

gsasl

2.2.1

Generated by Doxygen 1.9.1

1 GNU SASL Library	1
1.1 Introduction	1
1.2 Logical overview	2
1.3 Control flow in application using the library	2
1.4 Examples	3
2 Data Structure Index	11
2.1 Data Structures	11
3 File Index	13
3.1 File List	13
4 Data Structure Documentation	17
4.1 <code>_Gsasl_digest_md5_client_state</code> Struct Reference	17
4.1.1 Detailed Description	17
4.1.2 Field Documentation	17
4.1.2.1 challenge	17
4.1.2.2 finish	18
4.1.2.3 kcc	18
4.1.2.4 kcs	18
4.1.2.5 kic	18
4.1.2.6 kis	18
4.1.2.7 readseqnum	18
4.1.2.8 response	19
4.1.2.9 secret	19
4.1.2.10 sendseqnum	19
4.1.2.11 step	19
4.2 <code>_Gsasl_digest_md5_server_state</code> Struct Reference	19
4.2.1 Detailed Description	20
4.2.2 Field Documentation	20
4.2.2.1 challenge	20
4.2.2.2 finish	20
4.2.2.3 kcc	20
4.2.2.4 kcs	20
4.2.2.5 kic	20
4.2.2.6 kis	21
4.2.2.7 readseqnum	21
4.2.2.8 response	21
4.2.2.9 secret	21
4.2.2.10 sendseqnum	21
4.2.2.11 step	21
4.3 <code>_gsasl_gs2_client_state</code> Struct Reference	22
4.3.1 Detailed Description	22

4.3.2 Field Documentation	22
4.3.2.1 cb	22
4.3.2.2 context	22
4.3.2.3 mech_oid	22
4.3.2.4 service	23
4.3.2.5 step	23
4.3.2.6 token	23
4.4 _Gssasl_gs2_server_state Struct Reference	23
4.4.1 Detailed Description	23
4.4.2 Field Documentation	23
4.4.2.1 cb	24
4.4.2.2 client	24
4.4.2.3 context	24
4.4.2.4 cred	24
4.4.2.5 mech_oid	24
4.4.2.6 step	24
4.5 _Gssasl_gssapi_client_state Struct Reference	25
4.5.1 Detailed Description	25
4.5.2 Field Documentation	25
4.5.2.1 context	25
4.5.2.2 qop	25
4.5.2.3 service	25
4.5.2.4 step	25
4.6 _Gssasl_gssapi_server_state Struct Reference	26
4.6.1 Detailed Description	26
4.6.2 Field Documentation	26
4.6.2.1 client	26
4.6.2.2 context	26
4.6.2.3 cred	26
4.6.2.4 step	26
4.7 _Gssasl_login_client_state Struct Reference	27
4.7.1 Detailed Description	27
4.7.2 Field Documentation	27
4.7.2.1 step	27
4.8 _Gssasl_login_server_state Struct Reference	27
4.8.1 Detailed Description	27
4.8.2 Field Documentation	27
4.8.2.1 password	28
4.8.2.2 step	28
4.8.2.3 username	28
4.9 _Gssasl_ntlm_state Struct Reference	28
4.9.1 Detailed Description	28

4.9.2 Field Documentation	28
4.9.2.1 step	28
4.10 digest_md5_challenge Struct Reference	29
4.10.1 Detailed Description	29
4.10.2 Field Documentation	29
4.10.2.1 ciphers	29
4.10.2.2 nonce	29
4.10.2.3 nrealms	29
4.10.2.4 qops	30
4.10.2.5 realms	30
4.10.2.6 servermaxbuf	30
4.10.2.7 stale	30
4.10.2.8 utf8	30
4.11 digest_md5_finish Struct Reference	30
4.11.1 Detailed Description	31
4.11.2 Field Documentation	31
4.11.2.1 rspauth	31
4.12 digest_md5_response Struct Reference	31
4.12.1 Detailed Description	31
4.12.2 Field Documentation	32
4.12.2.1 authzid	32
4.12.2.2 cipher	32
4.12.2.3 clientmaxbuf	32
4.12.2.4 cnonce	32
4.12.2.5 digesturi	32
4.12.2.6 nc	33
4.12.2.7 nonce	33
4.12.2.8 qop	33
4.12.2.9 realm	33
4.12.2.10 response	33
4.12.2.11 username	33
4.12.2.12 utf8	34
4.13 Gsasl Struct Reference	34
4.13.1 Detailed Description	34
4.13.2 Field Documentation	34
4.13.2.1 application_hook	34
4.13.2.2 cb	34
4.13.2.3 client_mechs	35
4.13.2.4 n_client_mechs	35
4.13.2.5 n_server_mechs	35
4.13.2.6 server_mechs	35
4.14 Gsasl_mechanism Struct Reference	35

4.14.1 Detailed Description	35
4.14.2 Field Documentation	36
4.14.2.1 client	36
4.14.2.2 name	36
4.14.2.3 server	36
4.15 Gsasl_mechanism_functions Struct Reference	36
4.15.1 Detailed Description	36
4.15.2 Field Documentation	37
4.15.2.1 decode	37
4.15.2.2 done	37
4.15.2.3 encode	37
4.15.2.4 finish	37
4.15.2.5 init	38
4.15.2.6 start	38
4.15.2.7 step	38
4.16 Gsasl_session Struct Reference	38
4.16.1 Detailed Description	39
4.16.2 Field Documentation	39
4.16.2.1 anonymous_token	39
4.16.2.2 application_hook	39
4.16.2.3 authid	39
4.16.2.4 authzid	40
4.16.2.5 cb_tls_exporter	40
4.16.2.6 cb_tls_unique	40
4.16.2.7 clientp	40
4.16.2.8 ctx	40
4.16.2.9 digest_md5_hashed_password	40
4.16.2.10 gssapi_display_name	41
4.16.2.11 hostname	41
4.16.2.12 mech	41
4.16.2.13 mech_data	41
4.16.2.14 openid20_outcome_data	41
4.16.2.15 openid20_redirect_url	41
4.16.2.16 passcode	42
4.16.2.17 password	42
4.16.2.18 pin	42
4.16.2.19 qop	42
4.16.2.20 qops	42
4.16.2.21 realm	42
4.16.2.22 saml20_idp_identifier	43
4.16.2.23 saml20_redirect_url	43
4.16.2.24 scram_iter	43

4.16.2.25	scram_salt	43
4.16.2.26	scram_salted_password	43
4.16.2.27	scram_serverkey	43
4.16.2.28	scram_storedkey	44
4.16.2.29	service	44
4.16.2.30	suggestedpin	44
4.17	openid20_client_state Struct Reference	44
4.17.1	Detailed Description	44
4.17.2	Field Documentation	44
4.17.2.1	step	44
4.18	openid20_server_state Struct Reference	45
4.18.1	Detailed Description	45
4.18.2	Field Documentation	45
4.18.2.1	allow_error_step	45
4.18.2.2	step	45
4.19	saml20_client_state Struct Reference	45
4.19.1	Detailed Description	45
4.19.2	Field Documentation	46
4.19.2.1	step	46
4.20	saml20_server_state Struct Reference	46
4.20.1	Detailed Description	46
4.20.2	Field Documentation	46
4.20.2.1	step	46
4.21	scram_client_final Struct Reference	46
4.21.1	Detailed Description	47
4.21.2	Field Documentation	47
4.21.2.1	cbind	47
4.21.2.2	nonce	47
4.21.2.3	proof	47
4.22	scram_client_first Struct Reference	47
4.22.1	Detailed Description	48
4.22.2	Field Documentation	48
4.22.2.1	authzid	48
4.22.2.2	cbflag	48
4.22.2.3	cbname	48
4.22.2.4	client_nonce	48
4.22.2.5	username	49
4.23	scram_client_state Struct Reference	49
4.23.1	Detailed Description	49
4.23.2	Field Documentation	49
4.23.2.1	authmessage	49
4.23.2.2	cf	49

4.23.2.3 cfmb	50
4.23.2.4 cl	50
4.23.2.5 hash	50
4.23.2.6 plus	50
4.23.2.7 serversignature	50
4.23.2.8 sf	50
4.23.2.9 sl	51
4.23.2.10 step	51
4.24 scram_server_final Struct Reference	51
4.24.1 Detailed Description	51
4.24.2 Field Documentation	51
4.24.2.1 verifier	51
4.25 scram_server_first Struct Reference	52
4.25.1 Detailed Description	52
4.25.2 Field Documentation	52
4.25.2.1 iter	52
4.25.2.2 nonce	52
4.25.2.3 salt	52
4.26 scram_server_state Struct Reference	53
4.26.1 Detailed Description	53
4.26.2 Field Documentation	53
4.26.2.1 authmessage	53
4.26.2.2 cb	53
4.26.2.3 cbind	54
4.26.2.4 cblen	54
4.26.2.5 cf	54
4.26.2.6 cfmb_str	54
4.26.2.7 cl	54
4.26.2.8 clientproof	54
4.26.2.9 gs2header	55
4.26.2.10 hash	55
4.26.2.11 plus	55
4.26.2.12 serverkey	55
4.26.2.13 sf	55
4.26.2.14 sf_str	55
4.26.2.15 sl	56
4.26.2.16 snonce	56
4.26.2.17 step	56
4.26.2.18 storedkey	56
5 File Documentation	57
5.1 anonymous.h File Reference	57

5.1.1 Macro Definition Documentation	57
5.1.1.1 GSASL_ANONYMOUS_NAME	57
5.1.2 Function Documentation	58
5.1.2.1 _gsasl_anonymous_client_step()	58
5.1.2.2 _gsasl_anonymous_server_step()	58
5.1.3 Variable Documentation	58
5.1.3.1 _gsasl_anonymous_mechanism	58
5.2 base64.c File Reference	58
5.2.1 Function Documentation	59
5.2.1.1 gsasl_base64_from()	59
5.2.1.2 gsasl_base64_to()	59
5.2.1.3 gsasl_hex_from()	60
5.2.1.4 gsasl_hex_to()	60
5.3 callback.c File Reference	61
5.3.1 Function Documentation	61
5.3.1.1 gsasl_callback()	61
5.3.1.2 gsasl_callback_hook_get()	62
5.3.1.3 gsasl_callback_hook_set()	62
5.3.1.4 gsasl_callback_set()	63
5.3.1.5 gsasl_session_hook_get()	63
5.3.1.6 gsasl_session_hook_set()	64
5.4 challenge.c File Reference	64
5.4.1 Macro Definition Documentation	65
5.4.1.1 DIGIT	65
5.4.1.2 NONCELEN	65
5.4.1.3 TEMPLATE	65
5.4.2 Function Documentation	65
5.4.2.1 cram_md5_challenge()	65
5.5 challenge.h File Reference	65
5.5.1 Macro Definition Documentation	66
5.5.1.1 CRAM_MD5_CHALLENGE_LEN	66
5.5.2 Function Documentation	66
5.5.2.1 cram_md5_challenge()	66
5.6 client.c File Reference	66
5.6.1 Function Documentation	66
5.6.1.1 _gsasl_anonymous_client_step()	67
5.7 client.c File Reference	67
5.7.1 Function Documentation	67
5.7.1.1 _gsasl_cram_md5_client_step()	67
5.8 client.c File Reference	68
5.8.1 Macro Definition Documentation	68
5.8.1.1 CNONCE_ENTROPY_BYTES	68

5.8.2 Typedef Documentation	69
5.8.2.1 _Gssasl_digest_md5_client_state	69
5.8.3 Function Documentation	69
5.8.3.1 _gssasl_digest_md5_client_decode()	69
5.8.3.2 _gssasl_digest_md5_client_encode()	69
5.8.3.3 _gssasl_digest_md5_client_finish()	69
5.8.3.4 _gssasl_digest_md5_client_start()	70
5.8.3.5 _gssasl_digest_md5_client_step()	70
5.9 client.c File Reference	70
5.9.1 Function Documentation	70
5.9.1.1 _gssasl_external_client_step()	70
5.10 client.c File Reference	71
5.10.1 Typedef Documentation	71
5.10.1.1 _gssasl_gs2_client_state	71
5.10.2 Function Documentation	71
5.10.2.1 _gssasl_gs2_client_finish()	71
5.10.2.2 _gssasl_gs2_client_start()	72
5.10.2.3 _gssasl_gs2_client_step()	72
5.11 client.c File Reference	72
5.11.1 Typedef Documentation	73
5.11.1.1 _Gssasl_gssapi_client_state	73
5.11.2 Function Documentation	73
5.11.2.1 _gssasl_gssapi_client_decode()	73
5.11.2.2 _gssasl_gssapi_client_encode()	73
5.11.2.3 _gssasl_gssapi_client_finish()	73
5.11.2.4 _gssasl_gssapi_client_start()	74
5.11.2.5 _gssasl_gssapi_client_step()	74
5.12 client.c File Reference	74
5.12.1 Function Documentation	74
5.12.1.1 _gssasl_login_client_finish()	75
5.12.1.2 _gssasl_login_client_start()	75
5.12.1.3 _gssasl_login_client_step()	75
5.13 client.c File Reference	75
5.13.1 Macro Definition Documentation	76
5.13.1.1 ERR_PREFIX	76
5.13.2 Function Documentation	76
5.13.2.1 _gssasl_openid20_client_finish()	76
5.13.2.2 _gssasl_openid20_client_start()	76
5.13.2.3 _gssasl_openid20_client_step()	76
5.14 client.c File Reference	77
5.14.1 Function Documentation	77
5.14.1.1 _gssasl_plain_client_step()	77

5.15 client.c File Reference	77
5.15.1 Function Documentation	78
5.15.1.1 _gsasl_saml20_client_finish()	78
5.15.1.2 _gsasl_saml20_client_start()	78
5.15.1.3 _gsasl_saml20_client_step()	78
5.16 client.c File Reference	78
5.16.1 Macro Definition Documentation	79
5.16.1.1 CNONCE_ENTROPY_BYTES	79
5.16.2 Function Documentation	79
5.16.2.1 _gsasl_scram_client_finish()	79
5.16.2.2 _gsasl_scram_client_step()	79
5.17 client.c File Reference	80
5.17.1 Macro Definition Documentation	80
5.17.1.1 PASSCODE	80
5.17.1.2 PIN	80
5.17.2 Function Documentation	80
5.17.2.1 _gsasl_securid_client_finish()	81
5.17.2.2 _gsasl_securid_client_start()	81
5.17.2.3 _gsasl_securid_client_step()	81
5.18 cram-md5.h File Reference	81
5.18.1 Macro Definition Documentation	82
5.18.1.1 GSASL_CRAM_MD5_NAME	82
5.18.2 Function Documentation	82
5.18.2.1 _gsasl_cram_md5_client_step()	82
5.18.2.2 _gsasl_cram_md5_server_finish()	82
5.18.2.3 _gsasl_cram_md5_server_start()	82
5.18.2.4 _gsasl_cram_md5_server_step()	83
5.18.3 Variable Documentation	83
5.18.3.1 _gsasl_cram_md5_mechanism	83
5.19 crypto.c File Reference	83
5.19.1 Macro Definition Documentation	84
5.19.1.1 CLIENT_KEY	84
5.19.1.2 SERVER_KEY	84
5.19.2 Function Documentation	84
5.19.2.1 gsasl_hash_length()	84
5.19.2.2 gsasl_nonce()	84
5.19.2.3 gsasl_random()	85
5.19.2.4 gsasl_scram_secrets_from_password()	85
5.19.2.5 gsasl_scram_secrets_from_salted_password()	86
5.20 digest-md5.h File Reference	87
5.20.1 Macro Definition Documentation	87
5.20.1.1 GSASL_DIGEST_MD5_NAME	87

5.20.2 Function Documentation	87
5.20.2.1 _gsasl_digest_md5_client_decode()	88
5.20.2.2 _gsasl_digest_md5_client_encode()	88
5.20.2.3 _gsasl_digest_md5_client_finish()	88
5.20.2.4 _gsasl_digest_md5_client_start()	88
5.20.2.5 _gsasl_digest_md5_client_step()	88
5.20.2.6 _gsasl_digest_md5_server_decode()	89
5.20.2.7 _gsasl_digest_md5_server_encode()	89
5.20.2.8 _gsasl_digest_md5_server_finish()	89
5.20.2.9 _gsasl_digest_md5_server_start()	89
5.20.2.10 _gsasl_digest_md5_server_step()	89
5.20.3 Variable Documentation	90
5.20.3.1 _gsasl_digest_md5_mechanism	90
5.21 digest.c File Reference	90
5.21.1 Macro Definition Documentation	90
5.21.1.1 HEXCHAR	90
5.21.2 Function Documentation	90
5.21.2.1 cram_md5_digest()	91
5.22 digest.h File Reference	91
5.22.1 Macro Definition Documentation	91
5.22.1.1 CRAM_MD5_DIGEST_LEN	91
5.22.2 Function Documentation	91
5.22.2.1 cram_md5_digest()	91
5.23 digestmac.c File Reference	92
5.23.1 Macro Definition Documentation	92
5.23.1.1 A2_POST	92
5.23.1.2 A2_PRE	93
5.23.1.3 COLON	93
5.23.1.4 DERIVE_CLIENT_CONFIDENTIALITY_KEY_STRING	93
5.23.1.5 DERIVE_CLIENT_CONFIDENTIALITY_KEY_STRING_LEN	93
5.23.1.6 DERIVE_CLIENT_INTEGRITY_KEY_STRING	93
5.23.1.7 DERIVE_CLIENT_INTEGRITY_KEY_STRING_LEN	93
5.23.1.8 DERIVE_SERVER_CONFIDENTIALITY_KEY_STRING	94
5.23.1.9 DERIVE_SERVER_CONFIDENTIALITY_KEY_STRING_LEN	94
5.23.1.10 DERIVE_SERVER_INTEGRITY_KEY_STRING	94
5.23.1.11 DERIVE_SERVER_INTEGRITY_KEY_STRING_LEN	94
5.23.1.12 HEXCHAR	94
5.23.1.13 MD5LEN	94
5.23.1.14 QOP_AUTH	95
5.23.1.15 QOP_AUTH_CONF	95
5.23.1.16 QOP_AUTH_INT	95
5.23.2 Function Documentation	95

5.23.2.1 digest_md5_hmac()	95
5.24 digestmac.h File Reference	95
5.24.1 Function Documentation	96
5.24.1.1 digest_md5_hmac()	96
5.25 done.c File Reference	96
5.25.1 Function Documentation	96
5.25.1.1 gsasl_done()	96
5.26 doxygen.c File Reference	97
5.27 error.c File Reference	97
5.27.1 Macro Definition Documentation	97
5.27.1.1 _	97
5.27.1.2 ERR	98
5.27.1.3 gettext_noop	98
5.27.1.4 N_	98
5.27.2 Function Documentation	98
5.27.2.1 gsasl_strerror()	98
5.27.2.2 gsasl_strerror_name()	99
5.27.3 Variable Documentation	99
5.27.3.1 description	99
5.27.3.2 name	99
5.27.3.3 rc	99
5.28 external.h File Reference	100
5.28.1 Macro Definition Documentation	100
5.28.1.1 GSASL_EXTERNAL_NAME	100
5.28.2 Function Documentation	100
5.28.2.1 _gsasl_external_client_step()	100
5.28.2.2 _gsasl_external_server_step()	101
5.28.3 Variable Documentation	101
5.28.3.1 _gsasl_external_mechanism	101
5.29 free.c File Reference	101
5.29.1 Function Documentation	101
5.29.1.1 digest_md5_free_challenge()	101
5.29.1.2 digest_md5_free_finish()	102
5.29.1.3 digest_md5_free_response()	102
5.30 free.c File Reference	102
5.30.1 Function Documentation	102
5.30.1.1 gsasl_free()	102
5.31 free.h File Reference	103
5.31.1 Function Documentation	103
5.31.1.1 digest_md5_free_challenge()	103
5.31.1.2 digest_md5_free_finish()	103
5.31.1.3 digest_md5_free_response()	103

5.32	getsubopt.c File Reference	104
5.32.1	Function Documentation	104
5.32.1.1	digest_md5_getsubopt()	104
5.33	gs2.h File Reference	104
5.33.1	Macro Definition Documentation	105
5.33.1.1	GSASL_GS2_KRB5_NAME	105
5.33.2	Function Documentation	105
5.33.2.1	_gsasl_gs2_client_finish()	105
5.33.2.2	_gsasl_gs2_client_start()	105
5.33.2.3	_gsasl_gs2_client_step()	105
5.33.2.4	_gsasl_gs2_server_finish()	106
5.33.2.5	_gsasl_gs2_server_start()	106
5.33.2.6	_gsasl_gs2_server_step()	106
5.33.3	Variable Documentation	106
5.33.3.1	_gsasl_gs2_krb5_mechanism	106
5.34	gs2helper.c File Reference	106
5.34.1	Function Documentation	107
5.34.1.1	gs2_get_oid()	107
5.35	gs2helper.h File Reference	107
5.35.1	Function Documentation	107
5.35.1.1	gs2_get_oid()	107
5.36	gsasl-mech.h File Reference	107
5.36.1	Typedef Documentation	108
5.36.1.1	Gsasl_code_function	108
5.36.1.2	Gsasl_done_function	109
5.36.1.3	Gsasl_finish_function	109
5.36.1.4	Gsasl_init_function	109
5.36.1.5	Gsasl_mechanism	110
5.36.1.6	Gsasl_mechanism_functions	110
5.36.1.7	Gsasl_start_function	110
5.36.1.8	Gsasl_step_function	111
5.36.2	Function Documentation	111
5.36.2.1	gsasl_register()	111
5.37	gsasl-version.h File Reference	112
5.37.1	Macro Definition Documentation	112
5.37.1.1	GSASL_VERSION	112
5.37.1.2	GSASL_VERSION_MAJOR	112
5.37.1.3	GSASL_VERSION_MINOR	113
5.37.1.4	GSASL_VERSION_NUMBER	113
5.37.1.5	GSASL_VERSION_PATCH	113
5.38	gsasl.h File Reference	114
5.38.1	Macro Definition Documentation	116

5.38.1.1	<code>_GSASL_API</code>	116
5.38.2	Typedef Documentation	116
5.38.2.1	<code>Gsasl</code>	116
5.38.2.2	<code>Gsasl_callback_function</code>	117
5.38.2.3	<code>Gsasl_session</code>	117
5.38.3	Enumeration Type Documentation	117
5.38.3.1	<code>Gsasl_hash</code>	117
5.38.3.2	<code>Gsasl_hash_length</code>	118
5.38.3.3	<code>Gsasl_mechname_limits</code>	119
5.38.3.4	<code>Gsasl_property</code>	119
5.38.3.5	<code>Gsasl_qop</code>	121
5.38.3.6	<code>Gsasl_rc</code>	121
5.38.3.7	<code>Gsasl_saslprep_flags</code>	124
5.38.4	Function Documentation	124
5.38.4.1	<code>gsasl_base64_from()</code>	124
5.38.4.2	<code>gsasl_base64_to()</code>	125
5.38.4.3	<code>gsasl_callback()</code>	125
5.38.4.4	<code>gsasl_callback_hook_get()</code>	126
5.38.4.5	<code>gsasl_callback_hook_set()</code>	126
5.38.4.6	<code>gsasl_callback_set()</code>	127
5.38.4.7	<code>gsasl_check_version()</code>	127
5.38.4.8	<code>gsasl_client_mechlist()</code>	128
5.38.4.9	<code>gsasl_client_start()</code>	128
5.38.4.10	<code>gsasl_client_suggest_mechanism()</code>	129
5.38.4.11	<code>gsasl_client_support_p()</code>	129
5.38.4.12	<code>gsasl_decode()</code>	129
5.38.4.13	<code>gsasl_done()</code>	130
5.38.4.14	<code>gsasl_encode()</code>	130
5.38.4.15	<code>gsasl_finish()</code>	131
5.38.4.16	<code>gsasl_free()</code>	131
5.38.4.17	<code>gsasl_hash_length()</code>	132
5.38.4.18	<code>gsasl_hex_from()</code>	132
5.38.4.19	<code>gsasl_hex_to()</code>	133
5.38.4.20	<code>gsasl_init()</code>	133
5.38.4.21	<code>gsasl_mechanism_name()</code>	134
5.38.4.22	<code>gsasl_mechanism_name_p()</code>	134
5.38.4.23	<code>gsasl_nonce()</code>	135
5.38.4.24	<code>gsasl_property_fast()</code>	135
5.38.4.25	<code>gsasl_property_free()</code>	135
5.38.4.26	<code>gsasl_property_get()</code>	136
5.38.4.27	<code>gsasl_property_set()</code>	136
5.38.4.28	<code>gsasl_property_set_raw()</code>	137

5.38.4.29	gsasl_random()	138
5.38.4.30	gsasl_saslprep()	138
5.38.4.31	gsasl_scram_secrets_from_password()	138
5.38.4.32	gsasl_scram_secrets_from_salted_password()	139
5.38.4.33	gsasl_server_mechlist()	139
5.38.4.34	gsasl_server_start()	140
5.38.4.35	gsasl_server_support_p()	140
5.38.4.36	gsasl_session_hook_get()	141
5.38.4.37	gsasl_session_hook_set()	141
5.38.4.38	gsasl_simple_getpass()	142
5.38.4.39	gsasl_step()	142
5.38.4.40	gsasl_step64()	143
5.38.4.41	gsasl_strerror()	143
5.38.4.42	gsasl_strerror_name()	144
5.39	init.c File Reference	144
5.39.1	Function Documentation	145
5.39.1.1	gsasl_init()	145
5.40	internal.h File Reference	145
5.41	listmech.c File Reference	146
5.41.1	Function Documentation	146
5.41.1.1	gsasl_client_mechlist()	146
5.41.1.2	gsasl_server_mechlist()	146
5.42	login.h File Reference	147
5.42.1	Macro Definition Documentation	147
5.42.1.1	GSASL_LOGIN_NAME	147
5.42.2	Function Documentation	147
5.42.2.1	_gsasl_login_client_finish()	148
5.42.2.2	_gsasl_login_client_start()	148
5.42.2.3	_gsasl_login_client_step()	148
5.42.2.4	_gsasl_login_server_finish()	148
5.42.2.5	_gsasl_login_server_start()	148
5.42.2.6	_gsasl_login_server_step()	149
5.42.3	Variable Documentation	149
5.42.3.1	_gsasl_login_mechanism	149
5.43	md5pwd.c File Reference	149
5.43.1	Function Documentation	149
5.43.1.1	gsasl_simple_getpass()	149
5.44	mechinfo.c File Reference	150
5.44.1	Variable Documentation	150
5.44.1.1	_gsasl_anonymous_mechanism	150
5.45	mechinfo.c File Reference	151
5.45.1	Variable Documentation	151

