509\_CERT\_UNKNOWN\_VERSION, 622
- ERR\_X509\_FEATURE\_UNAVAILABLE, 622
- ERR\_X509\_KEY\_INVALID\_ENC\_IV, 622
- ERR\_X509\_KEY\_INVALID\_FORMAT, 622
- ERR\_X509\_KEY\_INVALID\_PEM, 622
- ERR\_X509\_KEY\_INVALID\_VERSION, 622
- ERR\_X509\_KEY\_PASSWORD\_-  
MISMATCH, 622
- ERR\_X509\_KEY\_PASSWORD\_-  
REQUIRED, 622
- ERR\_X509\_KEY\_UNKNOWN\_ENC\_ALG, 622
- ERR\_X509\_SIG\_VERIFY\_FAILED, 623
- OID\_PKCS1, 623
- OID\_PKCS1\_RSA, 623
- OID\_PKCS9, 623
- OID\_X520, 623
- PKCS9\_EMAIL, 623
- x509\_add\_certs, 624
- x509\_buf, 624
- x509\_cert, 624
- x509\_cert\_info, 625
- x509\_dn\_gets, 625
- x509\_free\_cert, 625
- x509\_is\_cert\_expired, 625
- x509\_name, 624
- x509\_parse\_key, 626
- x509\_read\_crtfile, 626
- x509\_read\_keyfile, 626
- x509\_self\_test, 627
- x509\_time, 624
- x509\_verify\_cert, 627
- X520\_COMMON\_NAME, 623
- X520\_COUNTRY, 623
- X520\_LOCALITY, 623
- X520\_ORG\_UNIT, 624
- X520\_ORGANIZATION, 624
- X520\_STATE, 624
- x509\_add\_certs
  - x509.h, 624
  - x509read.c, 754
- x509\_buf
  - x509.h, 624
- x509\_cert
  - x509.h, 624
- x509\_cert\_info
  - x509.h, 625
  - x509read.c, 755
- x509\_des3\_decrypt
  - x509read.c, 755
- x509\_des3\_getiv
  - x509read.c, 755
- x509\_dn\_gets
  - x509.h, 625
  - x509read.c, 755
- x509\_free\_cert
  - x509.h, 625
  - x509read.c, 755
- x509\_get\_alg
  - x509read.c, 756
- x509\_get\_dates
  - x509read.c, 756
- x509\_get\_ext
  - x509read.c, 756
- x509\_get\_name
  - x509read.c, 756
- x509\_get\_pubkey
  - x509read.c, 756
- x509\_get\_serial
  - x509read.c, 757
- x509\_get\_sig
  - x509read.c, 757
- x509\_get\_uid
  - x509read.c, 757
- x509\_get\_version
  - x509read.c, 757
- x509\_hash
  - x509read.c, 757
- x509\_is\_cert\_expired
  - x509.h, 625
  - x509read.c, 758
- x509\_name
  - x509.h, 624
- x509\_parse\_key
  - x509.h, 626
  - x509read.c, 758
- x509\_read\_crtfile
  - x509.h, 626
  - x509read.c, 758
- x509\_read\_keyfile
  - x509.h, 626
  - x509read.c, 759
- x509\_self\_test
  - x509.h, 627
  - x509read.c, 759
- x509\_time
  - x509.h, 624



- x509\_verify\_cert
  - x509.h, [627](#)
  - x509read.c, [759](#)
- x509read.c
  - \_CRT\_SECURE\_NO\_DEPRECATED, [753](#)
  - \_x509\_read\_src, [760](#)
  - asn1\_get\_bool, [753](#)
  - asn1\_get\_int, [753](#)
  - asn1\_get\_len, [754](#)
  - asn1\_get\_mpi, [754](#)
  - asn1\_get\_tag, [754](#)
  - x509\_add\_certs, [754](#)
  - x509\_cert\_info, [755](#)
  - x509\_des3\_decrypt, [755](#)
  - x509\_des3\_getiv, [755](#)
  - x509\_dn\_gets, [755](#)
  - x509\_free\_cert, [755](#)
  - x509\_get\_alg, [756](#)
  - x509\_get\_dates, [756](#)
  - x509\_get\_ext, [756](#)
  - x509\_get\_name, [756](#)
  - x509\_get\_pubkey, [756](#)
  - x509\_get\_serial, [757](#)
  - x509\_get\_sig, [757](#)
  - x509\_get\_uid, [757](#)
  - x509\_get\_version, [757](#)
  - x509\_hash, [757](#)
  - x509\_is\_cert\_expired, [758](#)
  - x509\_parse\_key, [758](#)
  - x509\_read\_crtfile, [758](#)
  - x509\_read\_keyfile, [759](#)
  - x509\_self\_test, [759](#)
  - x509\_verify\_cert, [759](#)
- X520\_COMMON\_NAME
  - x509.h, [623](#)
- X520\_COUNTRY
  - x509.h, [623](#)
- X520\_LOCALITY
  - x509.h, [623](#)
- X520\_ORG\_UNIT
  - x509.h, [624](#)
- X520\_ORGANIZATION
  - x509.h, [624](#)
- X520\_STATE
  - x509.h, [624](#)
- xml\_agent\_root
  - agent\_datastate\_s, [35](#)
- xml\_compose.c
  - agent\_xml\_compose, [797](#)
  - agent\_xml\_compose\_\_agent\_code, [797](#)
  - agent\_xml\_compose\_\_agent\_data, [797](#)
  - agent\_xml\_compose\_\_create\_row\_nodes, [798](#)
  - agent\_xml\_compose\_\_data, [798](#)
  - agent\_xml\_compose\_\_gaf\_message, [798](#)