5.45.1.1 _gsasl_cram_md5_mechanism	151
5.46 mechinfo.c File Reference	151
5.46.1 Variable Documentation	151
5.46.1.1 _gsasl_digest_md5_mechanism	151
5.47 mechinfo.c File Reference	152
5.47.1 Variable Documentation	152
5.47.1.1 _gsasl_external_mechanism	152
5.48 mechinfo.c File Reference	152
5.48.1 Variable Documentation	152
5.48.1.1 _gsasl_gs2_krb5_mechanism	153
5.49 mechinfo.c File Reference	153
5.49.1 Variable Documentation	153
5.49.1.1 _gsasl_gssapi_mechanism	153
5.50 mechinfo.c File Reference	153
5.50.1 Variable Documentation	153
5.50.1.1 _gsasl_login_mechanism	154
5.51 mechinfo.c File Reference	154
5.51.1 Variable Documentation	154
5.51.1.1 _gsasl_ntlm_mechanism	154
5.52 mechinfo.c File Reference	154
5.52.1 Variable Documentation	154
5.52.1.1 _gsasl_openid20_mechanism	155
5.53 mechinfo.c File Reference	155
5.53.1 Variable Documentation	155
5.53.1.1 _gsasl_plain_mechanism	155
5.54 mechinfo.c File Reference	155
5.54.1 Variable Documentation	156
5.54.1.1 _gsasl_saml20_mechanism	156
5.55 mechinfo.c File Reference	156
5.56 mechinfo.c File Reference	156
5.56.1 Variable Documentation	156
5.56.1.1 _gsasl_securid_mechanism	156
5.57 mechname.c File Reference	157
5.57.1 Function Documentation	157
5.57.1.1 gsasl_mechanism_name()	157
5.58 mechtools.c File Reference	157
5.58.1 Function Documentation	158
5.58.1.1 _gsasl_gs2_generate_header()	158
5.58.1.2 _gsasl_hash()	158
5.58.1.3 _gsasl_hex_decode()	158
5.58.1.4 _gsasl_hex_encode()	159
5.58.1.5 _gsasl_hex_p()	159

5.58.1.6	_gsasl_hmac()	159
5.58.1.7	_gsasl_parse_gs2_header()	159
5.58.1.8	_gsasl_pbkdf2()	160
5.59	mechtools.h File Reference	160
5.59.1	Function Documentation	160
5.59.1.1	_gsasl_gs2_generate_header()	160
5.59.1.2	_gsasl_hash()	161
5.59.1.3	_gsasl_hex_decode()	161
5.59.1.4	_gsasl_hex_encode()	161
5.59.1.5	_gsasl_hex_p()	161
5.59.1.6	_gsasl_hmac()	161
5.59.1.7	_gsasl_parse_gs2_header()	162
5.59.1.8	_gsasl_pbkdf2()	162
5.60	nonascii.c File Reference	162
5.60.1	Function Documentation	162
5.60.1.1	latin1toutf8()	162
5.60.1.2	utf8tolatin1ifpossible()	163
5.61	nonascii.h File Reference	163
5.61.1	Function Documentation	163
5.61.1.1	latin1toutf8()	163
5.61.1.2	utf8tolatin1ifpossible()	163
5.62	ntlm.c File Reference	163
5.62.1	Typedef Documentation	164
5.62.1.1	_Gsasl_ntlm_state	164
5.62.2	Function Documentation	164
5.62.2.1	_gsasl_ntlm_client_finish()	164
5.62.2.2	_gsasl_ntlm_client_start()	164
5.62.2.3	_gsasl_ntlm_client_step()	165
5.63	openid20.h File Reference	165
5.63.1	Macro Definition Documentation	165
5.63.1.1	GSASL_OPENID20_NAME	165
5.63.2	Function Documentation	166
5.63.2.1	_gsasl_openid20_client_finish()	166
5.63.2.2	_gsasl_openid20_client_start()	166
5.63.2.3	_gsasl_openid20_client_step()	166
5.63.2.4	_gsasl_openid20_server_finish()	166
5.63.2.5	_gsasl_openid20_server_start()	166
5.63.2.6	_gsasl_openid20_server_step()	167
5.63.3	Variable Documentation	167
5.63.3.1	_gsasl_openid20_mechanism	167
5.64	parser.c File Reference	167
5.64.1	Macro Definition Documentation	168

5.64.1.1	DEFAULT_ALGORITHM	168
5.64.1.2	DEFAULT_CHARSET	168
5.64.2	Enumeration Type Documentation	168
5.64.2.1	anonymous enum	168
5.64.2.2	anonymous enum	169
5.64.2.3	anonymous enum	169
5.64.2.4	anonymous enum	169
5.64.2.5	anonymous enum	170
5.64.3	Function Documentation	170
5.64.3.1	digest_md5_parse_challenge()	170
5.64.3.2	digest_md5_parse_finish()	170
5.64.3.3	digest_md5_parse_response()	170
5.65	parser.c File Reference	171
5.65.1	Function Documentation	171
5.65.1.1	scram_parse_client_final()	171
5.65.1.2	scram_parse_client_first()	171
5.65.1.3	scram_parse_server_final()	171
5.65.1.4	scram_parse_server_first()	172
5.66	parser.h File Reference	172
5.66.1	Function Documentation	172
5.66.1.1	digest_md5_getsubopt()	172
5.66.1.2	digest_md5_parse_challenge()	172
5.66.1.3	digest_md5_parse_finish()	173
5.66.1.4	digest_md5_parse_response()	173
5.67	parser.h File Reference	173
5.67.1	Function Documentation	173
5.67.1.1	scram_parse_client_final()	173
5.67.1.2	scram_parse_client_first()	174
5.67.1.3	scram_parse_server_final()	174
5.67.1.4	scram_parse_server_first()	174
5.68	plain.h File Reference	174
5.68.1	Macro Definition Documentation	175
5.68.1.1	GSASL_PLAIN_NAME	175
5.68.2	Function Documentation	175
5.68.2.1	_gsasl_plain_client_step()	175
5.68.2.2	_gsasl_plain_server_step()	175
5.68.3	Variable Documentation	175
5.68.3.1	_gsasl_plain_mechanism	176
5.69	printer.c File Reference	176
5.69.1	Function Documentation	176
5.69.1.1	digest_md5_print_challenge()	176
5.69.1.2	digest_md5_print_finish()	176

5.69.1.3 digest_md5_print_response()	176
5.70 printer.c File Reference	177
5.70.1 Function Documentation	177
5.70.1.1 scram_print_client_final()	177
5.70.1.2 scram_print_client_first()	177
5.70.1.3 scram_print_server_final()	177
5.70.1.4 scram_print_server_first()	178
5.71 printer.h File Reference	178
5.71.1 Function Documentation	178
5.71.1.1 digest_md5_print_challenge()	178
5.71.1.2 digest_md5_print_finish()	178
5.71.1.3 digest_md5_print_response()	178
5.72 printer.h File Reference	179
5.72.1 Function Documentation	179
5.72.1.1 scram_print_client_final()	179
5.72.1.2 scram_print_client_first()	179
5.72.1.3 scram_print_server_final()	179
5.72.1.4 scram_print_server_first()	180
5.73 property.c File Reference	180
5.73.1 Function Documentation	180
5.73.1.1 gssasl_property_fast()	180
5.73.1.2 gssasl_property_free()	181
5.73.1.3 gssasl_property_get()	181
5.73.1.4 gssasl_property_set()	182
5.73.1.5 gssasl_property_set_raw()	182
5.74 qop.c File Reference	183
5.74.1 Function Documentation	183
5.74.1.1 digest_md5_qops2qopstr()	183
5.74.1.2 digest_md5_qopstr2qops()	183
5.75 qop.h File Reference	183
5.75.1 Function Documentation	184
5.75.1.1 digest_md5_qops2qopstr()	184
5.75.1.2 digest_md5_qopstr2qops()	184
5.76 register.c File Reference	184
5.76.1 Function Documentation	184
5.76.1.1 gssasl_register()	184
5.77 saml20.h File Reference	185
5.77.1 Macro Definition Documentation	185
5.77.1.1 GSASL_SAML20_NAME	185
5.77.2 Function Documentation	185
5.77.2.1 _gssasl_saml20_client_finish()	186
5.77.2.2 _gssasl_saml20_client_start()	186

5.77.2.3	_gsasl_saml20_client_step()	186
5.77.2.4	_gsasl_saml20_server_finish()	186
5.77.2.5	_gsasl_saml20_server_start()	186
5.77.2.6	_gsasl_saml20_server_step()	187
5.77.3	Variable Documentation	187
5.77.3.1	_gsasl_saml20_mechanism	187
5.78	sasprep.c File Reference	187
5.78.1	Function Documentation	187
5.78.1.1	_gsasl_sasprep()	187
5.79	scram.h File Reference	188
5.80	securid.h File Reference	188
5.80.1	Macro Definition Documentation	188
5.80.1.1	GSASL_SECURID_NAME	188
5.80.2	Function Documentation	188
5.80.2.1	_gsasl_securid_client_finish()	189
5.80.2.2	_gsasl_securid_client_start()	189
5.80.2.3	_gsasl_securid_client_step()	189
5.80.2.4	_gsasl_securid_server_step()	189
5.80.3	Variable Documentation	189
5.80.3.1	_gsasl_securid_mechanism	189
5.81	server.c File Reference	190
5.81.1	Function Documentation	190
5.81.1.1	_gsasl_anonymous_server_step()	190
5.82	server.c File Reference	190
5.82.1	Macro Definition Documentation	191
5.82.1.1	MD5LEN	191
5.82.2	Function Documentation	191
5.82.2.1	_gsasl_cram_md5_server_finish()	191
5.82.2.2	_gsasl_cram_md5_server_start()	191
5.82.2.3	_gsasl_cram_md5_server_step()	191
5.83	server.c File Reference	192
5.83.1	Macro Definition Documentation	192
5.83.1.1	NONCE_ENTROPY_BYTES	193
5.83.2	Typedef Documentation	193
5.83.2.1	_Gsasl_digest_md5_server_state	193
5.83.3	Function Documentation	193
5.83.3.1	_gsasl_digest_md5_server_decode()	193
5.83.3.2	_gsasl_digest_md5_server_encode()	193
5.83.3.3	_gsasl_digest_md5_server_finish()	194
5.83.3.4	_gsasl_digest_md5_server_start()	194
5.83.3.5	_gsasl_digest_md5_server_step()	194
5.84	server.c File Reference	194

5.84.1 Function Documentation	194
5.84.1.1 _gsasl_external_server_step()	195
5.85 server.c File Reference	195
5.85.1 Typedef Documentation	195
5.85.1.1 _Gsasl_gs2_server_state	195
5.85.2 Function Documentation	196
5.85.2.1 _gsasl_gs2_server_finish()	196
5.85.2.2 _gsasl_gs2_server_start()	196
5.85.2.3 _gsasl_gs2_server_step()	196
5.86 server.c File Reference	196
5.86.1 Typedef Documentation	197
5.86.1.1 _Gsasl_gssapi_server_state	197
5.86.2 Function Documentation	197
5.86.2.1 _gsasl_gssapi_server_finish()	197
5.86.2.2 _gsasl_gssapi_server_start()	197
5.86.2.3 _gsasl_gssapi_server_step()	198
5.87 server.c File Reference	198
5.87.1 Macro Definition Documentation	198
5.87.1.1 CHALLENGE_PASSWORD	198
5.87.1.2 CHALLENGE_USERNAME	199
5.87.2 Function Documentation	199
5.87.2.1 _gsasl_login_server_finish()	199
5.87.2.2 _gsasl_login_server_start()	199
5.87.2.3 _gsasl_login_server_step()	199
5.88 server.c File Reference	199
5.88.1 Function Documentation	200
5.88.1.1 _gsasl_openid20_server_finish()	200
5.88.1.2 _gsasl_openid20_server_start()	200
5.88.1.3 _gsasl_openid20_server_step()	200
5.89 server.c File Reference	201
5.89.1 Function Documentation	201
5.89.1.1 _gsasl_plain_server_step()	201
5.90 server.c File Reference	201
5.90.1 Function Documentation	202
5.90.1.1 _gsasl_saml20_server_finish()	202
5.90.1.2 _gsasl_saml20_server_start()	202
5.90.1.3 _gsasl_saml20_server_step()	202
5.91 server.c File Reference	202
5.91.1 Macro Definition Documentation	203
5.91.1.1 DEFAULT_SALT_BYTES	203
5.91.1.2 SNONCE_ENTROPY_BYTES	203
5.91.2 Function Documentation	203

5.91.2.1 _gsasl_scram_server_finish()	203
5.91.2.2 _gsasl_scram_server_step()	204
5.92 server.c File Reference	204
5.92.1 Macro Definition Documentation	204
5.92.1.1 PASSCODE	204
5.92.1.2 PIN	204
5.92.2 Function Documentation	205
5.92.2.1 _gsasl_securid_server_step()	205
5.93 session.c File Reference	205
5.93.1 Macro Definition Documentation	205
5.93.1.1 C2I	206
5.93.1.2 MAC_DATA_LEN	206
5.93.1.3 MAC_HMAC_LEN	206
5.93.1.4 MAC_MSG_TYPE	206
5.93.1.5 MAC_MSG_TYPE_LEN	206
5.93.1.6 MAC_SEQNUM_LEN	207
5.93.1.7 MD5LEN	207
5.93.1.8 SASL_INTEGRITY_PREFIX_LENGTH	207
5.93.2 Function Documentation	207
5.93.2.1 digest_md5_decode()	207
5.93.2.2 digest_md5_encode()	207
5.94 session.h File Reference	208
5.94.1 Function Documentation	208
5.94.1.1 digest_md5_decode()	208
5.94.1.2 digest_md5_encode()	208
5.95 suggest.c File Reference	208
5.95.1 Function Documentation	209
5.95.1.1 gsas_client_suggest_mechanism()	209
5.95.1.2 gsas_mechanism_name_p()	209
5.95.2 Variable Documentation	210
5.95.2.1 _GSASL_VALID_MECHANISM_CHARACTERS	210
5.96 supportp.c File Reference	210
5.96.1 Function Documentation	210
5.96.1.1 gsas_client_support_p()	210
5.96.1.2 gsas_server_support_p()	211
5.97 test-parser.c File Reference	211
5.97.1 Function Documentation	212
5.97.1.1 main()	212
5.98 tokens.c File Reference	212
5.98.1 Function Documentation	212
5.98.1.1 scram_free_client_final()	212
5.98.1.2 scram_free_client_first()	212

5.98.1.3	scram_free_server_final()	213
5.98.1.4	scram_free_server_first()	213
5.99	tokens.h File Reference	213
5.99.1	Macro Definition Documentation	214
5.99.1.1	DIGEST_MD5_LENGTH	214
5.99.1.2	DIGEST_MD5_RESPONSE_LENGTH	214
5.99.2	Typedef Documentation	214
5.99.2.1	digest_md5_challenge	214
5.99.2.2	digest_md5_cipher	214
5.99.2.3	digest_md5_finish	214
5.99.2.4	digest_md5_qop	215
5.99.2.5	digest_md5_response	215
5.99.3	Enumeration Type Documentation	215
5.99.3.1	digest_md5_cipher	215
5.99.3.2	digest_md5_qop	215
5.100	tokens.h File Reference	216
5.100.1	Function Documentation	216
5.100.1.1	scram_free_client_final()	216
5.100.1.2	scram_free_client_first()	216
5.100.1.3	scram_free_server_final()	216
5.100.1.4	scram_free_server_first()	217
5.101	tools.c File Reference	217
5.101.1	Function Documentation	217
5.101.1.1	set_saltedpassword()	217
5.102	tools.h File Reference	217
5.102.1	Function Documentation	217
5.102.1.1	set_saltedpassword()	218
5.103	validate.c File Reference	218
5.103.1	Function Documentation	218
5.103.1.1	digest_md5_validate()	218
5.103.1.2	digest_md5_validate_challenge()	218
5.103.1.3	digest_md5_validate_finish()	219
5.103.1.4	digest_md5_validate_response()	219
5.104	validate.c File Reference	219
5.104.1	Function Documentation	219
5.104.1.1	scram_valid_client_final()	219
5.104.1.2	scram_valid_client_first()	220
5.104.1.3	scram_valid_server_final()	220
5.104.1.4	scram_valid_server_first()	220
5.105	validate.h File Reference	220
5.105.1	Function Documentation	220
5.105.1.1	digest_md5_validate()	220

5.105.1.2 digest_md5_validate_challenge()	221
5.105.1.3 digest_md5_validate_finish()	221
5.105.1.4 digest_md5_validate_response()	221
5.106 validate.h File Reference	221
5.106.1 Function Documentation	221
5.106.1.1 scram_valid_client_final()	221
5.106.1.2 scram_valid_client_first()	222
5.106.1.3 scram_valid_server_final()	222
5.106.1.4 scram_valid_server_first()	222
5.107 version.c File Reference	222
5.107.1 Function Documentation	222
5.107.1.1 gssasl_check_version()	222
5.108 x-gssapi.h File Reference	223
5.108.1 Macro Definition Documentation	223
5.108.1.1 GSASL_GSSAPI_NAME	224
5.108.2 Function Documentation	224
5.108.2.1 _gssasl_gssapi_client_decode()	224
5.108.2.2 _gssasl_gssapi_client_encode()	224
5.108.2.3 _gssasl_gssapi_client_finish()	224
5.108.2.4 _gssasl_gssapi_client_start()	225
5.108.2.5 _gssasl_gssapi_client_step()	225
5.108.2.6 _gssasl_gssapi_server_finish()	225
5.108.2.7 _gssasl_gssapi_server_start()	225
5.108.2.8 _gssasl_gssapi_server_step()	225
5.108.3 Variable Documentation	226
5.108.3.1 _gssasl_gssapi_mechanism	226
5.109 x-ntlm.h File Reference	226
5.109.1 Macro Definition Documentation	226
5.109.1.1 GSASL_NTLM_NAME	226
5.109.2 Function Documentation	226
5.109.2.1 _gssasl_ntlm_client_finish()	227
5.109.2.2 _gssasl_ntlm_client_start()	227
5.109.2.3 _gssasl_ntlm_client_step()	227
5.109.3 Variable Documentation	227
5.109.3.1 _gssasl_ntlm_mechanism	227
5.110 xcode.c File Reference	227
5.110.1 Function Documentation	228
5.110.1.1 gssasl_decode()	228
5.110.1.2 gssasl_encode()	228
5.111 xfinish.c File Reference	229
5.111.1 Function Documentation	229
5.111.1.1 gssasl_finish()	229

5.112 xstart.c File Reference	230
5.112.1 Function Documentation	230
5.112.1.1 gsas_client_start()	230
5.112.1.2 gsas_server_start()	230
5.113 xstep.c File Reference	231
5.113.1 Function Documentation	231
5.113.1.1 gsas_step()	231
5.113.1.2 gsas_step64()	232
Index	233

Chapter 1

GNU SASL Library

1.1 Introduction

GNU SASL is an implementation of the Simple Authentication and Security Layer framework and a few common SASL mechanisms. SASL is used by network servers (e.g., IMAP, SMTP) to request authentication from clients, and in clients to authenticate against servers.

GNU SASL consists of a library ('libgsasl'), a command line utility ('gsasl') to access the library from the shell, and a manual. The library includes support for the framework (with authentication functions and application data privacy and integrity functions) and at least partial support for the CRAM-MD5, EXTERNAL, GSSAPI, ANONYMOUS, PLAIN, SECURID, DIGEST-MD5, LOGIN, and NTLM mechanisms.

The library is easily ported because it does not do network communication by itself, but rather leaves it up to the calling application. The library is flexible with regards to the authorization infrastructure used, as it utilize a callback into the application to decide whether a user is authorized or not.

GNU SASL is developed for the GNU/Linux system, but runs on over 20 platforms including most major Unix platforms and Windows, and many kind of devices including iPAQ handhelds and S/390 mainframes.

GNU SASL is written in pure ANSI C89 to be portable to embedded and otherwise limited platforms. The entire library, with full support for ANONYMOUS, EXTERNAL, PLAIN, LOGIN and CRAM-MD5, and the front-end that support client and server mode, and the IMAP and SMTP protocols, fits in under 60kb on an Intel x86 platform, without any modifications to the code. (This figure was accurate as of version 0.0.13.)

The library is licensed under the GNU Lesser General Public License, and the command-line interface, self-tests and examples are licensed under the GNU General Public License.

The project web page:

<http://www.gnu.org/software/gsas1/>

The software archive:

<ftp://alpha.gnu.org/pub/gnu/gsas1/>

Further information and paid contract development:

Simon Josefsson simon@josefsson.org

1.2 Logical overview

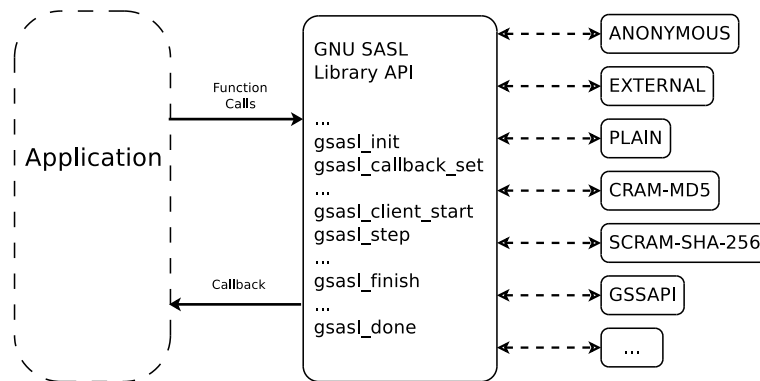


Figure 1.1 Logical overview

1.3 Control flow in application using the library

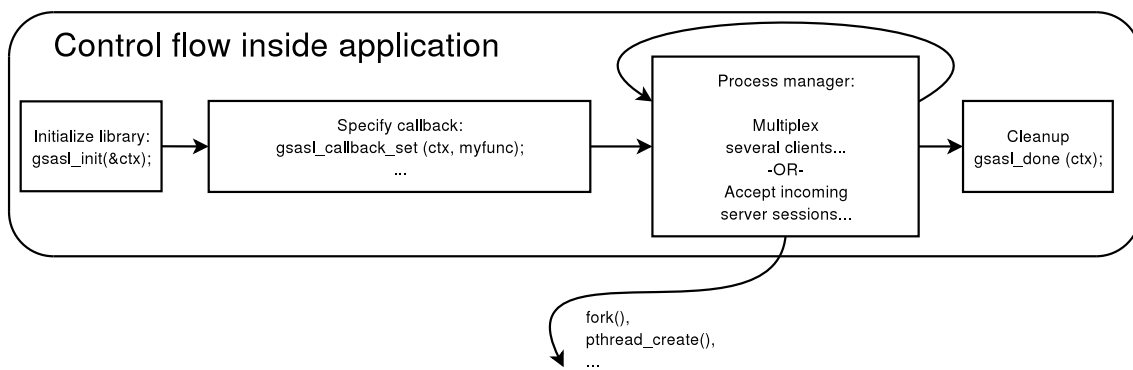


Figure 1.2 Control flow

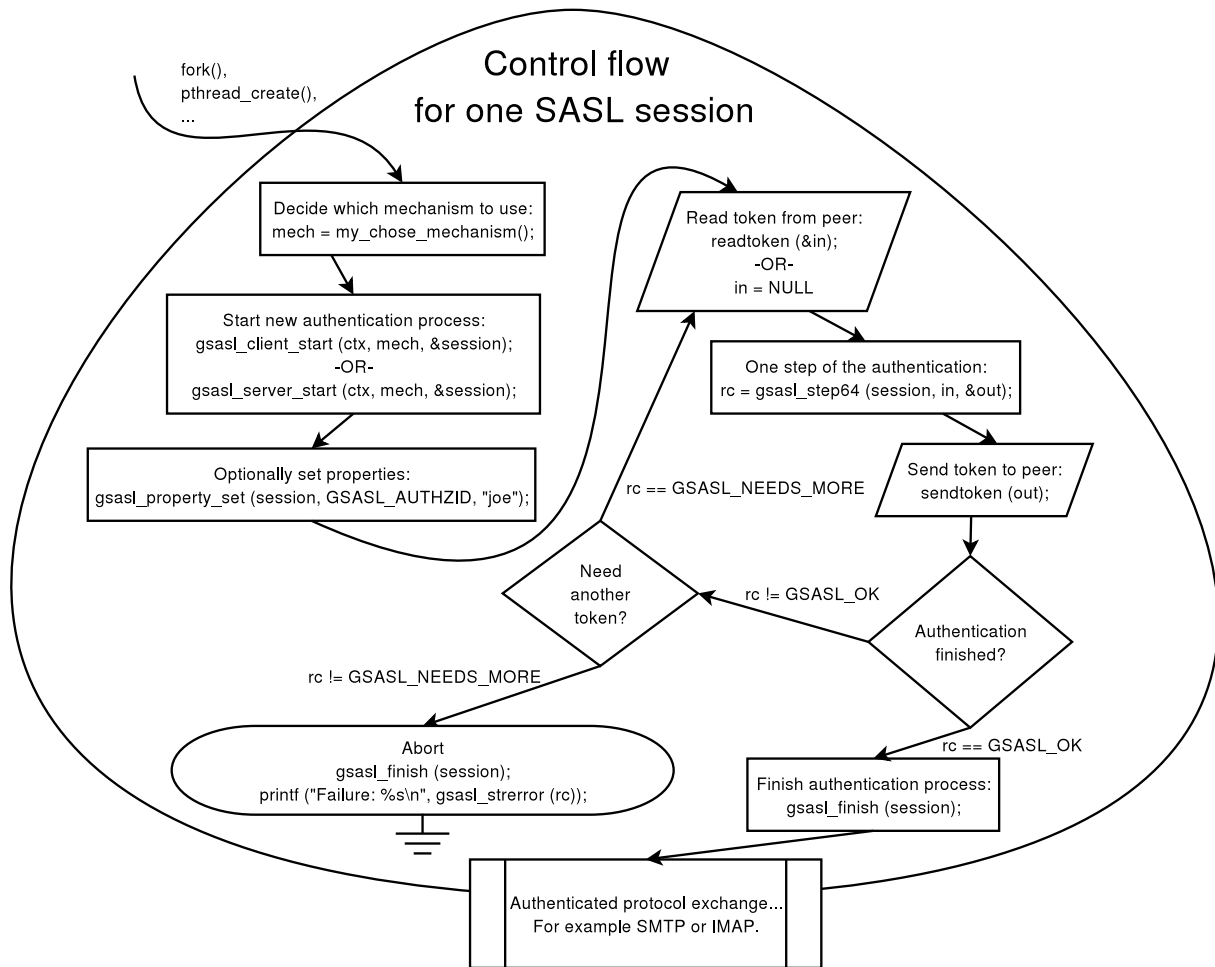


Figure 1.3 Control flow

1.4 Examples

```

/* client.c --- Example SASL client.
 * Copyright (C) 2004-2024 Simon Josefsson
 *
 * This file is part of GNU SASL.
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see <http://www.gnu.org/licenses/>.
 */
#include <config.h>
#include <stdarg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <gsasl.h>
static void
client_authenticate (Gsasl_session *session)
{
    char buf[BUFSIZ] = "";
    char *p;
    int rc;
    /* This loop mimics a protocol where the client send data first. */
    do
  
```

```

{
    /* Generate client output. */
    rc = gsasl_step64 (session, buf, &p);
    if (rc == GSASL_NEEDS_MORE || rc == GSASL_OK)
    {
        /* If successful, print it. */
        printf ("Output:\n%s\n", p);
        gsasl_free (p);
    }
    if (rc == GSASL_NEEDS_MORE)
    {
        /* If the client need more data from server, get it here. */
        printf ("Input base64 encoded data from server:\n");
        p = fgets (buf, sizeof (buf) - 1, stdin);
        if (p == NULL)
        {
            perror ("fgets");
            return;
        }
        if (buf[strlen (buf) - 1] == '\n')
            buf[strlen (buf) - 1] = '\0';
    }
}
while (rc == GSASL_NEEDS_MORE);
printf ("\n");
if (rc != GSASL_OK)
{
    printf ("Authentication error (%d): %s\n", rc, gsasl_strerror (rc));
    return;
}
/* The client is done. Here you would typically check if the server
   let the client in. If not, you could try again. */
printf ("If server accepted us, we're done.\n");
}
static void
client (Gsasl *ctx)
{
    Gsasl_session *session;
    const char *mech = "PLAIN";
    int rc;
    /* Create new authentication session. */
    if ((rc = gsasl_client_start (ctx, mech, &session)) != GSASL_OK)
    {
        printf ("Cannot initialize client (%d): %s\n", rc, gsasl_strerror (rc));
        return;
    }
    /* Set username and password in session handle. This info will be
       lost when this session is deallocated below. */
    rc = gsasl_property_set (session, GSASL_AUTHID, "jas");
    if (rc != GSASL_OK)
    {
        printf ("Cannot set property (%d): %s\n", rc, gsasl_strerror (rc));
        return;
    }
    rc = gsasl_property_set (session, GSASL_PASSWORD, "secret");
    if (rc != GSASL_OK)
    {
        printf ("Cannot set property (%d): %s\n", rc, gsasl_strerror (rc));
        return;
    }
    /* Do it. */
    client_authenticate (session);
    /* Cleanup. */
    gsasl_finish (session);
}
int
main (void)
{
    Gsasl *ctx = NULL;
    int rc;
    /* Initialize library. */
    if ((rc = gsasl_init (&ctx)) != GSASL_OK)
    {
        printf ("Cannot initialize libgsasl (%d): %s", rc, gsasl_strerror (rc));
        return 1;
    }
    /* Do it. */
    client (ctx);
    /* Cleanup. */
    gsasl_done (ctx);
    return 0;
}
/* client-serverfirst.c --- Example SASL client, where server send data first.
 * Copyright (C) 2004-2024 Simon Josefsson
 *
 * This file is part of GNU SASL.
 *

```

```

* This program is free software: you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation, either version 3 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program. If not, see <http://www.gnu.org/licenses/>.
*/
#include <config.h>
#include <stdarg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <gsasl.h>
static void
client_authenticate (Gsasl_session *session)
{
    char buf[BUFSIZ] = "";
    char *p;
    int rc;
    /* This loop mimics a protocol where the server send data first. */
    do
    {
        printf ("Input base64 encoded data from server:\n");
        p = fgets (buf, sizeof (buf) - 1, stdin);
        if (p == NULL)
        {
            perror ("fgets");
            return;
        }
        if (buf[strlen (buf) - 1] == '\n')
            buf[strlen (buf) - 1] = '\0';
        rc = gsasl_step64 (session, buf, &p);
        if (rc == GSASL_NEEDS_MORE || rc == GSASL_OK)
        {
            printf ("Output:\n%s\n", p);
            gsasl_free (p);
        }
    }
    while (rc == GSASL_NEEDS_MORE);
    printf ("\n");
    if (rc != GSASL_OK)
    {
        printf ("Authentication error (%d): %s\n", rc, gsasl_strerror (rc));
        return;
    }
    /* The client is done. Here you would typically check if the server
       let the client in. If not, you could try again. */
    printf ("If server accepted us, we're done.\n");
}
static void
client (Gsasl *ctx)
{
    Gsasl_session *session;
    const char *mech = "CRAM-MD5";
    int rc;
    /* Create new authentication session. */
    if ((rc = gsasl_client_start (ctx, mech, &session)) != GSASL_OK)
    {
        printf ("Cannot initialize client (%d): %s\n", rc, gsasl_strerror (rc));
        return;
    }
    /* Set username and password in session handle. This info will be
       lost when this session is deallocated below. */
    rc = gsasl_property_set (session, GSASL_AUTHID, "jas");
    if (rc != GSASL_OK)
    {
        printf ("Cannot set property (%d): %s\n", rc, gsasl_strerror (rc));
        return;
    }
    rc = gsasl_property_set (session, GSASL_PASSWORD, "secret");
    if (rc != GSASL_OK)
    {
        printf ("Cannot set property (%d): %s\n", rc, gsasl_strerror (rc));
        return;
    }
    /* Do it. */
    client_authenticate (session);
    /* Cleanup. */
    gsasl_finish (session);
}

```

```

int
main (void)
{
    Gsasl *ctx = NULL;
    int rc;
    /* Initialize library. */
    if ((rc = gsasl_init (&ctx)) != GSASL_OK)
    {
        printf ("Cannot initialize libgsasl (%d): %s", rc, gsasl_strerror (rc));
        return 1;
    }
    /* Do it. */
    client (ctx);
    /* Cleanup. */
    gsasl_done (ctx);
    return 0;
}
/* client-mech.c --- Example SASL client, with a choice of mechanism to use.
 * Copyright (C) 2004-2024 Simon Josefsson
 *
 * This file is part of GNU SASL.
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see <http://www.gnu.org/licenses/>.
 */
#include <config.h>
#include <stdarg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <gsasl.h>
static void
client_authenticate (Gsasl_session *session)
{
    char buf[BUFSIZ] = "";
    char *p;
    int rc;
    /* This loop mimics a protocol where the server send data first. */
    do
    {
        printf ("Input base64 encoded data from server:\n");
        p = fgets (buf, sizeof (buf) - 1, stdin);
        if (p == NULL)
        {
            perror ("fgets");
            return;
        }
        if (buf[strlen (buf) - 1] == '\n')
            buf[strlen (buf) - 1] = '\0';
        rc = gsasl_step64 (session, buf, &p);
        if (rc == GSASL_NEEDS_MORE || rc == GSASL_OK)
        {
            printf ("Output:\n%s\n", p);
            gsasl_free (p);
        }
    }
    while (rc == GSASL_NEEDS_MORE);
    printf ("\n");
    if (rc != GSASL_OK)
    {
        printf ("Authentication error (%d): %s\n", rc, gsasl_strerror (rc));
        return;
    }
    /* The client is done. Here you would typically check if the server
     * let the client in. If not, you could try again. */
    printf ("If server accepted us, we're done.\n");
}
static const char *
client_mechanism (Gsasl *ctx)
{
    static char mech[GSASL_MAX_MECHANISM_SIZE + 1] = "";
    char meclist[BUFSIZ] = "";
    const char *suggestion;
    char *p;
    printf ("Enter list of server supported mechanisms, separate by SPC:\n");
    p = fgets (mechlist, sizeof (mechlist) - 1, stdin);

```



```

if (p == NULL)
{
    perror ("fgets");
    return NULL;
}
suggestion = gssasl_client_suggest_mechanism (ctx, mechlist);
if (suggestion)
    printf ("Library suggests use of '%s'.\n", suggestion);
printf ("Enter mechanism to use:\n");
p = fgets (mech, sizeof (mech) - 1, stdin);
if (p == NULL)
{
    perror ("fgets");
    return NULL;
}
mech[strlen (mech) - 1] = '\0';
return mech;
}
static void
client (Gssasl *ctx)
{
    Gssasl_session *session;
    const char *mech;
    int rc;
    /* Find out which mechanism to use. */
    mech = client_mechanism (ctx);
    /* Create new authentication session. */
    if ((rc = gssasl_client_start (ctx, mech, &session)) != GSSASL_OK)
    {
        printf ("Cannot initialize client (%d): %s\n", rc, gssasl_strerror (rc));
        return;
    }
    /* Set username and password in session handle. This info will be
    lost when this session is deallocated below. */
    rc = gssasl_property_set (session, GSSASL_AUTHID, "jas");
    if (rc != GSSASL_OK)
    {
        printf ("Cannot set property (%d): %s\n", rc, gssasl_strerror (rc));
        return;
    }
    rc = gssasl_property_set (session, GSSASL_PASSWORD, "secret");
    if (rc != GSSASL_OK)
    {
        printf ("Cannot set property (%d): %s\n", rc, gssasl_strerror (rc));
        return;
    }
    /* Do it. */
    client_authenticate (session);
    /* Cleanup. */
    gssasl_finish (session);
}
int
main (void)
{
    Gssasl *ctx = NULL;
    int rc;
    /* Initialize library. */
    if ((rc = gssasl_init (&ctx)) != GSSASL_OK)
    {
        printf ("Cannot initialize libgssasl (%d): %s", rc, gssasl_strerror (rc));
        return 1;
    }
    /* Do it. */
    client (ctx);
    /* Cleanup. */
    gssasl_done (ctx);
    return 0;
}
/* client-callback.c --- Example SASL client, with callback for user info.
* Copyright (C) 2004-2024 Simon Josefsson
*
* This file is part of GNU SASL.
*
* This program is free software: you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation, either version 3 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program. If not, see <http://www.gnu.org/licenses/>.
*
*/

```

```

#include <config.h>
#include <stdarg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <gsasl.h>
static void
client_authenticate (Gsasl_session *session)
{
    char buf[BUFSIZ] = "";
    char *p;
    int rc;
    /* This loop mimics a protocol where the server send data first. */
    do
    {
        printf ("Input base64 encoded data from server:\n");
        p = fgets (buf, sizeof (buf) - 1, stdin);
        if (p == NULL)
        {
            perror ("fgets");
            return;
        }
        if (buf[strlen (buf) - 1] == '\n')
            buf[strlen (buf) - 1] = '\0';
        rc = gsasl_step64 (session, buf, &p);
        if (rc == GSASL_NEEDS_MORE || rc == GSASL_OK)
        {
            printf ("Output:\n%s\n", p);
            gsasl_free (p);
        }
    }
    while (rc == GSASL_NEEDS_MORE);
    printf ("\n");
    if (rc != GSASL_OK)
    {
        printf ("Authentication error (%d): %s\n", rc, gsasl_strerror (rc));
        return;
    }
    /* The client is done. Here you would typically check if the server
       let the client in. If not, you could try again. */
    printf ("If server accepted us, we're done.\n");
}
static void
client (Gsasl *ctx)
{
    Gsasl_session *session;
    const char *mech = "SECURID";
    int rc;
    /* Create new authentication session. */
    if ((rc = gsasl_client_start (ctx, mech, &session)) != GSASL_OK)
    {
        printf ("Cannot initialize client (%d): %s\n", rc, gsasl_strerror (rc));
        return;
    }
    /* Do it. */
    client_authenticate (session);
    /* Cleanup. */
    gsasl_finish (session);
}
static int
callback (Gsasl *ctx, Gsasl_session *sctx, Gsasl_property prop)
{
    char buf[BUFSIZ] = "";
    int rc = GSASL_NO_CALLBACK;
    char *p;
    (void) ctx;
    /* Get user info from user. */
    printf ("Callback invoked, for property %u.\n", prop);
    switch (prop)
    {
        {
            case GSASL_PASSCODE:
                printf ("Enter passcode:\n");
                p = fgets (buf, sizeof (buf) - 1, stdin);
                if (p == NULL)
                {
                    perror ("fgets");
                    break;
                }
                buf[strlen (buf) - 1] = '\0';
                rc = gsasl_property_set (sctx, GSASL_PASSCODE, buf);
                break;
            case GSASL_AUTHID:
                printf ("Enter username:\n");
                p = fgets (buf, sizeof (buf) - 1, stdin);
                if (p == NULL)
                {
                    perror ("fgets");
                }
        }
    }
}

```

```
        break;
    }
    buf[strlen (buf) - 1] = '\\0';
    rc = gssasl_property_set (sctx, GSASL_AUTHID, buf);
    break;
default:
    printf ("Unknown property! Don't worry.\\n");
    break;
}
return rc;
}
int
main (void)
{
    Gssasl *ctx = NULL;
    int rc;
    /* Initialize library. */
    if ((rc = gssasl_init (&ctx)) != GSASL_OK)
    {
        printf ("Cannot initialize libgssasl (%d): %s", rc, gssasl_strerror (rc));
        return 1;
    }
    /* Set the callback handler for the library. */
    gssasl_callback_set (ctx, callback);
    /* Do it. */
    client (ctx);
    /* Cleanup. */
    gssasl_done (ctx);
    return 0;
}
```


Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

_Gsasl_digest_md5_client_state	17
_Gsasl_digest_md5_server_state	19
_gsasl_gs2_client_state	22
_Gsasl_gs2_server_state	23
_Gsasl_gssapi_client_state	25
_Gsasl_gssapi_server_state	26
_Gsasl_login_client_state	27
_Gsasl_login_server_state	27
_Gsasl_ntlm_state	28
digest_md5_challenge	29
digest_md5_finish	30
digest_md5_response	31
Gsasl	34
Gsasl_mechanism	35
Gsasl_mechanism_functions	36
Gsasl_session	38
openid20_client_state	44
openid20_server_state	45
saml20_client_state	45
saml20_server_state	46
scram_client_final	46
scram_client_first	47
scram_client_state	49
scram_server_final	51
scram_server_first	52
scram_server_state	53

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

anonymous.h	57
base64.c	58
callback.c	61
challenge.c	64
challenge.h	65
anonymous/client.c	66
cram-md5/client.c	67
digest-md5/client.c	68
external/client.c	70
gs2/client.c	71
gssapi/client.c	72
login/client.c	74
openid20/client.c	75
plain/client.c	77
saml20/client.c	77
scram/client.c	78
securid/client.c	80
cram-md5.h	81
crypto.c	83
digest-md5.h	87
digest.c	90
digest.h	91
digesthmac.c	92
digesthmac.h	95
done.c	96
doxygen.c	97
error.c	97
external.h	100
digest-md5/free.c	101
src/free.c	102
free.h	103
getsubopt.c	104
gs2.h	104
gs2helper.c	106
gs2helper.h	107

gsasl-mech.h	107
gsasl-version.h	112
gsasl.h	114
init.c	144
internal.h	145
listmech.c	146
login.h	147
md5pwd.c	149
anonymous/mechinfo.c	150
cram-md5/mechinfo.c	151
digest-md5/mechinfo.c	151
external/mechinfo.c	152
gs2/mechinfo.c	152
gssapi/mechinfo.c	153
login/mechinfo.c	153
ntlm/mechinfo.c	154
openid20/mechinfo.c	154
plain/mechinfo.c	155
saml20/mechinfo.c	155
scram/mechinfo.c	156
securid/mechinfo.c	156
mechname.c	157
mechtools.c	157
mechtools.h	160
nonascii.c	162
nonascii.h	163
ntlm.c	163
openid20.h	165
digest-md5/parser.c	167
scram/parser.c	171
digest-md5/parser.h	172
scram/parser.h	173
plain.h	174
digest-md5/printer.c	176
scram/printer.c	177
digest-md5/printer.h	178
scram/printer.h	179
property.c	180
qop.c	183
qop.h	183
register.c	184
saml20.h	185
saslprep.c	187
scram.h	188
securid.h	188
anonymous/server.c	190
cram-md5/server.c	190
digest-md5/server.c	192
external/server.c	194
gs2/server.c	195
gssapi/server.c	196
login/server.c	198
openid20/server.c	199
plain/server.c	201
saml20/server.c	201
scram/server.c	202
securid/server.c	204
session.c	205

session.h	208
suggest.c	208
supportp.c	210
test-parser.c	211
tokens.c	212
digest-md5/tokens.h	213
scram/tokens.h	216
tools.c	217
tools.h	217
digest-md5/validate.c	218
scram/validate.c	219
digest-md5/validate.h	220
scram/validate.h	221
version.c	222
x-gssapi.h	223
x-ntlm.h	226
xcode.c	227
xfinish.c	229
xstart.c	230
xstep.c	231

Chapter 4

Data Structure Documentation

4.1 `_Gssl_digest_md5_client_state` Struct Reference

Data Fields

- int `step`
- unsigned long `readseqnum`
- unsigned long `sendseqnum`
- char `secret` [DIGEST_MD5_LENGTH]
- char `kic` [DIGEST_MD5_LENGTH]
- char `kcc` [DIGEST_MD5_LENGTH]
- char `kis` [DIGEST_MD5_LENGTH]
- char `kcs` [DIGEST_MD5_LENGTH]
- `digest_md5_challenge` challenge
- `digest_md5_response` response
- `digest_md5_finish` finish

4.1.1 Detailed Description

Definition at line 50 of file `digest-md5/client.c`.

4.1.2 Field Documentation

4.1.2.1 challenge

`digest_md5_challenge` `_Gssl_digest_md5_client_state::challenge`

Definition at line 59 of file `digest-md5/client.c`.

4.1.2.2 finish

```
digest_md5_finish _Gssasl_digest_md5_client_state::finish
```

Definition at line 61 of file digest-md5/client.c.

4.1.2.3 kcc

```
char _Gssasl_digest_md5_client_state::kcc[DIGEST_MD5_LENGTH]
```

Definition at line 56 of file digest-md5/client.c.

4.1.2.4 kcs

```
char _Gssasl_digest_md5_client_state::kcs[DIGEST_MD5_LENGTH]
```

Definition at line 58 of file digest-md5/client.c.

4.1.2.5 kic

```
char _Gssasl_digest_md5_client_state::kic[DIGEST_MD5_LENGTH]
```

Definition at line 55 of file digest-md5/client.c.

4.1.2.6 kis

```
char _Gssasl_digest_md5_client_state::kis[DIGEST_MD5_LENGTH]
```

Definition at line 57 of file digest-md5/client.c.

4.1.2.7 readseqnum

```
unsigned long _Gssasl_digest_md5_client_state::readseqnum
```

Definition at line 53 of file digest-md5/client.c.

4.1.2.8 response

`digest_md5_response` `_Gsassl_digest_md5_client_state::response`

Definition at line 60 of file `digest-md5/client.c`.

4.1.2.9 secret

`char` `_Gsassl_digest_md5_client_state::secret` [`DIGEST_MD5_LENGTH`]

Definition at line 54 of file `digest-md5/client.c`.

4.1.2.10 sendseqnum

`unsigned long` `_Gsassl_digest_md5_client_state::sendseqnum`

Definition at line 53 of file `digest-md5/client.c`.

4.1.2.11 step

`int` `_Gsassl_digest_md5_client_state::step`

Definition at line 52 of file `digest-md5/client.c`.

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

- [digest-md5/client.c](#)

4.2 _Gsassl_digest_md5_server_state Struct Reference

Data Fields

- `int` `step`
- `unsigned long` `readseqnum`
- `unsigned long` `sendseqnum`
- `char` `secret` [`DIGEST_MD5_LENGTH`]
- `char` `kic` [`DIGEST_MD5_LENGTH`]
- `char` `kcc` [`DIGEST_MD5_LENGTH`]
- `char` `kis` [`DIGEST_MD5_LENGTH`]
- `char` `kcs` [`DIGEST_MD5_LENGTH`]
- `digest_md5_challenge` `challenge`
- `digest_md5_response` `response`
- `digest_md5_finish` `finish`

4.2.1 Detailed Description

Definition at line 51 of file digest-md5/server.c.

4.2.2 Field Documentation

4.2.2.1 challenge

`digest_md5_challenge` `_Gsas1_digest_md5_server_state::challenge`

Definition at line 60 of file digest-md5/server.c.

4.2.2.2 finish

`digest_md5_finish` `_Gsas1_digest_md5_server_state::finish`

Definition at line 62 of file digest-md5/server.c.

4.2.2.3 kcc

`char` `_Gsas1_digest_md5_server_state::kcc` `[DIGEST_MD5_LENGTH]`

Definition at line 57 of file digest-md5/server.c.

4.2.2.4 kcs

`char` `_Gsas1_digest_md5_server_state::kcs` `[DIGEST_MD5_LENGTH]`

Definition at line 59 of file digest-md5/server.c.

4.2.2.5 kic

`char` `_Gsas1_digest_md5_server_state::kic` `[DIGEST_MD5_LENGTH]`

Definition at line 56 of file digest-md5/server.c.

4.2.2.6 `kis`

```
char _Gssasl_digest_md5_server_state::kis [DIGEST_MD5_LENGTH]
```

Definition at line 58 of file `digest-md5/server.c`.

4.2.2.7 `readseqnum`

```
unsigned long _Gssasl_digest_md5_server_state::readseqnum
```

Definition at line 54 of file `digest-md5/server.c`.

4.2.2.8 `response`

```
digest\_md5\_response _Gssasl_digest_md5_server_state::response
```

Definition at line 61 of file `digest-md5/server.c`.

4.2.2.9 `secret`

```
char _Gssasl_digest_md5_server_state::secret [DIGEST_MD5_LENGTH]
```

Definition at line 55 of file `digest-md5/server.c`.

4.2.2.10 `sendseqnum`

```
unsigned long _Gssasl_digest_md5_server_state::sendseqnum
```

Definition at line 54 of file `digest-md5/server.c`.

4.2.2.11 `step`

```
int _Gssasl_digest_md5_server_state::step
```

Definition at line 53 of file `digest-md5/server.c`.

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

- [digest-md5/server.c](#)

4.3 `_gsasl_gs2_client_state` Struct Reference

Data Fields

- int [step](#)
- gss_name_t [service](#)
- gss_ctx_id_t [context](#)
- gss_OID [mech_oid](#)
- gss_buffer_desc [token](#)
- struct gss_channel_bindings_struct [cb](#)

4.3.1 Detailed Description

Definition at line 37 of file `gs2/client.c`.

4.3.2 Field Documentation

4.3.2.1 `cb`

```
struct gss_channel_bindings_struct _gsasl_gs2_client_state::cb
```

Definition at line 44 of file `gs2/client.c`.

4.3.2.2 `context`

```
gss_ctx_id_t _gsasl_gs2_client_state::context
```

Definition at line 42 of file `gs2/client.c`.

4.3.2.3 `mech_oid`

```
gss_OID _gsasl_gs2_client_state::mech_oid
```

Definition at line 43 of file `gs2/client.c`.

4.3.2.4 service

```
gss_name_t _gssasl_gs2_client_state::service
```

Definition at line 41 of file gs2/client.c.

4.3.2.5 step

```
int _gssasl_gs2_client_state::step
```

Definition at line 40 of file gs2/client.c.

4.3.2.6 token

```
gss_buffer_desc _gssasl_gs2_client_state::token
```

Definition at line 44 of file gs2/client.c.

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

- [gs2/client.c](#)

4.4 _Gssasl_gs2_server_state Struct Reference

Data Fields

- int [step](#)
- gss_name_t [client](#)
- gss_cred_id_t [cred](#)
- gss_ctx_id_t [context](#)
- gss_OID [mech_oid](#)
- struct gss_channel_bindings_struct [cb](#)

4.4.1 Detailed Description

Definition at line 41 of file gs2/server.c.

4.4.2 Field Documentation

4.4.2.1 cb

```
struct gss_channel_bindings_struct _Gssasl_gs2_server_state::cb
```

Definition at line 48 of file gs2/server.c.

4.4.2.2 client

```
gss_name_t _Gssasl_gs2_server_state::client
```

Definition at line 45 of file gs2/server.c.

4.4.2.3 context

```
gss_ctx_id_t _Gssasl_gs2_server_state::context
```

Definition at line 47 of file gs2/server.c.

4.4.2.4 cred

```
gss_cred_id_t _Gssasl_gs2_server_state::cred
```

Definition at line 46 of file gs2/server.c.

4.4.2.5 mech_oid

```
gss_OID _Gssasl_gs2_server_state::mech_oid
```

Definition at line 48 of file gs2/server.c.

4.4.2.6 step

```
int _Gssasl_gs2_server_state::step
```

Definition at line 44 of file gs2/server.c.

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

- [gs2/server.c](#)

4.5 `_Gsasl_gssapi_client_state` Struct Reference

Data Fields

- int [step](#)
- gss_name_t [service](#)
- gss_ctx_id_t [context](#)
- gss_qop_t [qop](#)

4.5.1 Detailed Description

Definition at line 37 of file `gssapi/client.c`.

4.5.2 Field Documentation

4.5.2.1 context

```
gss_ctx_id_t _Gsasl_gssapi_client_state::context
```

Definition at line 41 of file `gssapi/client.c`.

4.5.2.2 qop

```
gss_qop_t _Gsasl_gssapi_client_state::qop
```

Definition at line 42 of file `gssapi/client.c`.

4.5.2.3 service

```
gss_name_t _Gsasl_gssapi_client_state::service
```

Definition at line 40 of file `gssapi/client.c`.

4.5.2.4 step

```
int _Gsasl_gssapi_client_state::step
```

Definition at line 39 of file `gssapi/client.c`.

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

- [gssapi/client.c](#)

4.6 `_Gssasl_gssapi_server_state` Struct Reference

Data Fields

- int [step](#)
- gss_name_t [client](#)
- gss_cred_id_t [cred](#)
- gss_ctx_id_t [context](#)

4.6.1 Detailed Description

Definition at line 37 of file `gssapi/server.c`.

4.6.2 Field Documentation

4.6.2.1 `client`

```
gss_name_t _Gssasl_gssapi_server_state::client
```

Definition at line 40 of file `gssapi/server.c`.

4.6.2.2 `context`

```
gss_ctx_id_t _Gssasl_gssapi_server_state::context
```

Definition at line 42 of file `gssapi/server.c`.

4.6.2.3 `cred`

```
gss_cred_id_t _Gssasl_gssapi_server_state::cred
```

Definition at line 41 of file `gssapi/server.c`.

4.6.2.4 `step`

```
int _Gssasl_gssapi_server_state::step
```

Definition at line 39 of file `gssapi/server.c`.

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

- [gssapi/server.c](#)

4.7 `_Gssasl_login_client_state` Struct Reference

Data Fields

- int [step](#)

4.7.1 Detailed Description

Definition at line 34 of file login/client.c.

4.7.2 Field Documentation

4.7.2.1 `step`

```
int _Gssasl_login_client_state::step
```

Definition at line 36 of file login/client.c.

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

- [login/client.c](#)

4.8 `_Gssasl_login_server_state` Struct Reference

Data Fields

- int [step](#)
- char * [username](#)
- char * [password](#)

4.8.1 Detailed Description

Definition at line 34 of file login/server.c.

4.8.2 Field Documentation

4.8.2.1 password

```
char* _Gssasl_login_server_state::password
```

Definition at line 38 of file login/server.c.

4.8.2.2 step

```
int _Gssasl_login_server_state::step
```

Definition at line 36 of file login/server.c.

4.8.2.3 username

```
char* _Gssasl_login_server_state::username
```

Definition at line 37 of file login/server.c.

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

- [login/server.c](#)

4.9 _Gssasl_ntlm_state Struct Reference

Data Fields

- int [step](#)

4.9.1 Detailed Description

Definition at line 36 of file ntlm.c.

4.9.2 Field Documentation

4.9.2.1 step

```
int _Gssasl_ntlm_state::step
```

Definition at line 38 of file ntlm.c.