  - agent\_xml\_compose\_\_home, [798](#)
  - agent\_xml\_compose\_\_message, [798](#)
  - agent\_xml\_compose\_\_mobile\_agent, [799](#)
  - agent\_xml\_compose\_\_name, [799](#)
  - agent\_xml\_compose\_\_owner, [799](#)
  - agent\_xml\_compose\_\_row, [799](#)
  - agent\_xml\_compose\_\_task, [799](#)
  - agent\_xml\_compose\_\_tasks, [799](#)
- xml\_compose.h
  - agent\_xml\_compose, [382](#)
  - agent\_xml\_compose\_\_agent\_code, [382](#)
  - agent\_xml\_compose\_\_agent\_data, [382](#)
  - agent\_xml\_compose\_\_create\_row\_nodes, [382](#)
  - agent\_xml\_compose\_\_data, [383](#)
  - agent\_xml\_compose\_\_gaf\_message, [383](#)
  - agent\_xml\_compose\_\_home, [383](#)
  - agent\_xml\_compose\_\_message, [383](#)
  - agent\_xml\_compose\_\_mobile\_agent, [383](#)
  - agent\_xml\_compose\_\_name, [384](#)
  - agent\_xml\_compose\_\_owner, [384](#)
  - agent\_xml\_compose\_\_row, [384](#)
  - agent\_xml\_compose\_\_task, [384](#)
  - agent\_xml\_compose\_\_tasks, [384](#)
- xml\_find\_sibling
  - xml\_helper.c, [801](#)
  - xml\_helper.h, [385](#)
- xml\_get\_cdata
  - xml\_helper.c, [801](#)
  - xml\_helper.h, [385](#)
- xml\_get\_child
  - xml\_helper.c, [801](#)
  - xml\_helper.h, [385](#)
- xml\_get\_deep\_child
  - xml\_helper.c, [802](#)
  - xml\_helper.h, [385](#)
- xml\_get\_element\_name
  - xml\_helper.c, [802](#)
  - xml\_helper.h, [386](#)
- xml\_get\_next\_element
  - xml\_helper.c, [802](#)
  - xml\_helper.h, [386](#)
- xml\_get\_text
  - xml\_helper.c, [802](#)
  - xml\_helper.h, [386](#)
- xml\_helper.c
  - whitespace\_cb, [801](#)
  - xml\_find\_sibling, [801](#)
  - xml\_get\_cdata, [801](#)
  - xml\_get\_child, [801](#)
  - xml\_get\_deep\_child, [802](#)
  - xml\_get\_element\_name, [802](#)
  - xml\_get\_next\_element, [802](#)
  - xml\_get\_text, [802](#)
  - xml\_new\_cdata, [802](#)

- xml\_helper.h
  - whitespace\_cb, 385
  - xml\_find\_sibling, 385
  - xml\_get\_cdata, 385
  - xml\_get\_child, 385
  - xml\_get\_deep\_child, 385
  - xml\_get\_element\_name, 386
  - xml\_get\_next\_element, 386
  - xml\_get\_text, 386
  - xml\_new\_cdata, 386
- xml\_new\_cdata
  - xml\_helper.c, 802
  - xml\_helper.h, 386
- xml\_parser.c
  - \_XOPEN\_SOURCE, 803
  - agent\_return\_xml\_parse, 804
  - agent\_xml\_parse, 804
  - agent\_xml\_parse\_\_agent\_code, 804
  - agent\_xml\_parse\_\_agent\_data, 804
  - agent\_xml\_parse\_\_data, 804
  - agent\_xml\_parse\_\_fill\_row\_data, 804
  - agent\_xml\_parse\_\_home, 805
  - agent\_xml\_parse\_\_mobile\_agent, 805
  - agent\_xml\_parse\_\_name, 805
  - agent\_xml\_parse\_\_owner, 805
  - agent\_xml\_parse\_\_row, 805
  - agent\_xml\_parse\_\_task, 805
  - agent\_xml\_parse\_\_tasks, 806
  - message\_xml\_parse, 806
  - message\_xml\_parse\_\_message, 806
- xml\_parser.h
  - agent\_xml\_parse, 387
  - agent\_xml\_parse\_\_agent\_code, 387
  - agent\_xml\_parse\_\_agent\_data, 387
  - agent\_xml\_parse\_\_data, 388
  - agent\_xml\_parse\_\_fill\_row\_data, 388
  - agent\_xml\_parse\_\_home, 388
  - agent\_xml\_parse\_\_mobile\_agent, 388
  - agent\_xml\_parse\_\_name, 388
  - agent\_xml\_parse\_\_owner, 389
  - agent\_xml\_parse\_\_row, 389
  - agent\_xml\_parse\_\_task, 389
  - agent\_xml\_parse\_\_tasks, 389
  - message\_xml\_parse, 389
  - message\_xml\_parse\_\_message, 389
  - STRUCT, 390
- xml\_payload
  - message\_s, 155
- xml\_root
  - agent\_datastate\_s, 35
  - message\_s, 155
- XTIME
  - aes.c, 631
- xyssl\_ca\_crt
  - certs.c, 658
  - certs.h, 536
- y
  - arc4\_context, 46
- year
  - \_x509\_time, 28
  - fipa\_DateTime\_s, 68
- ZERO
  - df.h, 287