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

- [ntlm.c](#)

4.10 digest_md5_challenge Struct Reference

```
#include <tokens.h>
```

Data Fields

- `size_t` [nrealms](#)
- `char **` [realms](#)
- `char *` [nonce](#)
- `int` [qops](#)
- `int` [stale](#)
- `unsigned long` [servermaxbuf](#)
- `int` [utf8](#)
- `int` [ciphers](#)

4.10.1 Detailed Description

Definition at line 82 of file digest-md5/tokens.h.

4.10.2 Field Documentation

4.10.2.1 ciphers

```
int digest_md5_challenge::ciphers
```

Definition at line 91 of file digest-md5/tokens.h.

4.10.2.2 nonce

```
char* digest_md5_challenge::nonce
```

Definition at line 86 of file digest-md5/tokens.h.

4.10.2.3 nrealms

```
size_t digest_md5_challenge::nrealms
```

Definition at line 84 of file digest-md5/tokens.h.

4.10.2.4 qops

```
int digest_md5_challenge::qops
```

Definition at line 87 of file digest-md5/tokens.h.

4.10.2.5 realms

```
char** digest_md5_challenge::realms
```

Definition at line 85 of file digest-md5/tokens.h.

4.10.2.6 servermaxbuf

```
unsigned long digest_md5_challenge::servermaxbuf
```

Definition at line 89 of file digest-md5/tokens.h.

4.10.2.7 stale

```
int digest_md5_challenge::stale
```

Definition at line 88 of file digest-md5/tokens.h.

4.10.2.8 utf8

```
int digest_md5_challenge::utf8
```

Definition at line 90 of file digest-md5/tokens.h.

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

- [digest-md5/tokens.h](#)

4.11 digest_md5_finish Struct Reference

```
#include <tokens.h>
```


Data Fields

- char [rspauth](#) [DIGEST_MD5_RESPONSE_LENGTH+1]

4.11.1 Detailed Description

Definition at line 146 of file digest-md5/tokens.h.

4.11.2 Field Documentation

4.11.2.1 rspauth

```
char digest_md5_finish::rspauth[DIGEST_MD5_RESPONSE_LENGTH+1]
```

Definition at line 148 of file digest-md5/tokens.h.

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

- [digest-md5/tokens.h](#)

4.12 digest_md5_response Struct Reference

```
#include <tokens.h>
```

Data Fields

- char * [username](#)
- char * [realm](#)
- char * [nonce](#)
- char * [cnonce](#)
- unsigned long [nc](#)
- [digest_md5_qop](#) qop
- char * [digesturi](#)
- unsigned long [clientmaxbuf](#)
- int [utf8](#)
- [digest_md5_cipher](#) cipher
- char * [authzid](#)
- char [response](#) [DIGEST_MD5_RESPONSE_LENGTH+1]

4.12.1 Detailed Description

Definition at line 126 of file digest-md5/tokens.h.

4.12.2 Field Documentation

4.12.2.1 authzid

```
char* digest_md5_response::authzid
```

Definition at line 138 of file digest-md5/tokens.h.

4.12.2.2 cipher

```
digest_md5_cipher digest_md5_response::cipher
```

Definition at line 137 of file digest-md5/tokens.h.

4.12.2.3 clientmaxbuf

```
unsigned long digest_md5_response::clientmaxbuf
```

Definition at line 135 of file digest-md5/tokens.h.

4.12.2.4 cnonce

```
char* digest_md5_response::cnonce
```

Definition at line 131 of file digest-md5/tokens.h.

4.12.2.5 digesturi

```
char* digest_md5_response::digesturi
```

Definition at line 134 of file digest-md5/tokens.h.

4.12.2.6 nc

```
unsigned long digest_md5_response::nc
```

Definition at line 132 of file digest-md5/tokens.h.

4.12.2.7 nonce

```
char* digest_md5_response::nonce
```

Definition at line 130 of file digest-md5/tokens.h.

4.12.2.8 qop

```
digest\_md5\_qop digest_md5_response::qop
```

Definition at line 133 of file digest-md5/tokens.h.

4.12.2.9 realm

```
char* digest_md5_response::realm
```

Definition at line 129 of file digest-md5/tokens.h.

4.12.2.10 response

```
char digest_md5_response::response[DIGEST\_MD5\_RESPONSE\_LENGTH+1]
```

Definition at line 139 of file digest-md5/tokens.h.

4.12.2.11 username

```
char* digest_md5_response::username
```

Definition at line 128 of file digest-md5/tokens.h.

4.12.2.12 utf8

```
int digest_md5_response::utf8
```

Definition at line 136 of file digest-md5/tokens.h.

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

- [digest-md5/tokens.h](#)

4.13 Gsasl Struct Reference

```
#include <internal.h>
```

Data Fields

- [size_t n_client_mechs](#)
- [Gsasl_mechanism * client_mechs](#)
- [size_t n_server_mechs](#)
- [Gsasl_mechanism * server_mechs](#)
- [Gsasl_callback_function cb](#)
- [void * application_hook](#)

4.13.1 Detailed Description

Definition at line 36 of file internal.h.

4.13.2 Field Documentation

4.13.2.1 application_hook

```
void* Gsasl::application_hook
```

Definition at line 44 of file internal.h.

4.13.2.2 cb

```
Gsasl_callback_function Gsasl::cb
```

Definition at line 43 of file internal.h.

4.13.2.3 client_mechs

`Gsasl_mechanism*` `Gsasl::client_mechs`

Definition at line 39 of file `internal.h`.

4.13.2.4 n_client_mechs

`size_t` `Gsasl::n_client_mechs`

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

4.13.2.5 n_server_mechs

`size_t` `Gsasl::n_server_mechs`

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

4.13.2.6 server_mechs

`Gsasl_mechanism*` `Gsasl::server_mechs`

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

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

- [internal.h](#)

4.14 Gsasl_mechanism Struct Reference

```
#include <gsasl-mech.h>
```

Data Fields

- `const char *` `name`
- `struct Gsasl_mechanism_functions` `client`
- `struct Gsasl_mechanism_functions` `server`

4.14.1 Detailed Description

`Gsasl_mechanism`:

Parameters

<i>name</i>	string holding name of mechanism, e.g., "PLAIN".
<i>client</i>	client-side Gsasl_mechanism_functions structure.
<i>server</i>	server-side Gsasl_mechanism_functions structure.

Holds all implementation details about a mechanism.

Definition at line 171 of file gsasl-mech.h.

4.14.2 Field Documentation**4.14.2.1 client**

```
struct Gsasl\_mechanism\_functions Gsasl_mechanism::client
```

Definition at line 173 of file gsasl-mech.h.

4.14.2.2 name

```
const char* Gsasl_mechanism::name
```

Definition at line 173 of file gsasl-mech.h.

4.14.2.3 server

```
struct Gsasl\_mechanism\_functions Gsasl_mechanism::server
```

Definition at line 173 of file gsasl-mech.h.

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

- [gsasl-mech.h](#)

4.15 Gsasl_mechanism_functions Struct Reference

```
#include <gsasl-mech.h>
```

Data Fields

- [Gsasl_init_function](#) init
- [Gsasl_done_function](#) done
- [Gsasl_start_function](#) start
- [Gsasl_step_function](#) step
- [Gsasl_finish_function](#) finish
- [Gsasl_code_function](#) encode
- [Gsasl_code_function](#) decode

4.15.1 Detailed Description

[Gsasl_mechanism_functions](#):

Parameters

<i>init</i>	a Gsasl_init_function() .
<i>done</i>	a Gsasl_done_function() .
<i>start</i>	a Gsasl_start_function() .
<i>step</i>	a Gsasl_step_function() .
<i>finish</i>	a Gsasl_finish_function() .
<i>encode</i>	a Gsasl_code_function() .
<i>decode</i>	a Gsasl_code_function() .

Holds all function pointers to implement a mechanism, in either client or server mode.

Definition at line 151 of file gsasl-mech.h.

4.15.2 Field Documentation

4.15.2.1 decode

[Gsasl_code_function](#) `Gsasl_mechanism_functions::decode`

Definition at line 159 of file gsasl-mech.h.

4.15.2.2 done

[Gsasl_done_function](#) `Gsasl_mechanism_functions::done`

Definition at line 154 of file gsasl-mech.h.

4.15.2.3 encode

[Gsasl_code_function](#) `Gsasl_mechanism_functions::encode`

Definition at line 158 of file gsasl-mech.h.

4.15.2.4 finish

[Gsasl_finish_function](#) `Gsasl_mechanism_functions::finish`

Definition at line 157 of file gsasl-mech.h.

4.15.2.5 init

[Gsasl_init_function](#) Gsasl_mechanism_functions::init

Definition at line 153 of file gsasl-mech.h.

4.15.2.6 start

[Gsasl_start_function](#) Gsasl_mechanism_functions::start

Definition at line 155 of file gsasl-mech.h.

4.15.2.7 step

[Gsasl_step_function](#) Gsasl_mechanism_functions::step

Definition at line 156 of file gsasl-mech.h.

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

- [gsasl-mech.h](#)

4.16 Gsasl_session Struct Reference

```
#include <internal.h>
```

Data Fields

- [Gsasl](#) * ctx
- int clientp
- [Gsasl_mechanism](#) * mech
- void * mech_data
- void * application_hook
- char * anonymous_token
- char * authid
- char * authzid
- char * password
- char * passcode
- char * pin
- char * suggestedpin
- char * service
- char * hostname
- char * gssapi_display_name
- char * realm
- char * digest_md5_hashed_password

- char * [qops](#)
- char * [qop](#)
- char * [scram_iter](#)
- char * [scram_salt](#)
- char * [scram_salted_password](#)
- char * [scram_serverkey](#)
- char * [scram_storedkey](#)
- char * [cb_tls_unique](#)
- char * [cb_tls_exporter](#)
- char * [saml20_idp_identifier](#)
- char * [saml20_redirect_url](#)
- char * [openid20_redirect_url](#)
- char * [openid20_outcome_data](#)

4.16.1 Detailed Description

Definition at line 48 of file internal.h.

4.16.2 Field Documentation

4.16.2.1 anonymous_token

```
char* Gsasl_session::anonymous_token
```

Definition at line 57 of file internal.h.

4.16.2.2 application_hook

```
void* Gsasl_session::application_hook
```

Definition at line 54 of file internal.h.

4.16.2.3 authid

```
char* Gsasl_session::authid
```

Definition at line 58 of file internal.h.

4.16.2.4 authzid

```
char* Gsasl_session::authzid
```

Definition at line 59 of file internal.h.

4.16.2.5 cb_tls_exporter

```
char* Gsasl_session::cb_tls_exporter
```

Definition at line 77 of file internal.h.

4.16.2.6 cb_tls_unique

```
char* Gsasl_session::cb_tls_unique
```

Definition at line 76 of file internal.h.

4.16.2.7 clientp

```
int Gsasl_session::clientp
```

Definition at line 51 of file internal.h.

4.16.2.8 ctx

```
Gsasl* Gsasl_session::ctx
```

Definition at line 50 of file internal.h.

4.16.2.9 digest_md5_hashed_password

```
char* Gsasl_session::digest_md5_hashed_password
```

Definition at line 68 of file internal.h.

4.16.2.10 gssapi_display_name

char* Gsasl_session::gssapi_display_name

Definition at line 66 of file internal.h.

4.16.2.11 hostname

char* Gsasl_session::hostname

Definition at line 65 of file internal.h.

4.16.2.12 mech

[Gsasl_mechanism*](#) Gsasl_session::mech

Definition at line 52 of file internal.h.

4.16.2.13 mech_data

void* Gsasl_session::mech_data

Definition at line 53 of file internal.h.

4.16.2.14 openid20_outcome_data

char* Gsasl_session::openid20_outcome_data

Definition at line 81 of file internal.h.

4.16.2.15 openid20_redirect_url

char* Gsasl_session::openid20_redirect_url

Definition at line 80 of file internal.h.

4.16.2.16 passcode

```
char* Gsasl_session::passcode
```

Definition at line 61 of file internal.h.

4.16.2.17 password

```
char* Gsasl_session::password
```

Definition at line 60 of file internal.h.

4.16.2.18 pin

```
char* Gsasl_session::pin
```

Definition at line 62 of file internal.h.

4.16.2.19 qop

```
char* Gsasl_session::qop
```

Definition at line 70 of file internal.h.

4.16.2.20 qops

```
char* Gsasl_session::qops
```

Definition at line 69 of file internal.h.

4.16.2.21 realm

```
char* Gsasl_session::realm
```

Definition at line 67 of file internal.h.

4.16.2.22 saml20_idp_identifier

```
char* Gsasl_session::saml20_idp_identifier
```

Definition at line 78 of file internal.h.

4.16.2.23 saml20_redirect_url

```
char* Gsasl_session::saml20_redirect_url
```

Definition at line 79 of file internal.h.

4.16.2.24 scram_iter

```
char* Gsasl_session::scram_iter
```

Definition at line 71 of file internal.h.

4.16.2.25 scram_salt

```
char* Gsasl_session::scram_salt
```

Definition at line 72 of file internal.h.

4.16.2.26 scram_saltd_password

```
char* Gsasl_session::scram_saltd_password
```

Definition at line 73 of file internal.h.

4.16.2.27 scram_serverkey

```
char* Gsasl_session::scram_serverkey
```

Definition at line 74 of file internal.h.

4.16.2.28 `scram_storedkey`

```
char* Gsasl_session::scram_storedkey
```

Definition at line 75 of file `internal.h`.

4.16.2.29 `service`

```
char* Gsasl_session::service
```

Definition at line 64 of file `internal.h`.

4.16.2.30 `suggestedpin`

```
char* Gsasl_session::suggestedpin
```

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

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

- [internal.h](#)

4.17 `openid20_client_state` Struct Reference

Data Fields

- int [step](#)

4.17.1 Detailed Description

Definition at line 40 of file `openid20/client.c`.

4.17.2 Field Documentation

4.17.2.1 `step`

```
int openid20_client_state::step
```

Definition at line 42 of file `openid20/client.c`.

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

- [openid20/client.c](#)

4.18 openid20_server_state Struct Reference

Data Fields

- int [step](#)
- int [allow_error_step](#)

4.18.1 Detailed Description

Definition at line 37 of file openid20/server.c.

4.18.2 Field Documentation

4.18.2.1 allow_error_step

```
int openid20_server_state::allow_error_step
```

Definition at line 40 of file openid20/server.c.

4.18.2.2 step

```
int openid20_server_state::step
```

Definition at line 39 of file openid20/server.c.

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

- [openid20/server.c](#)

4.19 saml20_client_state Struct Reference

Data Fields

- int [step](#)

4.19.1 Detailed Description

Definition at line 40 of file saml20/client.c.

4.19.2 Field Documentation

4.19.2.1 step

```
int saml20_client_state::step
```

Definition at line 42 of file saml20/client.c.

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

- [saml20/client.c](#)

4.20 saml20_server_state Struct Reference

Data Fields

- int [step](#)

4.20.1 Detailed Description

Definition at line 37 of file saml20/server.c.

4.20.2 Field Documentation

4.20.2.1 step

```
int saml20_server_state::step
```

Definition at line 39 of file saml20/server.c.

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

- [saml20/server.c](#)

4.21 scram_client_final Struct Reference

```
#include <tokens.h>
```


Data Fields

- char * [cbind](#)
- char * [nonce](#)
- char * [proof](#)

4.21.1 Detailed Description

Definition at line 45 of file `scram/tokens.h`.

4.21.2 Field Documentation

4.21.2.1 `cbind`

```
char* scram_client_final::cbind
```

Definition at line 47 of file `scram/tokens.h`.

4.21.2.2 `nonce`

```
char* scram_client_final::nonce
```

Definition at line 48 of file `scram/tokens.h`.

4.21.2.3 `proof`

```
char* scram_client_final::proof
```

Definition at line 49 of file `scram/tokens.h`.

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

- [scram/tokens.h](#)

4.22 `scram_client_first` Struct Reference

```
#include <tokens.h>
```

Data Fields

- char [cbflag](#)
- char * [cbname](#)
- char * [authzid](#)
- char * [username](#)
- char * [client_nonce](#)

4.22.1 Detailed Description

Definition at line 29 of file `scram/tokens.h`.

4.22.2 Field Documentation

4.22.2.1 `authzid`

```
char* scram_client_first::authzid
```

Definition at line 33 of file `scram/tokens.h`.

4.22.2.2 `cbflag`

```
char scram_client_first::cbflag
```

Definition at line 31 of file `scram/tokens.h`.

4.22.2.3 `cbname`

```
char* scram_client_first::cbname
```

Definition at line 32 of file `scram/tokens.h`.

4.22.2.4 `client_nonce`

```
char* scram_client_first::client_nonce
```

Definition at line 35 of file `scram/tokens.h`.

4.22.2.5 `username`

```
char* scram_client_first::username
```

Definition at line 34 of file `scram/tokens.h`.

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

- [scram/tokens.h](#)

4.23 `scram_client_state` Struct Reference

Data Fields

- bool [plus](#)
- [Gsasl_hash](#) hash
- int [step](#)
- char * [cfmb](#)
- char * [serversignature](#)
- char * [authmessage](#)
- struct [scram_client_first](#) cf
- struct [scram_server_first](#) sf
- struct [scram_client_final](#) cl
- struct [scram_server_final](#) sl

4.23.1 Detailed Description

Definition at line 47 of file `scram/client.c`.

4.23.2 Field Documentation

4.23.2.1 `authmessage`

```
char* scram_client_state::authmessage
```

Definition at line 54 of file `scram/client.c`.

4.23.2.2 `cf`

```
struct scram\_client\_first scram_client_state::cf
```

Definition at line 54 of file `scram/client.c`.

4.23.2.3 cfmb

```
char* scram_client_state::cfmb
```

Definition at line 52 of file scram/client.c.

4.23.2.4 cl

```
struct scram\_client\_final scram_client_state::cl
```

Definition at line 54 of file scram/client.c.

4.23.2.5 hash

```
Gsas1\_hash scram_client_state::hash
```

Definition at line 50 of file scram/client.c.

4.23.2.6 plus

```
bool scram_client_state::plus
```

Definition at line 49 of file scram/client.c.

4.23.2.7 serversignature

```
char* scram_client_state::serversignature
```

Definition at line 53 of file scram/client.c.

4.23.2.8 sf

```
struct scram\_server\_first scram_client_state::sf
```

Definition at line 54 of file scram/client.c.

4.23.2.9 `sl`

```
struct scram_server_final scram_client_state::sl
```

Definition at line 54 of file `scram/client.c`.

4.23.2.10 `step`

```
int scram_client_state::step
```

Definition at line 51 of file `scram/client.c`.

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

- [scram/client.c](#)

4.24 `scram_server_final` Struct Reference

```
#include <tokens.h>
```

Data Fields

- char * `verifier`

4.24.1 Detailed Description

Definition at line 52 of file `scram/tokens.h`.

4.24.2 Field Documentation

4.24.2.1 `verifier`

```
char* scram_server_final::verifier
```

Definition at line 54 of file `scram/tokens.h`.

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

- [scram/tokens.h](#)

4.25 `scram_server_first` Struct Reference

```
#include <tokens.h>
```

Data Fields

- char * [nonce](#)
- char * [salt](#)
- size_t [iter](#)

4.25.1 Detailed Description

Definition at line 38 of file `scram/tokens.h`.

4.25.2 Field Documentation

4.25.2.1 `iter`

```
size_t scram_server_first::iter
```

Definition at line 42 of file `scram/tokens.h`.

4.25.2.2 `nonce`

```
char* scram_server_first::nonce
```

Definition at line 40 of file `scram/tokens.h`.

4.25.2.3 `salt`

```
char* scram_server_first::salt
```

Definition at line 41 of file `scram/tokens.h`.

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

- [scram/tokens.h](#)

4.26 `scram_server_state` Struct Reference

Data Fields

- bool `plus`
- `Gsasl_hash` hash
- int `step`
- char * `cbind`
- char * `gs2header`
- char * `cfmb_str`
- char * `sf_str`
- char * `snonce`
- char * `clientproof`
- char `storedkey` [GSASL_HASH_MAX_SIZE]
- char `serverkey` [GSASL_HASH_MAX_SIZE]
- char * `authmessage`
- char * `cb`
- size_t `cblen`
- struct `scram_client_first` cf
- struct `scram_server_first` sf
- struct `scram_client_final` cl
- struct `scram_server_final` sl

4.26.1 Detailed Description

Definition at line 51 of file `scram/server.c`.

4.26.2 Field Documentation

4.26.2.1 `authmessage`

```
char* scram_server_state::authmessage
```

Definition at line 64 of file `scram/server.c`.

4.26.2.2 `cb`

```
char* scram_server_state::cb
```

Definition at line 65 of file `scram/server.c`.

4.26.2.3 cbind

```
char* scram_server_state::cbind
```

Definition at line 56 of file scram/server.c.

4.26.2.4 cblen

```
size_t scram_server_state::cblen
```

Definition at line 66 of file scram/server.c.

4.26.2.5 cf

```
struct scram\_client\_first scram_server_state::cf
```

Definition at line 66 of file scram/server.c.

4.26.2.6 cfmb_str

```
char* scram_server_state::cfmb_str
```

Definition at line 58 of file scram/server.c.

4.26.2.7 cl

```
struct scram\_client\_final scram_server_state::cl
```

Definition at line 66 of file scram/server.c.

4.26.2.8 clientproof

```
char* scram_server_state::clientproof
```

Definition at line 61 of file scram/server.c.

4.26.2.9 `gs2header`

```
char* scram_server_state::gs2header
```

Definition at line 57 of file `scram/server.c`.

4.26.2.10 `hash`

```
Gsas1_hash scram_server_state::hash
```

Definition at line 54 of file `scram/server.c`.

4.26.2.11 `plus`

```
bool scram_server_state::plus
```

Definition at line 53 of file `scram/server.c`.

4.26.2.12 `serverkey`

```
char scram_server_state::serverkey[GSASL_HASH_MAX_SIZE]
```

Definition at line 63 of file `scram/server.c`.

4.26.2.13 `sf`

```
struct scram_server_first scram_server_state::sf
```

Definition at line 66 of file `scram/server.c`.

4.26.2.14 `sf_str`

```
char* scram_server_state::sf_str
```

Definition at line 59 of file `scram/server.c`.

4.26.2.15 `sl`

```
struct scram_server_final scram_server_state::sl
```

Definition at line 66 of file `scram/server.c`.

4.26.2.16 `snonce`

```
char* scram_server_state::snonce
```

Definition at line 60 of file `scram/server.c`.

4.26.2.17 `step`

```
int scram_server_state::step
```

Definition at line 55 of file `scram/server.c`.

4.26.2.18 `storedkey`

```
char scram_server_state::storedkey[GSASL_HASH_MAX_SIZE]
```

Definition at line 62 of file `scram/server.c`.

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

- [scram/server.c](#)

Chapter 5

File Documentation

5.1 anonymous.h File Reference

```
#include <gsasl.h>
```

Macros

- `#define GSASL_ANONYMOUS_NAME "ANONYMOUS"`

Functions

- `int _gsasl_anonymous_client_step (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)`
- `int _gsasl_anonymous_server_step (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)`

Variables

- `Gsasl_mechanism _gsasl_anonymous_mechanism`

5.1.1 Macro Definition Documentation

5.1.1.1 GSASL_ANONYMOUS_NAME

```
#define GSASL_ANONYMOUS_NAME "ANONYMOUS"
```

Definition at line 28 of file anonymous.h.

5.1.2 Function Documentation

5.1.2.1 `_gsasl_anonymous_client_step()`

```
int _gsasl_anonymous_client_step (
    Gsasl_session * sctx,
    void * mech_data,
    const char * input,
    size_t input_len,
    char ** output,
    size_t * output_len )
```

5.1.2.2 `_gsasl_anonymous_server_step()`

```
int _gsasl_anonymous_server_step (
    Gsasl_session * sctx,
    void * mech_data,
    const char * input,
    size_t input_len,
    char ** output,
    size_t * output_len )
```

5.1.3 Variable Documentation

5.1.3.1 `_gsasl_anonymous_mechanism`

`Gsasl_mechanism` `_gsasl_anonymous_mechanism` [extern]

Definition at line 28 of file anonymous/mechinfo.c.

5.2 `base64.c` File Reference

```
#include <config.h>
#include "internal.h"
#include "base64.h"
#include "mechtools.h"
```

Functions

- int [gsasl_base64_to](#) (const char *in, size_t inlen, char **out, size_t *outlen)
- int [gsasl_base64_from](#) (const char *in, size_t inlen, char **out, size_t *outlen)
- int [gsasl_hex_to](#) (const char *in, size_t inlen, char **out, size_t *outlen)
- int [gsasl_hex_from](#) (const char *in, char **out, size_t *outlen)

5.2.1 Function Documentation

5.2.1.1 [gsasl_base64_from\(\)](#)

```
int gsasl_base64_from (
    const char * in,
    size_t inlen,
    char ** out,
    size_t * outlen )
```

[gsasl_base64_from](#):

Parameters

<i>in</i>	input byte array
<i>inlen</i>	size of input byte array
<i>out</i>	pointer to newly allocated output byte array
<i>outlen</i>	pointer to size of newly allocated output byte array

Decode Base64 data. The @out buffer must be deallocated by the caller.

Return value: Returns GSASL_OK on success, GSASL_BASE64_ERROR if input was invalid, and GSASL_MALLOC_ERROR on memory allocation errors.

Since: 0.2.2

Definition at line 75 of file base64.c.

5.2.1.2 [gsasl_base64_to\(\)](#)

```
int gsasl_base64_to (
    const char * in,
    size_t inlen,
    char ** out,
    size_t * outlen )
```

[gsasl_base64_to](#):

Parameters

<i>in</i>	input byte array.
<i>inlen</i>	size of input byte array.
<i>out</i>	pointer to newly allocated base64-encoded string.
<i>outlen</i>	pointer to size of newly allocated base64-encoded string.

Encode data as base64. The @out string is zero terminated, and @outlen holds the length excluding the terminating zero. The @out buffer must be deallocated by the caller.

Return value: Returns GSASL_OK on success, or GSASL_MALLOC_ERROR if input was too large or memory allocation fail.

Since: 0.2.2

Definition at line 45 of file base64.c.

5.2.1.3 gsasl_hex_from()

```
int gsasl_hex_from (
    const char * in,
    char ** out,
    size_t * outlen )
```

gsasl_hex_from:

Parameters

<i>in</i>	input byte array
<i>out</i>	pointer to newly allocated output byte array
<i>outlen</i>	pointer to size of newly allocated output byte array

Decode hex data. The @out buffer must be deallocated by the caller.

Return value: Returns GSASL_OK on success, GSASL_BASE64_ERROR if input was invalid, and GSASL_MALLOC_ERROR on memory allocation errors.

Since: 1.10

Definition at line 144 of file base64.c.

5.2.1.4 gsasl_hex_to()

```
int gsasl_hex_to (
    const char * in,
    size_t inlen,
    char ** out,
    size_t * outlen )
```

gsasl_hex_to:

Parameters

<i>in</i>	input byte array.
<i>inlen</i>	size of input byte array.
<i>out</i>	pointer to newly allocated hex-encoded string.
<i>outlen</i>	pointer to size of newly allocated hex-encoded string.

Hex encode data. The @out string is zero terminated, and @outlen holds the length excluding the terminating zero. The @out buffer must be deallocated by the caller.

Return value: Returns GSASL_OK on success, or GSASL_MALLOC_ERROR if input was too large or memory allocation fail.

Since: 1.10

Definition at line 111 of file base64.c.

5.3 callback.c File Reference

```
#include <config.h>
#include "internal.h"
```

Functions

- void [gsasl_callback_set](#) ([Gsasl](#) *ctx, [Gsasl_callback_function](#) cb)
- int [gsasl_callback](#) ([Gsasl](#) *ctx, [Gsasl_session](#) *sctx, [Gsasl_property](#) prop)
- void [gsasl_callback_hook_set](#) ([Gsasl](#) *ctx, void *hook)
- void * [gsasl_callback_hook_get](#) ([Gsasl](#) *ctx)
- void [gsasl_session_hook_set](#) ([Gsasl_session](#) *sctx, void *hook)
- void * [gsasl_session_hook_get](#) ([Gsasl_session](#) *sctx)

5.3.1 Function Documentation

5.3.1.1 gsasl_callback()

```
int gsasl_callback (
    Gsasl * ctx,
    Gsasl_session * sctx,
    Gsasl_property prop )
```

gsasl_callback:

Parameters

<i>ctx</i>	handle received from gsasl_init() , may be NULL to derive it from @sctx.
<i>sctx</i>	session handle.
<i>prop</i>	enumerated value of Gsasl_property type.

Invoke the application callback. The @prop value indicate what the callback is expected to do. For example, for GSASL_ANONYMOUS_TOKEN, the function is expected to invoke `gsasl_property_set(@SCTX, GSASL_↔ ANONYMOUS_TOKEN, "token")` where "token" is the anonymous token the application wishes the SASL mechanism to use. See the manual for the meaning of all parameters.

Return value: Returns whatever the application callback returns, or GSASL_NO_CALLBACK if no application was known.

Since: 0.2.0

Definition at line 71 of file callback.c.

5.3.1.2 `gsasl_callback_hook_get()`

```
void* gsasl_callback_hook_get (
    Gsasl * ctx )
```

`gsasl_callback_hook_get`:

Parameters

<i>ctx</i>	libgsasl handle.
------------	------------------

Retrieve application specific data from libgsasl handle.

The application data is set using `gsasl_callback_hook_set()`. This is normally used by the application to maintain a global state between the main program and callbacks.

Return value: Returns the application specific data, or NULL.

Since: 0.2.0

Definition at line 120 of file callback.c.

5.3.1.3 `gsasl_callback_hook_set()`

```
void gsasl_callback_hook_set (
    Gsasl * ctx,
    void * hook )
```

`gsasl_callback_hook_set`:

Parameters

<i>ctx</i>	libgsasl handle.
<i>hook</i>	opaque pointer to application specific data.

Store application specific data in the libgsasl handle.

The application data can be later (for instance, inside a callback) be retrieved by calling [gsasl_callback_hook_get\(\)](#). This is normally used by the application to maintain a global state between the main program and callbacks.

Since: 0.2.0

Definition at line 100 of file callback.c.

5.3.1.4 gsasl_callback_set()

```
void gsasl_callback_set (
    Gsasl * ctx,
    Gsasl_callback_function cb )
```

gsasl_callback_set:

Parameters

<i>ctx</i>	handle received from gsasl_init() .
<i>cb</i>	pointer to function implemented by application.

Store the pointer to the application provided callback in the library handle. The callback will be used, via [gsasl_callback\(\)](#), by mechanisms to discover various parameters (such as username and passwords). The callback function will be called with a `Gsasl_property` value indicating the requested behaviour. For example, for `GSASL_↔ ANONYMOUS_TOKEN`, the function is expected to invoke `gsasl_property_set(@CTX, GSASL_↔ ANONYMOUS_TOKEN, "token")` where "token" is the anonymous token the application wishes the SASL mechanism to use. See the manual for the meaning of all parameters.

Since: 0.2.0

Definition at line 45 of file callback.c.

5.3.1.5 gsasl_session_hook_get()

```
void* gsasl_session_hook_get (
    Gsasl_session * sctx )
```

gsasl_session_hook_get:

Parameters

<i>sctx</i>	libgsasl session handle.
-------------	--------------------------

Retrieve application specific data from libgsasl session handle.

The application data is set using [gsasl_callback_hook_set\(\)](#). This is normally used by the application to maintain a per-session state between the main program and callbacks.

Return value: Returns the application specific data, or NULL.

Since: 0.2.14

Definition at line 160 of file callback.c.

5.3.1.6 gsasl_session_hook_set()

```
void gsasl_session_hook_set (
    Gsasl_session * sctx,
    void * hook )
```

gsasl_session_hook_set:

Parameters

<i>sctx</i>	libgsasl session handle.
<i>hook</i>	opaque pointer to application specific data.

Store application specific data in the libgsasl session handle.

The application data can be later (for instance, inside a callback) be retrieved by calling [gsasl_session_hook_get\(\)](#). This is normally used by the application to maintain a per-session state between the main program and callbacks.

Since: 0.2.14

Definition at line 140 of file callback.c.

5.4 challenge.c File Reference

```
#include <config.h>
#include <stdio.h>
#include <string.h>
#include <assert.h>
#include "challenge.h"
#include <gc.h>
```

Macros

- #define [NONCELEN](#) 10
- #define [TEMPLATE](#) "<XXXXXXXXXXXXXXXXXXXXXXXXX.0@localhost>"
- #define [DIGIT](#)(c)

Functions

- int [cram_md5_challenge](#) (char challenge[[CRAM_MD5_CHALLENGE_LEN](#)])

5.4.1 Macro Definition Documentation

5.4.1.1 DIGIT

```
#define DIGIT(  
    c )
```

Value:

```
((c) & 0x0F) > 9 ? \   
'0' + ((c) & 0x0F) - 10 : \   
'0' + ((c) & 0x0F)
```

Definition at line 61 of file challenge.c.

5.4.1.2 NONCELEN

```
#define NONCELEN 10
```

Definition at line 56 of file challenge.c.

5.4.1.3 TEMPLATE

```
#define TEMPLATE "<XXXXXXXXXXXXXXXXXXXXX.0@localhost>"
```

Definition at line 57 of file challenge.c.

5.4.2 Function Documentation

5.4.2.1 cram_md5_challenge()

```
int cram_md5_challenge (  
    char challenge[CRAM_MD5_CHALLENGE_LEN] )
```

Definition at line 66 of file challenge.c.

5.5 challenge.h File Reference

Macros

- [#define CRAM_MD5_CHALLENGE_LEN 35](#)

Functions

- int [cram_md5_challenge](#) (char challenge[[CRAM_MD5_CHALLENGE_LEN](#)])

5.5.1 Macro Definition Documentation

5.5.1.1 CRAM_MD5_CHALLENGE_LEN

```
#define CRAM_MD5_CHALLENGE_LEN 35
```

Definition at line 26 of file challenge.h.

5.5.2 Function Documentation

5.5.2.1 [cram_md5_challenge\(\)](#)

```
int cram\_md5\_challenge (  
    char challenge[CRAM\_MD5\_CHALLENGE\_LEN] )
```

Definition at line 66 of file challenge.c.

5.6 client.c File Reference

```
#include <config.h>  
#include "anonymous.h"  
#include <string.h>
```

Functions

- int [_gsasl_anonymous_client_step](#) ([Gsasl_session](#) *sctx, void *mech_data_GL_UNUSED, const char *input_GL_UNUSED, size_t input_len_GL_UNUSED, char **output, size_t *output_len)

5.6.1 Function Documentation

5.6.1.1 `_gsasl_anonymous_client_step()`

```
int _gsasl_anonymous_client_step (
    Gsasl_session * sctx,
    void *mech_data _GL_UNUSED,
    const char *input _GL_UNUSED,
    size_t input_len _GL_UNUSED,
    char ** output,
    size_t * output_len )
```

Definition at line 32 of file anonymous/client.c.

5.7 client.c File Reference

```
#include <config.h>
#include "cram-md5.h"
#include <stdlib.h>
#include <string.h>
#include "digest.h"
```

Functions

- `int _gsasl_cram_md5_client_step(Gsasl_session *sctx, void *mech_data _GL_UNUSED, const char *input, size_t input_len, char **output, size_t *output_len)`

5.7.1 Function Documentation

5.7.1.1 `_gsasl_cram_md5_client_step()`

```
int _gsasl_cram_md5_client_step (
    Gsasl_session * sctx,
    void *mech_data _GL_UNUSED,
    const char * input,
    size_t input_len,
    char ** output,
    size_t * output_len )
```

Definition at line 38 of file cram-md5/client.c.

5.8 client.c File Reference

```
#include <config.h>
#include "digest-md5.h"
#include <stdlib.h>
#include <string.h>
#include "gc.h"
#include "nonascii.h"
#include "tokens.h"
#include "parser.h"
#include "printer.h"
#include "free.h"
#include "session.h"
#include "digesthmac.h"
#include "qop.h"
#include "mechtools.h"
```

Data Structures

- [struct _Gsasl_digest_md5_client_state](#)

Macros

- [#define CNONCE_ENTROPY_BYTES 16](#)

Typedefs

- [typedef struct _Gsasl_digest_md5_client_state _Gsasl_digest_md5_client_state](#)

Functions

- [int _gsasl_digest_md5_client_start](#) ([Gsasl_session](#) *sctx, [_GL_UNUSED](#), void **mech_data)
- [int _gsasl_digest_md5_client_step](#) ([Gsasl_session](#) *sctx, void *mech_data, const char *input, [size_t](#) input_len, char **output, [size_t](#) *output_len)
- [void _gsasl_digest_md5_client_finish](#) ([Gsasl_session](#) *sctx, [_GL_UNUSED](#), void *mech_data)
- [int _gsasl_digest_md5_client_encode](#) ([Gsasl_session](#) *sctx, [_GL_UNUSED](#), void *mech_data, const char *input, [size_t](#) input_len, char **output, [size_t](#) *output_len)
- [int _gsasl_digest_md5_client_decode](#) ([Gsasl_session](#) *sctx, [_GL_UNUSED](#), void *mech_data, const char *input, [size_t](#) input_len, char **output, [size_t](#) *output_len)

5.8.1 Macro Definition Documentation

5.8.1.1 CNONCE_ENTROPY_BYTES

```
#define CNONCE_ENTROPY_BYTES 16
```

Definition at line 48 of file digest-md5/client.c.

5.8.2 Typedef Documentation

5.8.2.1 `_Gsasl_digest_md5_client_state`

```
typedef struct _Gsasl_digest_md5_client_state _Gsasl_digest_md5_client_state
```

Definition at line 1 of file digest-md5/client.c.

5.8.3 Function Documentation

5.8.3.1 `_gsasl_digest_md5_client_decode()`

```
int _gsasl_digest_md5_client_decode (  
    Gsasl_session *sctx _GL_UNUSED,  
    void * mech_data,  
    const char * input,  
    size_t input_len,  
    char ** output,  
    size_t * output_len )
```

Definition at line 329 of file digest-md5/client.c.

5.8.3.2 `_gsasl_digest_md5_client_encode()`

```
int _gsasl_digest_md5_client_encode (  
    Gsasl_session *sctx _GL_UNUSED,  
    void * mech_data,  
    const char * input,  
    size_t input_len,  
    char ** output,  
    size_t * output_len )
```

Definition at line 305 of file digest-md5/client.c.

5.8.3.3 `_gsasl_digest_md5_client_finish()`

```
void _gsasl_digest_md5_client_finish (  
    Gsasl_session *sctx _GL_UNUSED,  
    void * mech_data )
```

Definition at line 289 of file digest-md5/client.c.

5.8.3.4 `_gsasl_digest_md5_client_start()`

```
int _gsasl_digest_md5_client_start (
    Gsasl_session *sctx _GL_UNUSED,
    void ** mech_data )
```

Definition at line 66 of file digest-md5/client.c.

5.8.3.5 `_gsasl_digest_md5_client_step()`

```
int _gsasl_digest_md5_client_step (
    Gsasl_session * sctx,
    void * mech_data,
    const char * input,
    size_t input_len,
    char ** output,
    size_t * output_len )
```

Definition at line 98 of file digest-md5/client.c.

5.9 client.c File Reference

```
#include <config.h>
#include "external.h"
#include <string.h>
```

Functions

- int `_gsasl_external_client_step` (`Gsasl_session` *sctx, void *mech_data `_GL_UNUSED`, const char *input `_GL_UNUSED`, size_t input_len `_GL_UNUSED`, char **output, size_t *output_len)

5.9.1 Function Documentation

5.9.1.1 `_gsasl_external_client_step()`

```
int _gsasl_external_client_step (
    Gsasl_session * sctx,
    void *mech_data _GL_UNUSED,
    const char *input _GL_UNUSED,
    size_t input_len _GL_UNUSED,
    char ** output,
    size_t * output_len )
```

Definition at line 32 of file external/client.c.

5.10 client.c File Reference

```
#include <config.h>
#include "gs2.h"
#include <stdlib.h>
#include <string.h>
#include "gss-extra.h"
#include "gs2helper.h"
```

Data Structures

- [struct _gsasl_gs2_client_state](#)

Typedefs

- typedef struct [_gsasl_gs2_client_state](#) [_gsasl_gs2_client_state](#)

Functions

- [int _gsasl_gs2_client_start](#) ([Gsasl_session](#) *sctx, void **mech_data)
- [int _gsasl_gs2_client_step](#) ([Gsasl_session](#) *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)
- [void _gsasl_gs2_client_finish](#) ([Gsasl_session](#) *sctx, void *mech_data)

5.10.1 Typedef Documentation

5.10.1.1 [_gsasl_gs2_client_state](#)

```
typedef struct \_gsasl\_gs2\_client\_state \_gsasl\_gs2\_client\_state
```

Definition at line 1 of file gs2/client.c.

5.10.2 Function Documentation

5.10.2.1 [_gsasl_gs2_client_finish\(\)](#)

```
void \_gsasl\_gs2\_client\_finish (  
    Gsasl\_session * sctx,  
    void * mech_data )
```

Definition at line 316 of file gs2/client.c.

5.10.2.2 `_gsasl_gs2_client_start()`

```
int _gsasl_gs2_client_start (
    Gsasl_session * sctx,
    void ** mech_data )
```

Definition at line 52 of file gs2/client.c.

5.10.2.3 `_gsasl_gs2_client_step()`

```
int _gsasl_gs2_client_step (
    Gsasl_session * sctx,
    void * mech_data,
    const char * input,
    size_t input_len,
    char ** output,
    size_t * output_len )
```

Definition at line 235 of file gs2/client.c.

5.11 `client.c` File Reference

```
#include <config.h>
#include <stdlib.h>
#include <string.h>
#include "x-gssapi.h"
#include "gss-extra.h"
```

Data Structures

- struct [_Gsasl_gssapi_client_state](#)

Typedefs

- typedef struct [_Gsasl_gssapi_client_state](#) [_Gsasl_gssapi_client_state](#)

Functions

- int [_gsasl_gssapi_client_start](#)(Gsasl_session *sctx, void **mech_data)
- int [_gsasl_gssapi_client_step](#)(Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)
- void [_gsasl_gssapi_client_finish](#)(Gsasl_session *sctx, void *mech_data)
- int [_gsasl_gssapi_client_encode](#)(Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)
- int [_gsasl_gssapi_client_decode](#)(Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)

5.11.1 Typedef Documentation

5.11.1.1 `_Gsasl_gssapi_client_state`

```
typedef struct _Gsasl_gssapi_client_state _Gsasl_gssapi_client_state
```

Definition at line 1 of file gssapi/client.c.

5.11.2 Function Documentation

5.11.2.1 `__gsasl_gssapi_client_decode()`

```
int __gsasl_gssapi_client_decode (
    Gsasl_session * sctx,
    void * mech_data,
    const char * input,
    size_t input_len,
    char ** output,
    size_t * output_len )
```

Definition at line 322 of file gssapi/client.c.

5.11.2.2 `__gsasl_gssapi_client_encode()`

```
int __gsasl_gssapi_client_encode (
    Gsasl_session * sctx,
    void * mech_data,
    const char * input,
    size_t input_len,
    char ** output,
    size_t * output_len )
```

Definition at line 267 of file gssapi/client.c.

5.11.2.3 `__gsasl_gssapi_client_finish()`

```
void __gsasl_gssapi_client_finish (
    Gsasl_session * sctx,
    void * mech_data )
```

Definition at line 249 of file gssapi/client.c.

5.11.2.4 `_gsasl_gssapi_client_start()`

```
int _gsasl_gssapi_client_start (
    Gsasl_session * sctx,
    void ** mech_data )
```

Definition at line 47 of file gssapi/client.c.

5.11.2.5 `_gsasl_gssapi_client_step()`

```
int _gsasl_gssapi_client_step (
    Gsasl_session * sctx,
    void * mech_data,
    const char * input,
    size_t input_len,
    char ** output,
    size_t * output_len )
```

Definition at line 66 of file gssapi/client.c.

5.12 `client.c` File Reference

```
#include <config.h>
#include <stdlib.h>
#include <string.h>
#include "login.h"
```

Data Structures

- struct [_Gsasl_login_client_state](#)

Functions

- int [_gsasl_login_client_start](#) ([Gsasl_session](#) *sctx [_GL_UNUSED](#), void **mech_data)
- int [_gsasl_login_client_step](#) ([Gsasl_session](#) *sctx [_GL_UNUSED](#), void *mech_data, const char *input [↔](#) [_GL_UNUSED](#), size_t input_len [_GL_UNUSED](#), char **output, size_t *output_len)
- void [_gsasl_login_client_finish](#) ([Gsasl_session](#) *sctx [_GL_UNUSED](#), void *mech_data)

5.12.1 Function Documentation

5.12.1.1 `_gsasl_login_client_finish()`

```
void _gsasl_login_client_finish (
    Gsasl_session *sctx _GL_UNUSED,
    void * mech_data )
```

Definition at line 103 of file login/client.c.

5.12.1.2 `_gsasl_login_client_start()`

```
int _gsasl_login_client_start (
    Gsasl_session *sctx _GL_UNUSED,
    void ** mech_data )
```

Definition at line 40 of file login/client.c.

5.12.1.3 `_gsasl_login_client_step()`

```
int _gsasl_login_client_step (
    Gsasl_session *sctx _GL_UNUSED,
    void * mech_data,
    const char *input _GL_UNUSED,
    size_t input_len _GL_UNUSED,
    char ** output,
    size_t * output_len )
```

Definition at line 56 of file login/client.c.

5.13 client.c File Reference

```
#include <config.h>
#include "openid20.h"
#include <string.h>
#include <stdlib.h>
#include <stdbool.h>
#include "mechtools.h"
```

Data Structures

- struct [openid20_client_state](#)

Macros

- #define [ERR_PREFIX](#) "openid.error="

Functions

- `int _gsasl_openid20_client_start(Gsasl_session *sctx _GL_UNUSED, void **mech_data)`
- `int _gsasl_openid20_client_step(Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)`
- `void _gsasl_openid20_client_finish(Gsasl_session *sctx _GL_UNUSED, void *mech_data)`

5.13.1 Macro Definition Documentation

5.13.1.1 ERR_PREFIX

```
#define ERR_PREFIX "openid.error="
```

5.13.2 Function Documentation

5.13.2.1 _gsasl_openid20_client_finish()

```
void _gsasl_openid20_client_finish (  
    Gsasl_session *sctx _GL_UNUSED,  
    void * mech_data )
```

Definition at line 167 of file openid20/client.c.

5.13.2.2 _gsasl_openid20_client_start()

```
int _gsasl_openid20_client_start (  
    Gsasl_session *sctx _GL_UNUSED,  
    void ** mech_data )
```

Definition at line 46 of file openid20/client.c.

5.13.2.3 _gsasl_openid20_client_step()

```
int _gsasl_openid20_client_step (  
    Gsasl_session * sctx,  
    void * mech_data,  
    const char * input,  
    size_t input_len,  
    char ** output,  
    size_t * output_len )
```

Definition at line 61 of file openid20/client.c.

5.14 client.c File Reference

```
#include <config.h>
#include "plain.h"
#include <string.h>
#include <stdlib.h>
```

Functions

- int [_gsasl_plain_client_step](#) ([Gsasl_session](#) *sctx, void *mech_data *_GL_UNUSED*, const char *input *_GL_UNUSED*, size_t input_len *_GL_UNUSED*, char **output, size_t *output_len)

5.14.1 Function Documentation

5.14.1.1 [_gsasl_plain_client_step\(\)](#)

```
int _gsasl_plain_client_step (
    Gsasl\_session * sctx,
    void *mech_data _GL_UNUSED,
    const char *input _GL_UNUSED,
    size_t input_len _GL_UNUSED,
    char ** output,
    size_t * output_len )
```

Definition at line 35 of file plain/client.c.

5.15 client.c File Reference

```
#include <config.h>
#include "saml20.h"
#include <string.h>
#include <stdlib.h>
#include <stdbool.h>
#include "mechtools.h"
```

Data Structures

- struct [saml20_client_state](#)

Functions

- int [_gsasl_saml20_client_start](#) ([Gsasl_session](#) *sctx *_GL_UNUSED*, void **mech_data)
- int [_gsasl_saml20_client_step](#) ([Gsasl_session](#) *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)
- void [_gsasl_saml20_client_finish](#) ([Gsasl_session](#) *sctx *_GL_UNUSED*, void *mech_data)

5.15.1 Function Documentation

5.15.1.1 `_gsasl_saml20_client_finish()`

```
void _gsasl_saml20_client_finish (
    Gsasl_session *sctx _GL_UNUSED,
    void * mech_data )
```

Definition at line 120 of file `saml20/client.c`.

5.15.1.2 `_gsasl_saml20_client_start()`

```
int _gsasl_saml20_client_start (
    Gsasl_session *sctx _GL_UNUSED,
    void ** mech_data )
```

Definition at line 46 of file `saml20/client.c`.

5.15.1.3 `_gsasl_saml20_client_step()`

```
int _gsasl_saml20_client_step (
    Gsasl_session * sctx,
    void * mech_data,
    const char * input,
    size_t input_len,
    char ** output,
    size_t * output_len )
```

Definition at line 60 of file `saml20/client.c`.

5.16 `client.c` File Reference

```
#include <config.h>
#include "scram.h"
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#include "tokens.h"
#include "parser.h"
#include "printer.h"
#include "gc.h"
#include "memxor.h"
#include "tools.h"
#include "mechtools.h"
```


Data Structures

- struct [scram_client_state](#)

Macros

- #define [CNONCE_ENTROPY_BYTES](#) 18

Functions

- int [_gsasl_scram_client_step](#) ([Gsasl_session](#) *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)
- void [_gsasl_scram_client_finish](#) ([Gsasl_session](#) *sctx, [_GL_UNUSED](#), void *mech_data)

5.16.1 Macro Definition Documentation

5.16.1.1 CNONCE_ENTROPY_BYTES

```
#define CNONCE_ENTROPY_BYTES 18
```

Definition at line 45 of file `scram/client.c`.

5.16.2 Function Documentation

5.16.2.1 _gsasl_scram_client_finish()

```
void _gsasl_scram_client_finish (  
    Gsasl\_session *sctx, \_GL\_UNUSED,  
    void * mech_data )
```

Definition at line 423 of file `scram/client.c`.

5.16.2.2 _gsasl_scram_client_step()

```
int _gsasl_scram_client_step (  
    Gsasl\_session * sctx,  
    void * mech_data,  
    const char * input,  
    size_t input_len,  
    char ** output,  
    size_t * output_len )
```

Definition at line 125 of file `scram/client.c`.

5.17 client.c File Reference

```
#include <config.h>
#include "securid.h"
#include <stdlib.h>
#include <string.h>
```

Macros

- #define `PASSCODE` "passcode"
- #define `PIN` "pin"

Functions

- int `_gsasl_securid_client_start` (`Gsasl_session` *sctx, `_GL_UNUSED`, void **mech_data)
- int `_gsasl_securid_client_step` (`Gsasl_session` *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)
- void `_gsasl_securid_client_finish` (`Gsasl_session` *sctx, `_GL_UNUSED`, void *mech_data)

5.17.1 Macro Definition Documentation

5.17.1.1 PASSCODE

```
#define PASSCODE "passcode"
```

Definition at line 34 of file securid/client.c.

5.17.1.2 PIN

```
#define PIN "pin"
```

Definition at line 35 of file securid/client.c.

5.17.2 Function Documentation

5.17.2.1 `_gsasl_secured_client_finish()`

```
void _gsasl_secured_client_finish (
    Gsasl_session *sctx _GL_UNUSED,
    void * mech_data )
```

Definition at line 165 of file `secured/client.c`.

5.17.2.2 `_gsasl_secured_client_start()`

```
int _gsasl_secured_client_start (
    Gsasl_session *sctx _GL_UNUSED,
    void ** mech_data )
```

Definition at line 38 of file `secured/client.c`.

5.17.2.3 `_gsasl_secured_client_step()`

```
int _gsasl_secured_client_step (
    Gsasl_session * sctx,
    void * mech_data,
    const char * input,
    size_t input_len,
    char ** output,
    size_t * output_len )
```

Definition at line 54 of file `secured/client.c`.

5.18 `cram-md5.h` File Reference

```
#include <gsasl.h>
```

Macros

- #define `GSASL_CRAM_MD5_NAME` "CRAM-MD5"

Functions

- int `_gsasl_cram_md5_client_step` (`Gsasl_session` *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)
- int `_gsasl_cram_md5_server_start` (`Gsasl_session` *sctx, void **mech_data)
- int `_gsasl_cram_md5_server_step` (`Gsasl_session` *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)
- void `_gsasl_cram_md5_server_finish` (`Gsasl_session` *sctx, void *mech_data)

Variables

- [Gsasl_mechanism_gsasl_cram_md5_mechanism](#)

5.18.1 Macro Definition Documentation

5.18.1.1 GSASL_CRAM_MD5_NAME

```
#define GSASL_CRAM_MD5_NAME "CRAM-MD5"
```

Definition at line 28 of file cram-md5.h.

5.18.2 Function Documentation

5.18.2.1 __gsasl_cram_md5_client_step()

```
int __gsasl_cram_md5_client_step (  
    Gsasl_session * sctx,  
    void * mech_data,  
    const char * input,  
    size_t input_len,  
    char ** output,  
    size_t * output_len )
```

5.18.2.2 __gsasl_cram_md5_server_finish()

```
void __gsasl_cram_md5_server_finish (  
    Gsasl_session * sctx,  
    void * mech_data )
```

5.18.2.3 __gsasl_cram_md5_server_start()

```
int __gsasl_cram_md5_server_start (  
    Gsasl_session * sctx,  
    void ** mech_data )
```

5.18.2.4 `_gsasl_cram_md5_server_step()`

```
int _gsasl_cram_md5_server_step (
    Gsasl_session * sctx,
    void * mech_data,
    const char * input,
    size_t input_len,
    char ** output,
    size_t * output_len )
```

Definition at line 66 of file `cram-md5/server.c`.

5.18.3 Variable Documentation

5.18.3.1 `_gsasl_cram_md5_mechanism`

`Gsasl_mechanism` `_gsasl_cram_md5_mechanism` [extern]

Definition at line 28 of file `cram-md5/mechinfo.c`.

5.19 crypto.c File Reference

```
#include <config.h>
#include "internal.h"
#include "mechtools.h"
#include "gc.h"
```

Macros

- `#define CLIENT_KEY` "Client Key"
- `#define SERVER_KEY` "Server Key"

Functions

- `int gsasl_nonce` (char *data, size_t datalen)
- `int gsasl_random` (char *data, size_t datalen)
- `size_t gsasl_hash_length` (`Gsasl_hash` hash)
- `int gsasl_scram_secrets_from_salted_password` (`Gsasl_hash` hash, const char *salted_password, char *client_key, char *server_key, char *stored_key)
- `int gsasl_scram_secrets_from_password` (`Gsasl_hash` hash, const char *password, unsigned int iteration_count, const char *salt, size_t saltlen, char *salted_password, char *client_key, char *server_key, char *stored_key)

5.19.1 Macro Definition Documentation

5.19.1.1 CLIENT_KEY

```
#define CLIENT_KEY "Client Key"
```

5.19.1.2 SERVER_KEY

```
#define SERVER_KEY "Server Key"
```

5.19.2 Function Documentation

5.19.2.1 gsasl_hash_length()

```
size_t gsasl_hash_length (  
    Gsasl_hash hash )
```

gsasl_hash_length:

Parameters

<i>hash</i>	a Gsasl_hash element, e.g., GSASL_HASH_SHA256 .
-------------	---

Return the digest output size for hash function @hash. For example, gsasl_hash_length(GSASL_HASH_SHA256) returns GSASL_HASH_SHA256_SIZE which is 32.

Returns: size of supplied Gsasl_hash element.

Since: 1.10

Definition at line 73 of file crypto.c.

5.19.2.2 gsasl_nonce()

```
int gsasl_nonce (  
    char * data,  
    size_t datalen )
```

gsasl_nonce:

Parameters

<i>data</i>	output array to be filled with unpredictable random data.
<i>datalen</i>	size of output array.

Store unpredictable data of given size in the provided buffer.

Return value: Returns GSASL_OK iff successful.

Definition at line 39 of file crypto.c.

5.19.2.3 gsasl_random()

```
int gsasl_random (
    char * data,
    size_t datalen )
```

gsasl_random:

Parameters

<i>data</i>	output array to be filled with strong random data.
<i>datalen</i>	size of output array.

Store cryptographically strong random data of given size in the provided buffer.

Return value: Returns GSASL_OK iff successful.

Definition at line 55 of file crypto.c.

5.19.2.4 gsasl_scram_secrets_from_password()

```
int gsasl_scram_secrets_from_password (
    Gsasl_hash hash,
    const char * password,
    unsigned int iteration_count,
    const char * salt,
    size_t saltlen,
    char * salted_password,
    char * client_key,
    char * server_key,
    char * stored_key )
```

gsasl_scram_secrets_from_password:

Parameters

<i>hash</i>	a Gsasl_hash element, e.g., GSASL_HASH_SHA256 .
<i>password</i>	input parameter with password.
<i>iteration_count</i>	number of PBKDF2 rounds to apply.
<i>salt</i>	input character array of @saltlen length with salt for PBKDF2.
<i>saltlen</i>	length of @salt.
<i>salted_password</i>	pre-allocated output array with derived salted password.
<i>client_key</i>	pre-allocated output array with derived client key.
<i>server_key</i>	pre-allocated output array with derived server key.
<i>stored_key</i>	pre-allocated output array with derived stored key.

Helper function to generate SCRAM secrets from a password. The @salted_password, @client_key, @server_key, and @stored_key buffers must have room to hold digest for given @hash, use [GSASL_HASH_MAX_SIZE](#) which is sufficient for all hashes.

Return value: Returns GSASL_OK if successful, or error code.

Since: 1.10

Definition at line 156 of file crypto.c.

5.19.2.5 gsasl_scram_secrets_from_salted_password()

```
int gsasl_scram_secrets_from_salted_password (
    Gsasl_hash hash,
    const char * salted_password,
    char * client_key,
    char * server_key,
    char * stored_key )
```

gsasl_scram_secrets_from_salted_password:

Parameters

<i>hash</i>	a Gsasl_hash element, e.g., GSASL_HASH_SHA256 .
<i>salted_password</i>	input array with salted password.
<i>client_key</i>	pre-allocated output array with derived client key.
<i>server_key</i>	pre-allocated output array with derived server key.
<i>stored_key</i>	pre-allocated output array with derived stored key.

Helper function to derive SCRAM ClientKey/ServerKey/StoredKey. The @client_key, @server_key, and @stored_key buffers must have room to hold digest for given @hash, use [GSASL_HASH_MAX_SIZE](#) which is sufficient for all hashes.

Return value: Returns GSASL_OK if successful, or error code.

Since: 1.10

Definition at line 104 of file crypto.c.

5.20 digest-md5.h File Reference

```
#include <gsasl.h>
```

Macros

- `#define GSASL_DIGEST_MD5_NAME "DIGEST-MD5"`

Functions

- `int _gsasl_digest_md5_client_start (Gsasl_session *sctx, void **mech_data)`
- `int _gsasl_digest_md5_client_step (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)`
- `void _gsasl_digest_md5_client_finish (Gsasl_session *sctx, void *mech_data)`
- `int _gsasl_digest_md5_client_encode (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)`
- `int _gsasl_digest_md5_client_decode (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)`
- `int _gsasl_digest_md5_server_start (Gsasl_session *sctx, void **mech_data)`
- `int _gsasl_digest_md5_server_step (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)`
- `void _gsasl_digest_md5_server_finish (Gsasl_session *sctx, void *mech_data)`
- `int _gsasl_digest_md5_server_encode (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)`
- `int _gsasl_digest_md5_server_decode (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)`

Variables

- `Gsasl_mechanism_gsasl_digest_md5_mechanism`

5.20.1 Macro Definition Documentation

5.20.1.1 GSASL_DIGEST_MD5_NAME

```
#define GSASL_DIGEST_MD5_NAME "DIGEST-MD5"
```

Definition at line 28 of file digest-md5.h.

5.20.2 Function Documentation

5.20.2.1 `_gsasl_digest_md5_client_decode()`

```
int _gsasl_digest_md5_client_decode (
    Gsasl_session * sctx,
    void * mech_data,
    const char * input,
    size_t input_len,
    char ** output,
    size_t * output_len )
```

5.20.2.2 `_gsasl_digest_md5_client_encode()`

```
int _gsasl_digest_md5_client_encode (
    Gsasl_session * sctx,
    void * mech_data,
    const char * input,
    size_t input_len,
    char ** output,
    size_t * output_len )
```

5.20.2.3 `_gsasl_digest_md5_client_finish()`

```
void _gsasl_digest_md5_client_finish (
    Gsasl_session * sctx,
    void * mech_data )
```

5.20.2.4 `_gsasl_digest_md5_client_start()`

```
int _gsasl_digest_md5_client_start (
    Gsasl_session * sctx,
    void ** mech_data )
```

5.20.2.5 `_gsasl_digest_md5_client_step()`

```
int _gsasl_digest_md5_client_step (
    Gsasl_session * sctx,
    void * mech_data,
    const char * input,
    size_t input_len,
    char ** output,
    size_t * output_len )
```

Definition at line 98 of file digest-md5/client.c.

5.20.2.6 `_gsasl_digest_md5_server_decode()`

```
int _gsasl_digest_md5_server_decode (
    Gsasl_session * sctx,
    void * mech_data,
    const char * input,
    size_t input_len,
    char ** output,
    size_t * output_len )
```

5.20.2.7 `_gsasl_digest_md5_server_encode()`

```
int _gsasl_digest_md5_server_encode (
    Gsasl_session * sctx,
    void * mech_data,
    const char * input,
    size_t input_len,
    char ** output,
    size_t * output_len )
```

5.20.2.8 `_gsasl_digest_md5_server_finish()`

```
void _gsasl_digest_md5_server_finish (
    Gsasl_session * sctx,
    void * mech_data )
```

5.20.2.9 `_gsasl_digest_md5_server_start()`

```
int _gsasl_digest_md5_server_start (
    Gsasl_session * sctx,
    void ** mech_data )
```

5.20.2.10 `_gsasl_digest_md5_server_step()`

```
int _gsasl_digest_md5_server_step (
    Gsasl_session * sctx,
    void * mech_data,
    const char * input,
    size_t input_len,
    char ** output,
    size_t * output_len )
```

Definition at line 146 of file digest-md5/server.c.

5.20.3 Variable Documentation

5.20.3.1 `_gsasl_digest_md5_mechanism`

`Gsasl_mechanism` `_gsasl_digest_md5_mechanism` [extern]

Definition at line 28 of file digest-md5/mechinfo.c.

5.21 `digest.c` File Reference

```
#include <config.h>
#include <string.h>
#include "digest.h"
#include "gc.h"
```

Macros

- #define `HEXCHAR(c)` `((c & 0x0F) > 9 ? 'a' + (c & 0x0F) - 10 : '0' + (c & 0x0F))`

Functions

- void `cram_md5_digest` (const char *challenge, size_t challengelen, const char *secret, size_t secretlen, char response[`CRAM_MD5_DIGEST_LEN`])

5.21.1 Macro Definition Documentation

5.21.1.1 `HEXCHAR`

```
#define HEXCHAR(  
    c ) ((c & 0x0F) > 9 ? 'a' + (c & 0x0F) - 10 : '0' + (c & 0x0F))
```

Definition at line 57 of file digest.c.

5.21.2 Function Documentation

5.21.2.1 cram_md5_digest()

```
void cram_md5_digest (
    const char * challenge,
    size_t challengelen,
    const char * secret,
    size_t secretlen,
    char response[CRAM_MD5_DIGEST_LEN] )
```

Definition at line 60 of file digest.c.

5.22 digest.h File Reference

```
#include <stddef.h>
```

Macros

- #define [CRAM_MD5_DIGEST_LEN](#) 32

Functions

- void [cram_md5_digest](#) (const char *challenge, size_t challengelen, const char *secret, size_t secretlen, char response[[CRAM_MD5_DIGEST_LEN](#)])

5.22.1 Macro Definition Documentation

5.22.1.1 CRAM_MD5_DIGEST_LEN

```
#define CRAM_MD5_DIGEST_LEN 32
```

Definition at line 29 of file digest.h.

5.22.2 Function Documentation

5.22.2.1 cram_md5_digest()

```
void cram_md5_digest (
    const char * challenge,
    size_t challengelen,
    const char * secret,
    size_t secretlen,
    char response[CRAM_MD5_DIGEST_LEN] )
```

Definition at line 60 of file digest.c.

5.23.1.2 A2_PRE

```
#define A2_PRE "AUTHENTICATE:"
```

Definition at line 46 of file digestmac.c.

5.23.1.3 COLON

```
#define COLON ":"
```

Definition at line 48 of file digestmac.c.

5.23.1.4 DERIVE_CLIENT_CONFIDENTIALITY_KEY_STRING

```
#define DERIVE_CLIENT_CONFIDENTIALITY_KEY_STRING "Digest H(A1) to client-to-server sealing  
key magic constant"
```

Definition at line 56 of file digestmac.c.

5.23.1.5 DERIVE_CLIENT_CONFIDENTIALITY_KEY_STRING_LEN

```
#define DERIVE_CLIENT_CONFIDENTIALITY_KEY_STRING_LEN 59
```

Definition at line 58 of file digestmac.c.

5.23.1.6 DERIVE_CLIENT_INTEGRITY_KEY_STRING

```
#define DERIVE_CLIENT_INTEGRITY_KEY_STRING "Digest session key to client-to-server signing  
key magic constant"
```

Definition at line 50 of file digestmac.c.

5.23.1.7 DERIVE_CLIENT_INTEGRITY_KEY_STRING_LEN

```
#define DERIVE_CLIENT_INTEGRITY_KEY_STRING_LEN 65
```

Definition at line 52 of file digestmac.c.

5.23.1.8 DERIVE_SERVER_CONFIDENTIALITY_KEY_STRING

```
#define DERIVE_SERVER_CONFIDENTIALITY_KEY_STRING "Digest H(A1) to server-to-client sealing  
key magic constant"
```

Definition at line 59 of file digestmac.c.

5.23.1.9 DERIVE_SERVER_CONFIDENTIALITY_KEY_STRING_LEN

```
#define DERIVE_SERVER_CONFIDENTIALITY_KEY_STRING_LEN 59
```

Definition at line 61 of file digestmac.c.

5.23.1.10 DERIVE_SERVER_INTEGRITY_KEY_STRING

```
#define DERIVE_SERVER_INTEGRITY_KEY_STRING "Digest session key to server-to-client signing  
key magic constant"
```

Definition at line 53 of file digestmac.c.

5.23.1.11 DERIVE_SERVER_INTEGRITY_KEY_STRING_LEN

```
#define DERIVE_SERVER_INTEGRITY_KEY_STRING_LEN 65
```

Definition at line 55 of file digestmac.c.

5.23.1.12 HEXCHAR

```
#define HEXCHAR(  
    c ) ((c & 0x0F) > 9 ? 'a' + (c & 0x0F) - 10 : '0' + (c & 0x0F))
```

Definition at line 40 of file digestmac.c.

5.23.1.13 MD5LEN

```
#define MD5LEN 16
```

Definition at line 49 of file digestmac.c.

5.23.1.14 QOP_AUTH

```
#define QOP_AUTH "auth"
```

Definition at line 42 of file digestmac.c.

5.23.1.15 QOP_AUTH_CONF

```
#define QOP_AUTH_CONF "auth-conf"
```

Definition at line 44 of file digestmac.c.

5.23.1.16 QOP_AUTH_INT

```
#define QOP_AUTH_INT "auth-int"
```

Definition at line 43 of file digestmac.c.

5.23.2 Function Documentation

5.23.2.1 digest_md5_hmac()

```
int digest_md5_hmac (  
    char * output,  
    char secret[MD5LEN],  
    const char * nonce,  
    unsigned long nc,  
    const char * cnonce,  
    digest_md5_qop qop,  
    const char * authzid,  
    const char * digesturi,  
    int rspauth,  
    digest_md5_cipher cipher,  
    char * kic,  
    char * kis,  
    char * kcc,  
    char * kcs )
```

Definition at line 77 of file digestmac.c.

5.24 digestmac.h File Reference

```
#include "tokens.h"
```

Functions

- int [digest_md5_hmac](#) (char *output, char secret[[DIGEST_MD5_LENGTH](#)], const char *nonce, unsigned long nc, const char *cnonce, [digest_md5_qop](#) qop, const char *authzid, const char *digesturi, int rspauth, [digest_md5_cipher](#) cipher, char *kic, char *kis, char *kcc, char *kcs)

5.24.1 Function Documentation

5.24.1.1 [digest_md5_hmac\(\)](#)

```
int digest_md5_hmac (
    char * output,
    char secret [DIGEST\_MD5\_LENGTH],
    const char * nonce,
    unsigned long nc,
    const char * cnonce,
    digest\_md5\_qop qop,
    const char * authzid,
    const char * digesturi,
    int rspauth,
    digest\_md5\_cipher cipher,
    char * kic,
    char * kis,
    char * kcc,
    char * kcs )
```

5.25 done.c File Reference

```
#include <config.h>
#include "internal.h"
```

Functions

- void [gsasl_done](#) ([Gsasl](#) *ctx)

5.25.1 Function Documentation

5.25.1.1 [gsasl_done\(\)](#)

```
void gsasl_done (
    Gsasl * ctx )
```

[gsasl_done](#):

Parameters

<i>ctx</i>	libgsasl handle.
------------	------------------

This function destroys a libgsasl handle. The handle must not be used with other libgsasl functions after this call.

Definition at line 34 of file done.c.

5.26 doxygen.c File Reference

5.27 error.c File Reference

```
#include <config.h>
#include "internal.h"
#include "gettext.h"
```

Macros

- #define `_(String) dgettext (PACKAGE, String)`
- #define `gettext_noop(String) String`
- #define `N_(String) gettext_noop (String)`
- #define `ERR(name, desc) { name, #name, desc }`

Functions

- const char * `gsasl_strerror (int err)`
- const char * `gsasl_strerror_name (int err)`

5.27.1 Macro Definition Documentation

5.27.1.1 `_`

```
#define _(  
    String ) dgettext (PACKAGE, String)
```

Definition at line 28 of file error.c.

5.27.1.2 ERR

```
#define ERR(  
    name,  
    desc ) { name, #name, desc }
```

Definition at line 32 of file error.c.

5.27.1.3 gettext_noop

```
#define gettext_noop(  
    String ) String
```

Definition at line 29 of file error.c.

5.27.1.4 N_

```
#define N_(  
    String ) gettext\_noop (String)
```

Definition at line 30 of file error.c.

5.27.2 Function Documentation

5.27.2.1 gssasl_strerror()

```
const char* gssasl_strerror (  
    int err )
```

gssasl_strerror:

Parameters

<i>err</i>	libgssasl error code
------------	----------------------

Convert return code to human readable string explanation of the reason for the particular error code.

This string can be used to output a diagnostic message to the user.

This function is one of few in the library that can be used without a successful call to [gssasl_init\(\)](#).

Return value: Returns a pointer to a statically allocated string containing an explanation of the error code @err.

Definition at line 185 of file error.c.

5.27.2.2 `gsasl_strerror_name()`

```
const char* gsasl_strerror_name (
    int err )
```

`gsasl_strerror_name`:

Parameters

<code>err</code>	libgsasl error code
------------------	---------------------

Convert return code to human readable string representing the error code symbol itself. For example, `gsasl_strerror_name(GSASL_OK)` returns the string "GSASL_OK".

This string can be used to output a diagnostic message to the user.

This function is one of few in the library that can be used without a successful call to [gsasl_init\(\)](#).

Return value: Returns a pointer to a statically allocated string containing a string version of the error code @err, or NULL if the error code is not known.

Since: 0.2.29

Definition at line 223 of file error.c.

5.27.3 Variable Documentation

5.27.3.1 description

```
const char* description
```

Definition at line 39 of file error.c.

5.27.3.2 name

```
const char* name
```

Definition at line 38 of file error.c.

5.27.3.3 rc

```
int rc
```

Definition at line 37 of file error.c.

5.28 external.h File Reference

```
#include <gsasl.h>
```

Macros

- #define [GSASL_EXTERNAL_NAME](#) "EXTERNAL"

Functions

- int [_gsasl_external_client_step](#) ([Gsasl_session](#) *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)
- int [_gsasl_external_server_step](#) ([Gsasl_session](#) *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)

Variables

- [Gsasl_mechanism_gsasl_external_mechanism](#)

5.28.1 Macro Definition Documentation

5.28.1.1 GSASL_EXTERNAL_NAME

```
#define GSASL_EXTERNAL_NAME "EXTERNAL"
```

Definition at line 28 of file external.h.

5.28.2 Function Documentation

5.28.2.1 _gsasl_external_client_step()

```
int _gsasl_external_client_step (  
    Gsasl\_session * sctx,  
    void * mech_data,  
    const char * input,  
    size_t input_len,  
    char ** output,  
    size_t * output_len )
```

5.28.2.2 `_gsasl_external_server_step()`

```
int _gsasl_external_server_step (
    Gsasl_session * sctx,
    void * mech_data,
    const char * input,
    size_t input_len,
    char ** output,
    size_t * output_len )
```

5.28.3 Variable Documentation

5.28.3.1 `_gsasl_external_mechanism`

`Gsasl_mechanism` `_gsasl_external_mechanism` [extern]

Definition at line 28 of file external/mechinfo.c.

5.29 free.c File Reference

```
#include <config.h>
#include "free.h"
#include <stdlib.h>
#include <string.h>
```

Functions

- void `digest_md5_free_challenge` (`digest_md5_challenge` *c)
- void `digest_md5_free_response` (`digest_md5_response` *r)
- void `digest_md5_free_finish` (`digest_md5_finish` *f)

5.29.1 Function Documentation

5.29.1.1 `digest_md5_free_challenge()`

```
void digest_md5_free_challenge (
    digest_md5_challenge * c )
```

Definition at line 35 of file digest-md5/free.c.

5.29.1.2 `digest_md5_free_finish()`

```
void digest_md5_free_finish (
    digest_md5_finish * f )
```

Definition at line 61 of file digest-md5/free.c.

5.29.1.3 `digest_md5_free_response()`

```
void digest_md5_free_response (
    digest_md5_response * r )
```

Definition at line 48 of file digest-md5/free.c.

5.30 `free.c` File Reference

```
#include <config.h>
#include "internal.h"
```

Functions

- void `gsasl_free` (void *ptr)

5.30.1 Function Documentation

5.30.1.1 `gsasl_free()`

```
void gsasl_free (
    void * ptr )
```

`gsasl_free`:

Parameters

<i>ptr</i>	memory pointer
------------	----------------

Invoke `free(@ptr)` to de-allocate memory pointer. Typically used on strings allocated by other libgsasl functions.

This is useful on Windows where libgsasl is linked to one CRT and the application is linked to another CRT. Then `malloc/free` will not use the same heap. This happens if you build libgsasl using mingw32 and the application with Visual Studio.

Since: 0.2.19

Definition at line 41 of file src/free.c.

5.31 free.h File Reference

```
#include "tokens.h"
```

Functions

- void [digest_md5_free_challenge](#) ([digest_md5_challenge](#) *c)
- void [digest_md5_free_response](#) ([digest_md5_response](#) *r)
- void [digest_md5_free_finish](#) ([digest_md5_finish](#) *f)

5.31.1 Function Documentation

5.31.1.1 [digest_md5_free_challenge\(\)](#)

```
void digest_md5_free_challenge (  
    digest\_md5\_challenge * c )
```

Definition at line 35 of file digest-md5/free.c.

5.31.1.2 [digest_md5_free_finish\(\)](#)

```
void digest_md5_free_finish (  
    digest\_md5\_finish * f )
```

Definition at line 61 of file digest-md5/free.c.

5.31.1.3 [digest_md5_free_response\(\)](#)

```
void digest_md5_free_response (  
    digest\_md5\_response * r )
```

Definition at line 48 of file digest-md5/free.c.

5.32 getsubopt.c File Reference

```
#include <config.h>
#include "parser.h"
#include <string.h>
```

Functions

- int [digest_md5_getsubopt](#) (char **optionp, const char *const *tokens, char **valuep)

5.32.1 Function Documentation

5.32.1.1 digest_md5_getsubopt()

```
int digest_md5_getsubopt (
    char ** optionp,
    const char *const * tokens,
    char ** valuep )
```

Definition at line 44 of file getsubopt.c.

5.33 gs2.h File Reference

```
#include <gsasl.h>
```

Macros

- #define [GSASL_GS2_KRB5_NAME](#) "GS2-KRB5"

Functions

- int [_gsasl_gs2_client_start](#) (Gsasl_session *sctx, void **mech_data)
- int [_gsasl_gs2_client_step](#) (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)
- void [_gsasl_gs2_client_finish](#) (Gsasl_session *sctx, void *mech_data)
- int [_gsasl_gs2_server_start](#) (Gsasl_session *sctx, void **mech_data)
- int [_gsasl_gs2_server_step](#) (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)
- void [_gsasl_gs2_server_finish](#) (Gsasl_session *sctx, void *mech_data)

Variables

- [Gsasl_mechanism_gsasl_gs2_krb5_mechanism](#)

5.33.1 Macro Definition Documentation

5.33.1.1 GSASL_GS2_KRB5_NAME

```
#define GSASL_GS2_KRB5_NAME "GS2-KRB5"
```

Definition at line 28 of file gs2.h.

5.33.2 Function Documentation

5.33.2.1 __gsasl_gs2_client_finish()

```
void __gsasl_gs2_client_finish (  
    Gsasl_session * sctx,  
    void * mech_data )
```

Definition at line 316 of file gs2/client.c.

5.33.2.2 __gsasl_gs2_client_start()

```
int __gsasl_gs2_client_start (  
    Gsasl_session * sctx,  
    void ** mech_data )
```

Definition at line 52 of file gs2/client.c.

5.33.2.3 __gsasl_gs2_client_step()

```
int __gsasl_gs2_client_step (  
    Gsasl_session * sctx,  
    void * mech_data,  
    const char * input,  
    size_t input_len,  
    char ** output,  
    size_t * output_len )
```

Definition at line 235 of file gs2/client.c.

5.33.2.4 `_gsasl_gs2_server_finish()`

```
void _gsasl_gs2_server_finish (
    Gsasl_session * sctx,
    void * mech_data )
```

Definition at line 299 of file gs2/server.c.

5.33.2.5 `_gsasl_gs2_server_start()`

```
int _gsasl_gs2_server_start (
    Gsasl_session * sctx,
    void ** mech_data )
```

Definition at line 119 of file gs2/server.c.

5.33.2.6 `_gsasl_gs2_server_step()`

```
int _gsasl_gs2_server_step (
    Gsasl_session * sctx,
    void * mech_data,
    const char * input,
    size_t input_len,
    char ** output,
    size_t * output_len )
```

Definition at line 162 of file gs2/server.c.

5.33.3 Variable Documentation

5.33.3.1 `_gsasl_gs2_krb5_mechanism`

```
Gsasl_mechanism _gsasl_gs2_krb5_mechanism [extern]
```

Definition at line 28 of file gs2/mechinfo.c.

5.34 gs2helper.c File Reference

```
#include <config.h>
#include <string.h>
#include <stdlib.h>
#include "gs2helper.h"
```

Functions

- int [gs2_get_oid](#) ([Gsasl_session](#) *sctx, gss_OID *mech_oid)

5.34.1 Function Documentation

5.34.1.1 gs2_get_oid()

```
int gs2_get_oid (
    Gsasl\_session * sctx,
    gss_OID * mech_oid )
```

Definition at line 38 of file gs2helper.c.

5.35 gs2helper.h File Reference

```
#include "gss-extra.h"
#include <gsasl.h>
```

Functions

- int [gs2_get_oid](#) ([Gsasl_session](#) *sctx, gss_OID *mech_oid)

5.35.1 Function Documentation

5.35.1.1 gs2_get_oid()

```
int gs2_get_oid (
    Gsasl\_session * sctx,
    gss_OID * mech_oid )
```

Definition at line 38 of file gs2helper.c.

5.36 gsasl-mech.h File Reference

Data Structures

- struct [Gsasl_mechanism_functions](#)
- struct [Gsasl_mechanism](#)

Typedefs

- typedef int(* [Gsasl_init_function](#)) (Gsasl *ctx)
- typedef void(* [Gsasl_done_function](#)) (Gsasl *ctx)
- typedef int(* [Gsasl_start_function](#)) (Gsasl_session *sctx, void **mech_data)
- typedef int(* [Gsasl_step_function](#)) (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)
- typedef void(* [Gsasl_finish_function](#)) (Gsasl_session *sctx, void *mech_data)
- typedef int(* [Gsasl_code_function](#)) (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)
- typedef struct [Gsasl_mechanism_functions](#) Gsasl_mechanism_functions
- typedef struct [Gsasl_mechanism](#) Gsasl_mechanism

Functions

- [_GSASL_API](#) int [gsasl_register](#) (Gsasl *ctx, const [Gsasl_mechanism](#) *mech)

5.36.1 Typedef Documentation

5.36.1.1 Gsasl_code_function

```
typedef int(* Gsasl_code_function) (Gsasl_session *sctx, void *mech_data, const char *input,
size_t input_len, char **output, size_t *output_len)
```

Gsasl_code_function:

Parameters

<i>sctx</i>	a Gsasl_session session handle.
<i>mech_data</i>	pointer to void* with mechanism-specific data.
<i>input</i>	input byte array.
<i>input_len</i>	size of input byte array.
<i>output</i>	newly allocated output byte array.
<i>output_len</i>	pointer to output variable with size of output byte array.

The implementation of this function should perform data encoding or decoding for the mechanism, after authentication has completed. This might mean that data is integrity or privacy protected.

The @output buffer is allocated by this function, and it is the responsibility of caller to deallocate it by calling [gsasl_free\(@output\)](#).

Return value: Returns GSASL_OK if encoding was successful, otherwise an error code.

Definition at line 134 of file [gsasl-mech.h](#).

5.36.1.2 Gssasl_done_function

```
typedef void(* Gssasl_done_function) (Gssasl *ctx)
```

Gssasl_done_function:

Parameters

<i>ctx</i>	a Gssasl libgssasl handle.
------------	----------------------------

The implementation of this function pointer deallocate all resources associated with the mechanism.

Definition at line 58 of file gssasl-mech.h.

5.36.1.3 Gssasl_finish_function

```
typedef void(* Gssasl_finish_function) (Gssasl_session *sctx, void *mech_data)
```

Gssasl_finish_function:

Parameters

<i>sctx</i>	a Gssasl_session session handle.
<i>mech_data</i>	pointer to void* with mechanism-specific data.

The implementation of this function should release all resources associated with the particular authentication process.

Definition at line 112 of file gssasl-mech.h.

5.36.1.4 Gssasl_init_function

```
typedef int(* Gssasl_init_function) (Gssasl *ctx)
```

SECTION:gssasl-mech

Parameters

<i>title</i>	gssasl-mech.h
<i>short_description</i>	register new application-defined mechanism

The builtin mechanisms should suffice for most applications. Applications can register a new mechanism in the library using application-supplied functions. The mechanism will operate as the builtin mechanisms, and the supplied functions will be invoked when necessary. The application uses the normal logic, e.g., calls [gssasl_client_start\(\)](#) followed by a sequence of calls to [gssasl_step\(\)](#) and finally [gssasl_finish\(\)](#). Gssasl_init_function:

Parameters

<i>ctx</i>	a Gsasl libgsasl handle.
------------	--------------------------

The implementation of this function pointer should fail if the mechanism for some reason is not available for further use.

Return value: Returns GSASL_OK iff successful.

Definition at line 49 of file gsasl-mech.h.

5.36.1.5 Gsasl_mechanism

```
typedef struct Gsasl_mechanism Gsasl_mechanism
```

Definition at line 134 of file gsasl-mech.h.

5.36.1.6 Gsasl_mechanism_functions

```
typedef struct Gsasl_mechanism_functions Gsasl_mechanism_functions
```

Definition at line 134 of file gsasl-mech.h.

5.36.1.7 Gsasl_start_function

```
typedef int(* Gsasl_start_function) (Gsasl_session *sctx, void **mech_data)
```

Gsasl_start_function:

Parameters

<i>sctx</i>	a Gsasl_session session handle.
<i>mech_data</i>	pointer to void* with mechanism-specific data.

The implementation of this function should start a new authentication process.

Return value: Returns GSASL_OK iff successful.

Definition at line 70 of file gsasl-mech.h.

5.36.1.8 Gssapi_step_function

```
typedef int (* Gssapi_step_function) (Gssapi_session *sctx, void *mech_data, const char *input,
size_t input_len, char **output, size_t *output_len)
```

Gssapi_step_function:

Parameters

<i>sctx</i>	a Gssapi_session session handle.
<i>mech_data</i>	pointer to void* with mechanism-specific data.
<i>input</i>	input byte array.
<i>input_len</i>	size of input byte array.
<i>output</i>	newly allocated output byte array.
<i>output_len</i>	pointer to output variable with size of output byte array.

The implementation of this function should perform one step of the authentication process.

This reads data from the other end (from @input and @input_len), processes it (potentially invoking callbacks to the application), and writes data to server (into newly allocated variable @output and @output_len that indicate the length of @output).

The contents of the @output buffer is unspecified if this functions returns anything other than GSASL_OK or GSASL_NEEDS_MORE. If this function return GSASL_OK or GSASL_NEEDS_MORE, however, the @output buffer is allocated by this function, and it is the responsibility of caller to deallocate it by calling gssapi_free(@output).

Return value: Returns GSASL_OK if authenticated terminated successfully, GSASL_NEEDS_MORE if more data is needed, or error code.

Definition at line 100 of file gssapi-mech.h.

5.36.2 Function Documentation

5.36.2.1 gssapi_register()

```
_GSASL_API int gssapi_register (
    Gssapi * ctx,
    const Gssapi_mechanism * mech )
```

gssapi_register:

Parameters

<i>ctx</i>	pointer to libgssapi handle.
<i>mech</i>	plugin structure with information about plugin.

This function initialize given mechanism, and if successful, add it to the list of plugins that is used by the library.

Return value: GSASL_OK iff successful, otherwise GSASL_MALLOC_ERROR.

Since: 0.2.0

Definition at line 39 of file register.c.

5.37 gsasl-version.h File Reference

Macros

- #define [GSASL_VERSION](#) "2.2.1"
- #define [GSASL_VERSION_MAJOR](#) 2
- #define [GSASL_VERSION_MINOR](#) 2
- #define [GSASL_VERSION_PATCH](#) 1
- #define [GSASL_VERSION_NUMBER](#) 0x020201

5.37.1 Macro Definition Documentation

5.37.1.1 GSASL_VERSION

```
#define GSASL_VERSION "2.2.1"
```

SECTION:gsasl-version

Parameters

<i>title</i>	gsasl-version.h
<i>short_description</i>	version symbols

The [gsasl-version.h](#) file contains version symbols. It should not be included directly, only via [gsasl.h](#). GSASL_↔
VERSION

Pre-processor symbol with a string that describe the header file version number. Used together with [gsasl_check_version\(\)](#) to verify header file and run-time library consistency.

Definition at line 42 of file gsasl-version.h.

5.37.1.2 GSASL_VERSION_MAJOR

```
#define GSASL_VERSION_MAJOR 2
```

GSASL_VERSION_MAJOR

Pre-processor symbol with a decimal value that describe the major level of the header file version number. For example, when the header version is 1.2.3 this symbol will be 1.

Since: 1.1

Definition at line 53 of file gsasl-version.h.

5.37.1.3 GSASL_VERSION_MINOR

```
#define GSASL_VERSION_MINOR 2
```

GSASL_VERSION_MINOR

Pre-processor symbol with a decimal value that describe the minor level of the header file version number. For example, when the header version is 1.2.3 this symbol will be 2.

Since: 1.1

Definition at line 64 of file gsasl-version.h.

5.37.1.4 GSASL_VERSION_NUMBER

```
#define GSASL_VERSION_NUMBER 0x020201
```

GSASL_VERSION_NUMBER

Pre-processor symbol with a hexadecimal value describing the header file version number. For example, when the header version is 1.2.3 this symbol will have the value 0x010203.

Since: 1.1

Definition at line 86 of file gsasl-version.h.

5.37.1.5 GSASL_VERSION_PATCH

```
#define GSASL_VERSION_PATCH 1
```

GSASL_VERSION_PATCH

Pre-processor symbol with a decimal value that describe the patch level of the header file version number. For example, when the header version is 1.2.3 this symbol will be 3.

Since: 1.1

Definition at line 75 of file gsasl-version.h.

5.38 gssapi.h File Reference

```
#include <stdio.h>
#include <stddef.h>
#include <unistd.h>
#include <gssapi-version.h>
#include <gssapi-mech.h>
```

Macros

- `#define _GSSAPI_API`

Typedefs

- typedef struct `Gssapi Gssapi`
- typedef struct `Gssapi_session Gssapi_session`
- typedef int(* `Gssapi_callback_function`) (`Gssapi *ctx`, `Gssapi_session *sctx`, `Gssapi_property prop`)

Enumerations

- enum `Gssapi_rc` {
 - `GSSAPI_OK` = 0, `GSSAPI_NEEDS_MORE` = 1, `GSSAPI_UNKNOWN_MECHANISM` = 2, `GSSAPI_MECHANISM_CALLED_TOO_MANY_TIMES` = 3,
 - `GSSAPI_MALLOC_ERROR` = 7, `GSSAPI_BASE64_ERROR` = 8, `GSSAPI_CRYPT_ERROR` = 9,
 - `GSSAPI_SASLPREP_ERROR` = 29,
 - `GSSAPI_MECHANISM_PARSE_ERROR` = 30, `GSSAPI_AUTHENTICATION_ERROR` = 31, `GSSAPI_INTEGRITY_ERROR` = 33, `GSSAPI_NO_CLIENT_CODE` = 35,
 - `GSSAPI_NO_SERVER_CODE` = 36, `GSSAPI_NO_CALLBACK` = 51, `GSSAPI_NO_ANONYMOUS_TOKEN` = 52, `GSSAPI_NO_AUTHID` = 53,
 - `GSSAPI_NO_AUTHZID` = 54, `GSSAPI_NO_PASSWORD` = 55, `GSSAPI_NO_PASSCODE` = 56,
 - `GSSAPI_NO_PIN` = 57,
 - `GSSAPI_NO_SERVICE` = 58, `GSSAPI_NO_HOSTNAME` = 59, `GSSAPI_NO_CB_TLS_UNIQUE` = 65,
 - `GSSAPI_NO_SAML20_IDP_IDENTIFIER` = 66,
 - `GSSAPI_NO_SAML20_REDIRECT_URL` = 67, `GSSAPI_NO_OPENID20_REDIRECT_URL` = 68,
 - `GSSAPI_NO_CB_TLS_EXPORTER` = 69, `GSSAPI_GSSAPI_RELEASE_BUFFER_ERROR` = 37,
 - `GSSAPI_GSSAPI_IMPORT_NAME_ERROR` = 38, `GSSAPI_GSSAPI_INIT_SEC_CONTEXT_ERROR` = 39,
 - `GSSAPI_GSSAPI_ACCEPT_SEC_CONTEXT_ERROR` = 40, `GSSAPI_GSSAPI_UNWRAP_ERROR` = 41,
 - `GSSAPI_GSSAPI_WRAP_ERROR` = 42, `GSSAPI_GSSAPI_ACQUIRE_CRED_ERROR` = 43, `GSSAPI_GSSAPI_DISPLAY_NAME_ERROR` = 44, `GSSAPI_GSSAPI_UNSUPPORTED_PROTECTION_ERROR` = 45,
 - `GSSAPI_SECURID_SERVER_NEED_ADDITIONAL_PASSCODE` = 48, `GSSAPI_SECURID_SERVER_NEED_NEW_PIN` = 49, `GSSAPI_GSSAPI_ENCAPSULATE_TOKEN_ERROR` = 60, `GSSAPI_GSSAPI_DECAPSULATE_TOKEN_ERROR` = 61,
 - `GSSAPI_GSSAPI_INQUIRE_MECH_FOR_SASLNAME_ERROR` = 62, `GSSAPI_GSSAPI_TEST_OID_SET_MEMBER_ERROR` = 63, `GSSAPI_GSSAPI_RELEASE_OID_SET_ERROR` = 64 }
- enum `Gssapi_property` {
 - `GSSAPI_AUTHID` = 1, `GSSAPI_AUTHZID` = 2, `GSSAPI_PASSWORD` = 3, `GSSAPI_ANONYMOUS_TOKEN` = 4,
 - `GSSAPI_SERVICE` = 5, `GSSAPI_HOSTNAME` = 6, `GSSAPI_GSSAPI_DISPLAY_NAME` = 7,
 - `GSSAPI_PASSCODE` = 8,
 - `GSSAPI_SUGGESTED_PIN` = 9, `GSSAPI_PIN` = 10, `GSSAPI_REALM` = 11, `GSSAPI_DIGEST_MD5_HASHED_PASSWORD` = 12,
 - `GSSAPI_QOPS` = 13, `GSSAPI_QOP` = 14, `GSSAPI_SCRAM_ITER` = 15, `GSSAPI_SCRAM_SALT` = 16,

- ```

GSASL_SCRAM_SALTED_PASSWORD = 17 , GSASL_SCRAM_SERVERKEY = 23 , GSASL_SCRAM_STOREDKEY
= 24 , GSASL_CB_TLS_UNIQUE = 18 ,
GSASL_SAML20_IDP_IDENTIFIER = 19 , GSASL_SAML20_REDIRECT_URL = 20 , GSASL_OPENID20_REDIRECT_URL
= 21 , GSASL_OPENID20_OUTCOME_DATA = 22 ,
GSASL_CB_TLS_EXPORTER = 25 , GSASL_SAML20_AUTHENTICATE_IN_BROWSER = 250 ,
GSASL_OPENID20_AUTHENTICATE_IN_BROWSER = 251 , GSASL_VALIDATE_SIMPLE = 500 ,
GSASL_VALIDATE_EXTERNAL = 501 , GSASL_VALIDATE_ANONYMOUS = 502 , GSASL_VALIDATE_GSSAPI
= 503 , GSASL_VALIDATE_SECURID = 504 ,
GSASL_VALIDATE_SAML20 = 505 , GSASL_VALIDATE_OPENID20 = 506 }
• enum Gsasl_mechname_limits { GSASL_MIN_MECHANISM_SIZE = 1 , GSASL_MAX_MECHANISM_SIZE
= 20 }
• enum Gsasl_qop { GSASL_QOP_AUTH = 1 , GSASL_QOP_AUTH_INT = 2 , GSASL_QOP_AUTH_CONF =
4 }
• enum Gsasl_saslprep_flags { GSASL_ALLOW_UNASSIGNED = 1 }
• enum Gsasl_hash { GSASL_HASH_SHA1 = 2 , GSASL_HASH_SHA256 = 3 }
• enum Gsasl_hash_length { GSASL_HASH_SHA1_SIZE = 20 , GSASL_HASH_SHA256_SIZE = 32 ,
GSASL_HASH_MAX_SIZE = GSASL_HASH_SHA256_SIZE }

```

## Functions

- `_GSASL_API int gssapi_init (Gssapi **ctx)`
- `_GSASL_API void gssapi_done (Gssapi *ctx)`
- `_GSASL_API const char * gssapi_check_version (const char *req_version)`
- `_GSASL_API void gssapi_callback_set (Gssapi *ctx, Gssapi_callback_function cb)`
- `_GSASL_API int gssapi_callback (Gssapi *ctx, Gssapi_session *sctx, Gssapi_property prop)`
- `_GSASL_API void gssapi_callback_hook_set (Gssapi *ctx, void *hook)`
- `_GSASL_API void * gssapi_callback_hook_get (Gssapi *ctx)`
- `_GSASL_API void gssapi_session_hook_set (Gssapi_session *sctx, void *hook)`
- `_GSASL_API void * gssapi_session_hook_get (Gssapi_session *sctx)`
- `_GSASL_API int gssapi_property_set (Gssapi_session *sctx, Gssapi_property prop, const char *data)`
- `_GSASL_API int gssapi_property_set_raw (Gssapi_session *sctx, Gssapi_property prop, const char *data, size_t len)`
- `_GSASL_API void gssapi_property_free (Gssapi_session *sctx, Gssapi_property prop)`
- `_GSASL_API const char * gssapi_property_get (Gssapi_session *sctx, Gssapi_property prop)`
- `_GSASL_API const char * gssapi_property_fast (Gssapi_session *sctx, Gssapi_property prop)`
- `_GSASL_API int gssapi_client_mechlist (Gssapi *ctx, char **out)`
- `_GSASL_API int gssapi_client_support_p (Gssapi *ctx, const char *name)`
- `_GSASL_API const char * gssapi_client_suggest_mechanism (Gssapi *ctx, const char *mechlist)`
- `_GSASL_API int gssapi_server_mechlist (Gssapi *ctx, char **out)`
- `_GSASL_API int gssapi_server_support_p (Gssapi *ctx, const char *name)`
- `_GSASL_API int gssapi_mechanism_name_p (const char *mech)`
- `_GSASL_API int gssapi_client_start (Gssapi *ctx, const char *mech, Gssapi_session **sctx)`
- `_GSASL_API int gssapi_server_start (Gssapi *ctx, const char *mech, Gssapi_session **sctx)`
- `_GSASL_API int gssapi_step (Gssapi_session *sctx, const char *input, size_t input_len, char **output, size_t *output_len)`
- `_GSASL_API int gssapi_step64 (Gssapi_session *sctx, const char *b64input, char **b64output)`
- `_GSASL_API void gssapi_finish (Gssapi_session *sctx)`
- `_GSASL_API int gssapi_encode (Gssapi_session *sctx, const char *input, size_t input_len, char **output, size_t *output_len)`
- `_GSASL_API int gssapi_decode (Gssapi_session *sctx, const char *input, size_t input_len, char **output, size_t *output_len)`
- `_GSASL_API const char * gssapi_mechanism_name (Gssapi_session *sctx)`
- `_GSASL_API const char * gssapi_strerror (int err)`
- `_GSASL_API const char * gssapi_strerror_name (int err)`

- `_GSASL_API` int `gsasl_saslprep` (const char \*in, `Gsasl_saslprep_flags` flags, char \*\*out, int \*stringpreproc)
- `_GSASL_API` int `gsasl_nonce` (char \*data, size\_t datalen)
- `_GSASL_API` int `gsasl_random` (char \*data, size\_t datalen)
- `_GSASL_API` size\_t `gsasl_hash_length` (`Gsasl_hash` hash)
- `_GSASL_API` int `gsasl_scram_secrets_from_salted_password` (`Gsasl_hash` hash, const char \*salted\_password, char \*client\_key, char \*server\_key, char \*stored\_key)
- `_GSASL_API` int `gsasl_scram_secrets_from_password` (`Gsasl_hash` hash, const char \*password, unsigned int iteration\_count, const char \*salt, size\_t saltlen, char \*salted\_password, char \*client\_key, char \*server\_key, char \*stored\_key)
- `_GSASL_API` int `gsasl_simple_getpass` (const char \*filename, const char \*username, char \*\*key)
- `_GSASL_API` int `gsasl_base64_to` (const char \*in, size\_t inlen, char \*\*out, size\_t \*outlen)
- `_GSASL_API` int `gsasl_base64_from` (const char \*in, size\_t inlen, char \*\*out, size\_t \*outlen)
- `_GSASL_API` int `gsasl_hex_to` (const char \*in, size\_t inlen, char \*\*out, size\_t \*outlen)
- `_GSASL_API` int `gsasl_hex_from` (const char \*in, char \*\*out, size\_t \*outlen)
- `_GSASL_API` void `gsasl_free` (void \*ptr)

## 5.38.1 Macro Definition Documentation

### 5.38.1.1 `_GSASL_API`

```
#define _GSASL_API
```

SECTION:gsasl

Parameters

|                          |                         |
|--------------------------|-------------------------|
| <i>title</i>             | <a href="#">gsasl.h</a> |
| <i>short_description</i> | main library interfaces |

The main library interfaces are declared in [gsasl.h](#).

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

## 5.38.2 Typedef Documentation

### 5.38.2.1 `Gsasl`

```
typedef struct Gsasl Gsasl
```

[Gsasl](#):

Handle to global library context.

Definition at line 1 of file `gsasl.h`.

### 5.38.2.2 Gssapi\_callback\_function

```
typedef int (* Gssapi_callback_function) (Gssapi *ctx, Gssapi_session *sctx, Gssapi_property prop)
```

Gssapi\_callback\_function:

#### Parameters

|             |                                           |
|-------------|-------------------------------------------|
| <i>ctx</i>  | libgssapi handle.                         |
| <i>sctx</i> | session handle, may be NULL.              |
| <i>prop</i> | enumerated value of Gssapi_property type. |

Prototype of function that the application should implement. Use [gssapi\\_callback\\_set\(\)](#) to inform the library about your callback function.

It is called by the SASL library when it need some information from the application. Depending on the value of @prop, it should either set some property (e.g., username or password) using [gssapi\\_property\\_set\(\)](#), or it should extract some properties (e.g., authentication and authorization identities) using [gssapi\\_property\\_fast\(\)](#) and use them to make a policy decision, perhaps returning GSASL\_AUTHENTICATION\_ERROR or GSASL\_OK depending on whether the policy permitted the operation.

Return value: Any valid return code, the interpretation of which depend on the @prop value.

Since: 0.2.0

Definition at line 286 of file gssapi.h.

### 5.38.2.3 Gssapi\_session

```
typedef struct Gssapi_session Gssapi_session
```

[Gssapi\\_session](#):

Handle to SASL session context.

Definition at line 1 of file gssapi.h.

## 5.38.3 Enumeration Type Documentation

### 5.38.3.1 Gssapi\_hash

```
enum Gssapi_hash
```

Gssapi\_hash:

## Parameters

|                          |                        |
|--------------------------|------------------------|
| <i>GSASL_HASH_SHA1</i>   | Hash function SHA-1.   |
| <i>GSASL_HASH_SHA256</i> | Hash function SHA-256. |

Hash functions. You may use [gsasl\\_hash\\_length\(\)](#) to get the output size of a hash function.

Currently only used as parameter to [gsasl\\_scram\\_secrets\\_from\\_salted\\_password\(\)](#) and [gsasl\\_scram\\_secrets\\_from\\_password\(\)](#) to specify for which SCRAM mechanism to prepare secrets for.

Since: 1.10

## Enumerator

|                          |  |
|--------------------------|--|
| <i>GSASL_HASH_SHA1</i>   |  |
| <i>GSASL_HASH_SHA256</i> |  |

Definition at line 427 of file `gsasl.h`.

### 5.38.3.2 Gsasl\_hash\_length

enum [Gsasl\\_hash\\_length](#)

Gsasl\_hash\_length:

## Parameters

|                               |                                               |
|-------------------------------|-----------------------------------------------|
| <i>GSASL_HASH_SHA1_SIZE</i>   | Output size of hash function SHA-1.           |
| <i>GSASL_HASH_SHA256_SIZE</i> | Output size of hash function SHA-256.         |
| <i>GSASL_HASH_MAX_SIZE</i>    | Maximum output size of any Gsasl_hash_length. |

Identifiers specifying the output size of hash functions.

These can be used when statically allocating the buffers needed for, e.g., [gsasl\\_scram\\_secrets\\_from\\_password\(\)](#).

Since: 1.10

## Enumerator

|                               |  |
|-------------------------------|--|
| <i>GSASL_HASH_SHA1_SIZE</i>   |  |
| <i>GSASL_HASH_SHA256_SIZE</i> |  |
| <i>GSASL_HASH_MAX_SIZE</i>    |  |

Definition at line 447 of file `gsasl.h`.



### 5.38.3.3 Gsasl\_mechname\_limits

enum [Gsasl\\_mechname\\_limits](#)

Gsasl\_mechname\_limits:

#### Parameters

|                                 |                                         |
|---------------------------------|-----------------------------------------|
| <i>GSASL_MIN_MECHANISM_SIZE</i> | Minimum size of mechanism name strings. |
| <i>GSASL_MAX_MECHANISM_SIZE</i> | Maximum size of mechanism name strings. |

SASL mechanisms are named by strings, from 1 to 20 characters in length, consisting of upper-case letters, digits, hyphens, and/or underscores. See also [gsasl\\_mechanism\\_name\\_p\(\)](#).

#### Enumerator

|                                 |  |
|---------------------------------|--|
| <i>GSASL_MIN_MECHANISM_SIZE</i> |  |
| <i>GSASL_MAX_MECHANISM_SIZE</i> |  |

Definition at line 298 of file gssapi.h.

### 5.38.3.4 Gsasl\_property

enum [Gsasl\\_property](#)

Gsasl\_property:

#### Parameters

|                                         |                                                                                   |
|-----------------------------------------|-----------------------------------------------------------------------------------|
| <i>GSASL_AUTHID</i>                     | Authentication identity (username).                                               |
| <i>GSASL_AUTHZID</i>                    | Authorization identity.                                                           |
| <i>GSASL_PASSWORD</i>                   | Password.                                                                         |
| <i>GSASL_ANONYMOUS_TOKEN</i>            | Anonymous identifier.                                                             |
| <i>GSASL_SERVICE</i>                    | Service name                                                                      |
| <i>GSASL_HOSTNAME</i>                   | Host name.                                                                        |
| <i>GSASL_GSSAPI_DISPLAY_NAME</i>        | GSS-API credential principal name.                                                |
| <i>GSASL_PASSCODE</i>                   | SecurID passcode.                                                                 |
| <i>GSASL_SUGGESTED_PIN</i>              | SecurID suggested PIN.                                                            |
| <i>GSASL_PIN</i>                        | SecurID PIN.                                                                      |
| <i>GSASL_REALM</i>                      | User realm.                                                                       |
| <i>GSASL_DIGEST_MD5_HASHED_PASSWORD</i> | Pre-computed hashed DIGEST-MD5 password, to avoid storing passwords in the clear. |
| <i>GSASL_QOPS</i>                       | Set of quality-of-protection values.                                              |
| <i>GSASL_QOP</i>                        | Quality-of-protection value.                                                      |
| <i>GSASL_SCRAM_ITER</i>                 | Number of iterations in password-to-key hashing.                                  |
| <i>GSASL_SCRAM_SALT</i>                 | Salt for password-to-key hashing.                                                 |
| <i>GSASL_SCRAM_SALTED_PASSWORD</i>      | Hex-encoded hashed/salted password.                                               |

## Parameters

|                                               |                                                           |
|-----------------------------------------------|-----------------------------------------------------------|
| <i>GSASL_SCRAM_SERVERKEY</i>                  | Hex-encoded SCRAM ServerKey derived from users' password. |
| <i>GSASL_SCRAM_STOREDKEY</i>                  | Hex-encoded SCRAM StoredKey derived from users' password. |
| <i>GSASL_CB_TLS_UNIQUE</i>                    | Base64 encoded tls-unique channel binding.                |
| <i>GSASL_CB_TLS_EXPORTER</i>                  | Base64 encoded tls-exporter channel binding.              |
| <i>GSASL_SAML20_IDP_IDENTIFIER</i>            | SAML20 user IdP URL.                                      |
| <i>GSASL_SAML20_REDIRECT_URL</i>              | SAML 2.0 URL to access in browser.                        |
| <i>GSASL_OPENID20_REDIRECT_URL</i>            | OpenID 2.0 URL to access in browser.                      |
| <i>GSASL_OPENID20_OUTCOME_DATA</i>            | OpenID 2.0 authentication outcome data.                   |
| <i>GSASL_SAML20_AUTHENTICATE_IN_BROWSER</i>   | Request to perform SAML 2.0 authentication in browser.    |
| <i>GSASL_OPENID20_AUTHENTICATE_IN_BROWSER</i> | Request to perform OpenID 2.0 authentication in browser.  |
| <i>GSASL_VALIDATE_SIMPLE</i>                  | Request for simple validation.                            |
| <i>GSASL_VALIDATE_EXTERNAL</i>                | Request for validation of EXTERNAL.                       |
| <i>GSASL_VALIDATE_ANONYMOUS</i>               | Request for validation of ANONYMOUS.                      |
| <i>GSASL_VALIDATE_GSSAPI</i>                  | Request for validation of GSSAPI/GS2.                     |
| <i>GSASL_VALIDATE_SECURID</i>                 | Request for validation of SecurID.                        |
| <i>GSASL_VALIDATE_SAML20</i>                  | Request for validation of SAML20.                         |
| <i>GSASL_VALIDATE_OPENID20</i>                | Request for validation of OpenID 2.0 login.               |

Callback/property types.

## Enumerator

|                                  |  |
|----------------------------------|--|
| GSASL_AUTHID                     |  |
| GSASL_AUTHZID                    |  |
| GSASL_PASSWORD                   |  |
| GSASL_ANONYMOUS_TOKEN            |  |
| GSASL_SERVICE                    |  |
| GSASL_HOSTNAME                   |  |
| GSASL_GSSAPI_DISPLAY_NAME        |  |
| GSASL_PASSCODE                   |  |
| GSASL_SUGGESTED_PIN              |  |
| GSASL_PIN                        |  |
| GSASL_REALM                      |  |
| GSASL_DIGEST_MD5_HASHED_PASSWORD |  |
| GSASL_QOPS                       |  |
| GSASL_QOP                        |  |
| GSASL_SCRAM_ITER                 |  |
| GSASL_SCRAM_SALT                 |  |
| GSASL_SCRAM_SALTED_PASSWORD      |  |
| GSASL_SCRAM_SERVERKEY            |  |
| GSASL_SCRAM_STOREDKEY            |  |
| GSASL_CB_TLS_UNIQUE              |  |
| GSASL_SAML20_IDP_IDENTIFIER      |  |
| GSASL_SAML20_REDIRECT_URL        |  |
| GSASL_OPENID20_REDIRECT_URL      |  |

## Enumerator

|                                        |  |
|----------------------------------------|--|
| GSASL_OPENID20_OUTCOME_DATA            |  |
| GSASL_CB_TLS_EXPORTER                  |  |
| GSASL_SAML20_AUTHENTICATE_IN_BROWSER   |  |
| GSASL_OPENID20_AUTHENTICATE_IN_BROWSER |  |
| GSASL_VALIDATE_SIMPLE                  |  |
| GSASL_VALIDATE_EXTERNAL                |  |
| GSASL_VALIDATE_ANONYMOUS               |  |
| GSASL_VALIDATE_GSSAPI                  |  |
| GSASL_VALIDATE_SECURID                 |  |
| GSASL_VALIDATE_SAML20                  |  |
| GSASL_VALIDATE_OPENID20                |  |

Definition at line 221 of file gssapi.h.

## 5.38.3.5 Gssapi\_qop

enum [Gssapi\\_qop](#)

Gssapi\_qop:

## Parameters

|                            |                                                |
|----------------------------|------------------------------------------------|
| <i>GSASL_QOP_AUTH</i>      | Authentication only.                           |
| <i>GSASL_QOP_AUTH_INT</i>  | Authentication and integrity.                  |
| <i>GSASL_QOP_AUTH_CONF</i> | Authentication, integrity and confidentiality. |

Quality of Protection types (DIGEST-MD5 and GSSAPI). The integrity and confidentiality values is about application data wrapping. We recommend that you use `@GSASL_QOP_AUTH` with TLS as that combination is generally more secure and have better chance of working than the integrity/confidentiality layers of SASL.

## Enumerator

|                     |  |
|---------------------|--|
| GSASL_QOP_AUTH      |  |
| GSASL_QOP_AUTH_INT  |  |
| GSASL_QOP_AUTH_CONF |  |

Definition at line 316 of file gssapi.h.

## 5.38.3.6 Gssapi\_rc

enum [Gssapi\\_rc](#)

Gssapi\_rc:

## Parameters

|                                                     |                                                            |
|-----------------------------------------------------|------------------------------------------------------------|
| <i>GSASL_OK</i>                                     | Successful return code, guaranteed to be always 0.         |
| <i>GSASL_NEEDS_MORE</i>                             | Mechanism expects another round-trip.                      |
| <i>GSASL_UNKNOWN_MECHANISM</i>                      | Application requested an unknown mechanism.                |
| <i>GSASL_MECHANISM_CALLED_TOO_MANY_TIMES</i>        | Application requested too many round trips from mechanism. |
| <i>GSASL_MALLOC_ERROR</i>                           | Memory allocation failed.                                  |
| <i>GSASL_BASE64_ERROR</i>                           | Base64 encoding/decoding failed.                           |
| <i>GSASL_CRYPT_ERROR</i>                            | Cryptographic error.                                       |
| <i>GSASL_SASLPREP_ERROR</i>                         | Failed to prepare internationalized string.                |
| <i>GSASL_MECHANISM_PARSE_ERROR</i>                  | Mechanism could not parse input.                           |
| <i>GSASL_AUTHENTICATION_ERROR</i>                   | Authentication has failed.                                 |
| <i>GSASL_INTEGRITY_ERROR</i>                        | Application data integrity check failed.                   |
| <i>GSASL_NO_CLIENT_CODE</i>                         | Library was built with client functionality.               |
| <i>GSASL_NO_SERVER_CODE</i>                         | Library was built with server functionality.               |
| <i>GSASL_NO_CALLBACK</i>                            | Application did not provide a callback.                    |
| <i>GSASL_NO_ANONYMOUS_TOKEN</i>                     | Could not get required anonymous token.                    |
| <i>GSASL_NO_AUTHID</i>                              | Could not get required authentication identity (username). |
| <i>GSASL_NO_AUTHZID</i>                             | Could not get required authorization identity.             |
| <i>GSASL_NO_PASSWORD</i>                            | Could not get required password.                           |
| <i>GSASL_NO_PASSCODE</i>                            | Could not get required SecurID PIN.                        |
| <i>GSASL_NO_PIN</i>                                 | Could not get required SecurID PIN.                        |
| <i>GSASL_NO_SERVICE</i>                             | Could not get required service name.                       |
| <i>GSASL_NO_HOSTNAME</i>                            | Could not get required hostname.                           |
| <i>GSASL_NO_CB_TLS_UNIQUE</i>                       | Could not get required tls-unique CB.                      |
| <i>GSASL_NO_CB_TLS_EXPORTER</i>                     | Could not get required tls-exporter CB.                    |
| <i>GSASL_NO_SAML20_IDP_IDENTIFIER</i>               | Could not get required SAML IdP.                           |
| <i>GSASL_NO_SAML20_REDIRECT_URL</i>                 | Could not get required SAML redirect URL.                  |
| <i>GSASL_NO_OPENID20_REDIRECT_URL</i>               | Could not get required OpenID redirect URL.                |
| <i>GSASL_GSSAPI_RELEASE_BUFFER_ERROR</i>            | GSS-API library call error.                                |
| <i>GSASL_GSSAPI_IMPORT_NAME_ERROR</i>               | GSS-API library call error.                                |
| <i>GSASL_GSSAPI_INIT_SEC_CONTEXT_ERROR</i>          | GSS-API library call error.                                |
| <i>GSASL_GSSAPI_ACCEPT_SEC_CONTEXT_ERROR</i>        | GSS-API library call error.                                |
| <i>GSASL_GSSAPI_UNWRAP_ERROR</i>                    | GSS-API library call error.                                |
| <i>GSASL_GSSAPI_WRAP_ERROR</i>                      | GSS-API library call error.                                |
| <i>GSASL_GSSAPI_ACQUIRE_CRED_ERROR</i>              | GSS-API library call error.                                |
| <i>GSASL_GSSAPI_DISPLAY_NAME_ERROR</i>              | GSS-API library call error.                                |
| <i>GSASL_GSSAPI_UNSUPPORTED_PROTECTION_ERROR</i>    | An unsupported quality-of-protection layer was requested.  |
| <i>GSASL_GSSAPI_ENCAPSULATE_TOKEN_ERROR</i>         | GSS-API library call error.                                |
| <i>GSASL_GSSAPI_DECAPSULATE_TOKEN_ERROR</i>         | GSS-API library call error.                                |
| <i>GSASL_GSSAPI_INQUIRE_MECH_FOR_SASLNAME_ERROR</i> | GSS-API library call error.                                |
| <i>GSASL_GSSAPI_TEST_OID_SET_MEMBER_ERROR</i>       | GSS-API library call error.                                |
| <i>GSASL_GSSAPI_RELEASE_OID_SET_ERROR</i>           | GSS-API library call error.                                |

## Parameters

|                                                            |                                                 |
|------------------------------------------------------------|-------------------------------------------------|
| <code>GSASL_SECURID_SERVER_NEED_ADDITIONAL_PASSCODE</code> | SecurID mechanism needs an additional passcode. |
| <code>GSASL_SECURID_SERVER_NEED_NEW_PIN</code>             | SecurID mechanism needs a new PIN.              |

Error codes for library functions.

## Enumerator

|                                                            |
|------------------------------------------------------------|
| <code>GSASL_OK</code>                                      |
| <code>GSASL_NEEDS_MORE</code>                              |
| <code>GSASL_UNKNOWN_MECHANISM</code>                       |
| <code>GSASL_MECHANISM_CALLED_TOO_MANY_TIMES</code>         |
| <code>GSASL_MALLOC_ERROR</code>                            |
| <code>GSASL_BASE64_ERROR</code>                            |
| <code>GSASL_CRYPTO_ERROR</code>                            |
| <code>GSASL_SASLPREP_ERROR</code>                          |
| <code>GSASL_MECHANISM_PARSE_ERROR</code>                   |
| <code>GSASL_AUTHENTICATION_ERROR</code>                    |
| <code>GSASL_INTEGRITY_ERROR</code>                         |
| <code>GSASL_NO_CLIENT_CODE</code>                          |
| <code>GSASL_NO_SERVER_CODE</code>                          |
| <code>GSASL_NO_CALLBACK</code>                             |
| <code>GSASL_NO_ANONYMOUS_TOKEN</code>                      |
| <code>GSASL_NO_AUTHID</code>                               |
| <code>GSASL_NO_AUTHZID</code>                              |
| <code>GSASL_NO_PASSWORD</code>                             |
| <code>GSASL_NO_PASSCODE</code>                             |
| <code>GSASL_NO_PIN</code>                                  |
| <code>GSASL_NO_SERVICE</code>                              |
| <code>GSASL_NO_HOSTNAME</code>                             |
| <code>GSASL_NO_CB_TLS_UNIQUE</code>                        |
| <code>GSASL_NO_SAML20_IDP_IDENTIFIER</code>                |
| <code>GSASL_NO_SAML20_REDIRECT_URL</code>                  |
| <code>GSASL_NO_OPENID20_REDIRECT_URL</code>                |
| <code>GSASL_NO_CB_TLS_EXPORTER</code>                      |
| <code>GSASL_GSSAPI_RELEASE_BUFFER_ERROR</code>             |
| <code>GSASL_GSSAPI_IMPORT_NAME_ERROR</code>                |
| <code>GSASL_GSSAPI_INIT_SEC_CONTEXT_ERROR</code>           |
| <code>GSASL_GSSAPI_ACCEPT_SEC_CONTEXT_ERROR</code>         |
| <code>GSASL_GSSAPI_UNWRAP_ERROR</code>                     |
| <code>GSASL_GSSAPI_WRAP_ERROR</code>                       |
| <code>GSASL_GSSAPI_ACQUIRE_CRED_ERROR</code>               |
| <code>GSASL_GSSAPI_DISPLAY_NAME_ERROR</code>               |
| <code>GSASL_GSSAPI_UNSUPPORTED_PROTECTION_ERROR</code>     |
| <code>GSASL_SECURID_SERVER_NEED_ADDITIONAL_PASSCODE</code> |
| <code>GSASL_SECURID_SERVER_NEED_NEW_PIN</code>             |
| <code>GSASL_GSSAPI_ENCAPSULATE_TOKEN_ERROR</code>          |
| <code>GSASL_GSSAPI_DECAPSULATE_TOKEN_ERROR</code>          |
| <code>GSASL_GSSAPI_INQUIRE_MECH_FOR_SASLNAME_ERROR</code>  |

## Enumerator

|                                        |  |
|----------------------------------------|--|
| GSASL_GSSAPI_TEST_OID_SET_MEMBER_ERROR |  |
| GSASL_GSSAPI_RELEASE_OID_SET_ERROR     |  |

Definition at line 127 of file gsasl.h.

### 5.38.3.7 Gsasl\_saslprep\_flags

enum `Gsasl_saslprep_flags`

`Gsasl_saslprep_flags`:

## Parameters

|                                     |                               |
|-------------------------------------|-------------------------------|
| <code>GSASL_ALLOW_UNASSIGNED</code> | Allow unassigned code points. |
|-------------------------------------|-------------------------------|

Flags for the SASLprep function, see [gsasl\\_saslprep\(\)](#). For background, see the GNU Libidn documentation.

## Enumerator

|                        |  |
|------------------------|--|
| GSASL_ALLOW_UNASSIGNED |  |
|------------------------|--|

Definition at line 330 of file gsasl.h.

## 5.38.4 Function Documentation

### 5.38.4.1 gsasl\_base64\_from()

```
__GSASL_API int gsasl_base64_from (
 const char * in,
 size_t inlen,
 char ** out,
 size_t * outlen)
```

`gsasl_base64_from`:

## Parameters

|               |                                                      |
|---------------|------------------------------------------------------|
| <i>in</i>     | input byte array                                     |
| <i>inlen</i>  | size of input byte array                             |
| <i>out</i>    | pointer to newly allocated output byte array         |
| <i>outlen</i> | pointer to size of newly allocated output byte array |

Decode Base64 data. The @out buffer must be deallocated by the caller.

Return value: Returns GSASL\_OK on success, GSASL\_BASE64\_ERROR if input was invalid, and GSASL\_MALLOC\_ERROR on memory allocation errors.

Since: 0.2.2

Definition at line 75 of file base64.c.

#### 5.38.4.2 gssapi\_base64\_to()

```
__GSASL_API int gssapi_base64_to (
 const char * in,
 size_t inlen,
 char ** out,
 size_t * outlen)
```

gssapi\_base64\_to:

##### Parameters

|               |                                                           |
|---------------|-----------------------------------------------------------|
| <i>in</i>     | input byte array.                                         |
| <i>inlen</i>  | size of input byte array.                                 |
| <i>out</i>    | pointer to newly allocated base64-encoded string.         |
| <i>outlen</i> | pointer to size of newly allocated base64-encoded string. |

Encode data as base64. The @out string is zero terminated, and @outlen holds the length excluding the terminating zero. The @out buffer must be deallocated by the caller.

Return value: Returns GSASL\_OK on success, or GSASL\_MALLOC\_ERROR if input was too large or memory allocation fail.

Since: 0.2.2

Definition at line 45 of file base64.c.

#### 5.38.4.3 gssapi\_callback()

```
__GSASL_API int gssapi_callback (
 Gssapi * ctx,
 Gssapi_session * sctx,
 Gssapi_property prop)
```

gssapi\_callback:

##### Parameters

|             |                                                                                           |
|-------------|-------------------------------------------------------------------------------------------|
| <i>ctx</i>  | handle received from <a href="#">gssapi_init()</a> , may be NULL to derive it from @sctx. |
| <i>sctx</i> | session handle.                                                                           |
| <i>prop</i> | enumerated value of Gssapi_property type.                                                 |

Invoke the application callback. The @prop value indicate what the callback is expected to do. For example, for GSASL\_ANONYMOUS\_TOKEN, the function is expected to invoke `gsasl_property_set(@SCTX, GSASL_↔ ANONYMOUS_TOKEN, "token")` where "token" is the anonymous token the application wishes the SASL mechanism to use. See the manual for the meaning of all parameters.

Return value: Returns whatever the application callback returns, or GSASL\_NO\_CALLBACK if no application was known.

Since: 0.2.0

Definition at line 71 of file `callback.c`.

#### 5.38.4.4 `gsasl_callback_hook_get()`

```
__GSASL_API void* gsasl_callback_hook_get (
 Gsasl * ctx)
```

`gsasl_callback_hook_get`:

##### Parameters

|                  |                  |
|------------------|------------------|
| <code>ctx</code> | libgsasl handle. |
|------------------|------------------|

Retrieve application specific data from libgsasl handle.

The application data is set using `gsasl_callback_hook_set()`. This is normally used by the application to maintain a global state between the main program and callbacks.

Return value: Returns the application specific data, or NULL.

Since: 0.2.0

Definition at line 120 of file `callback.c`.

#### 5.38.4.5 `gsasl_callback_hook_set()`

```
__GSASL_API void gsasl_callback_hook_set (
 Gsasl * ctx,
 void * hook)
```

`gsasl_callback_hook_set`:

##### Parameters

|                   |                                              |
|-------------------|----------------------------------------------|
| <code>ctx</code>  | libgsasl handle.                             |
| <code>hook</code> | opaque pointer to application specific data. |



Store application specific data in the libgssapi handle.

The application data can be later (for instance, inside a callback) be retrieved by calling [gssapi\\_callback\\_hook\\_get\(\)](#). This is normally used by the application to maintain a global state between the main program and callbacks.

Since: 0.2.0

Definition at line 100 of file callback.c.

#### 5.38.4.6 gssapi\_callback\_set()

```
_GSSAPI_API void gssapi_callback_set (
 Gssapi * ctx,
 Gssapi_callback_function cb)
```

`gssapi_callback_set`:

##### Parameters

|            |                                                      |
|------------|------------------------------------------------------|
| <i>ctx</i> | handle received from <a href="#">gssapi_init()</a> . |
| <i>cb</i>  | pointer to function implemented by application.      |

Store the pointer to the application provided callback in the library handle. The callback will be used, via [gssapi\\_callback\(\)](#), by mechanisms to discover various parameters (such as username and passwords). The callback function will be called with a `Gssapi_property` value indicating the requested behaviour. For example, for `GSSAPI_↔ ANONYMOUS_TOKEN`, the function is expected to invoke `gssapi_property_set(@CTX, GSSAPI_↔ ANONYMOUS_TOKEN, "token")` where "token" is the anonymous token the application wishes the SASL mechanism to use. See the manual for the meaning of all parameters.

Since: 0.2.0

Definition at line 45 of file callback.c.

#### 5.38.4.7 gssapi\_check\_version()

```
_GSSAPI_API const char* gssapi_check_version (
 const char * req_version)
```

`gssapi_check_version`:

##### Parameters

|                    |                                          |
|--------------------|------------------------------------------|
| <i>req_version</i> | version string to compare with, or NULL. |
|--------------------|------------------------------------------|

Check GNU SASL Library version.

See `GSSAPI_VERSION` for a suitable `@req_version` string.

This function is one of few in the library that can be used without a successful call to [gsasl\\_init\(\)](#).

Return value: Check that the version of the library is at minimum the one given as a string in `@req_version` and return the actual version string of the library; return NULL if the condition is not met. If NULL is passed to this function no check is done and only the version string is returned.

Definition at line 46 of file `version.c`.

#### 5.38.4.8 `gsasl_client_mechlist()`

```
_GSASL_API int gsasl_client_mechlist (
 Gsasl * ctx,
 char ** out)
```

`gsasl_client_mechlist`:

##### Parameters

|            |                                         |
|------------|-----------------------------------------|
| <i>ctx</i> | libgsasl handle.                        |
| <i>out</i> | newly allocated output character array. |

Return a newly allocated string containing SASL names, separated by space, of mechanisms supported by the libgsasl client. `@out` is allocated by this function, and it is the responsibility of caller to deallocate it.

Return value: Returns GSASL\_OK if successful, or error code.

Definition at line 75 of file `listmech.c`.

#### 5.38.4.9 `gsasl_client_start()`

```
_GSASL_API int gsasl_client_start (
 Gsasl * ctx,
 const char * mech,
 Gsasl_session ** sctx)
```

`gsasl_client_start`:

##### Parameters

|             |                           |
|-------------|---------------------------|
| <i>ctx</i>  | libgsasl handle.          |
| <i>mech</i> | name of SASL mechanism.   |
| <i>sctx</i> | pointer to client handle. |

This functions initiates a client SASL authentication. This function must be called before any other `gsasl_client_*` function is called.

Return value: Returns GSASL\_OK if successful, or error code.

Definition at line 120 of file xstart.c.

#### 5.38.4.10 gssapi\_client\_suggest\_mechanism()

```
_GSSAPI_API const char* gssapi_client_suggest_mechanism (
 Gssapi * ctx,
 const char * meclist)
```

gssapi\_client\_suggest\_mechanism:

##### Parameters

|                |                                                                                              |
|----------------|----------------------------------------------------------------------------------------------|
| <i>ctx</i>     | libgssapi handle.                                                                            |
| <i>meclist</i> | input character array with SASL mechanism names, separated by invalid characters (e.g. SPC). |

Given a list of mechanisms, suggest which to use.

Return value: Returns name of "best" SASL mechanism supported by the libgssapi client which is present in the input string, or NULL if no supported mechanism is found.

Definition at line 88 of file suggest.c.

#### 5.38.4.11 gssapi\_client\_support\_p()

```
_GSSAPI_API int gssapi_client_support_p (
 Gssapi * ctx,
 const char * name)
```

gssapi\_client\_support\_p:

##### Parameters

|             |                         |
|-------------|-------------------------|
| <i>ctx</i>  | libgssapi handle.       |
| <i>name</i> | name of SASL mechanism. |

Decide whether there is client-side support for a specified mechanism.

Return value: Returns 1 if the libgssapi client supports the named mechanism, otherwise 0.

Definition at line 50 of file supportp.c.

#### 5.38.4.12 gssapi\_decode()

```
_GSSAPI_API int gssapi_decode (
 Gssapi_session * sctx,
```

```

const char * input,
size_t input_len,
char ** output,
size_t * output_len)

```

gsasl\_decode:

#### Parameters

|                   |                                                            |
|-------------------|------------------------------------------------------------|
| <i>sctx</i>       | libgsasl session handle.                                   |
| <i>input</i>      | input byte array.                                          |
| <i>input_len</i>  | size of input byte array.                                  |
| <i>output</i>     | newly allocated output byte array.                         |
| <i>output_len</i> | pointer to output variable with size of output byte array. |

Decode data according to negotiated SASL mechanism. This might mean that data is integrity or privacy protected.

The @output buffer is allocated by this function, and it is the responsibility of caller to deallocate it by calling `gsasl_free(@output)`.

Return value: Returns GSASL\_OK if encoding was successful, otherwise an error code.

Definition at line 99 of file xcode.c.

#### 5.38.4.13 gsasl\_done()

```

_GSASL_API void gsasl_done (
 Gsasl * ctx)

```

gsasl\_done:

#### Parameters

|            |                  |
|------------|------------------|
| <i>ctx</i> | libgsasl handle. |
|------------|------------------|

This function destroys a libgsasl handle. The handle must not be used with other libgsasl functions after this call.

Definition at line 34 of file done.c.

#### 5.38.4.14 gsasl\_encode()

```

_GSASL_API int gsasl_encode (
 Gsasl_session * sctx,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)

```

gsasl\_encode:

## Parameters

|                   |                                                            |
|-------------------|------------------------------------------------------------|
| <i>sctx</i>       | libgsasl session handle.                                   |
| <i>input</i>      | input byte array.                                          |
| <i>input_len</i>  | size of input byte array.                                  |
| <i>output</i>     | newly allocated output byte array.                         |
| <i>output_len</i> | pointer to output variable with size of output byte array. |

Encode data according to negotiated SASL mechanism. This might mean that data is integrity or privacy protected.

The @output buffer is allocated by this function, and it is the responsibility of caller to deallocate it by calling `gsasl_free(@output)`.

Return value: Returns GSASL\_OK if encoding was successful, otherwise an error code.

Definition at line 66 of file xcode.c.

#### 5.38.4.15 gsasl\_finish()

```
_GSASL_API void gsasl_finish (
 Gsasl_session * sctx)
```

gsasl\_finish:

## Parameters

|             |                          |
|-------------|--------------------------|
| <i>sctx</i> | libgsasl session handle. |
|-------------|--------------------------|

Destroy a libgsasl client or server handle. The handle must not be used with other libgsasl functions after this call.

Definition at line 34 of file xfinish.c.

#### 5.38.4.16 gsasl\_free()

```
_GSASL_API void gsasl_free (
 void * ptr)
```

gsasl\_free:

## Parameters

|            |                |
|------------|----------------|
| <i>ptr</i> | memory pointer |
|------------|----------------|

Invoke `free(@ptr)` to de-allocate memory pointer. Typically used on strings allocated by other libgsasl functions.

This is useful on Windows where libgsasl is linked to one CRT and the application is linked to another CRT. Then malloc/free will not use the same heap. This happens if you build libgsasl using mingw32 and the application with Visual Studio.

Since: 0.2.19

Definition at line 41 of file src/free.c.

#### 5.38.4.17 gsasl\_hash\_length()

```
_GSASL_API size_t gsasl_hash_length (
 Gsasl_hash hash)
```

gsasl\_hash\_length:

##### Parameters

|             |                                                                 |
|-------------|-----------------------------------------------------------------|
| <i>hash</i> | a Gsasl_hash element, e.g., <a href="#">GSASL_HASH_SHA256</a> . |
|-------------|-----------------------------------------------------------------|

Return the digest output size for hash function @hash. For example, gsasl\_hash\_length(GSASL\_HASH\_SHA256) returns GSASL\_HASH\_SHA256\_SIZE which is 32.

Returns: size of supplied Gsasl\_hash element.

Since: 1.10

Definition at line 73 of file crypto.c.

#### 5.38.4.18 gsasl\_hex\_from()

```
_GSASL_API int gsasl_hex_from (
 const char * in,
 char ** out,
 size_t * outlen)
```

gsasl\_hex\_from:

##### Parameters

|               |                                                      |
|---------------|------------------------------------------------------|
| <i>in</i>     | input byte array                                     |
| <i>out</i>    | pointer to newly allocated output byte array         |
| <i>outlen</i> | pointer to size of newly allocated output byte array |

Decode hex data. The @out buffer must be deallocated by the caller.

Return value: Returns GSASL\_OK on success, GSASL\_BASE64\_ERROR if input was invalid, and GSASL\_MALLOCC\_ERROR on memory allocation errors.

Since: 1.10

Definition at line 144 of file base64.c.

#### 5.38.4.19 gsasl\_hex\_to()

```
_GSASL_API int gsasl_hex_to (
 const char * in,
 size_t inlen,
 char ** out,
 size_t * outlen)
```

gsasl\_hex\_to:

##### Parameters

|               |                                                        |
|---------------|--------------------------------------------------------|
| <i>in</i>     | input byte array.                                      |
| <i>inlen</i>  | size of input byte array.                              |
| <i>out</i>    | pointer to newly allocated hex-encoded string.         |
| <i>outlen</i> | pointer to size of newly allocated hex-encoded string. |

Hex encode data. The @out string is zero terminated, and @outlen holds the length excluding the terminating zero. The @out buffer must be deallocated by the caller.

Return value: Returns GSASL\_OK on success, or GSASL\_MALLOC\_ERROR if input was too large or memory allocation fail.

Since: 1.10

Definition at line 111 of file base64.c.

#### 5.38.4.20 gsasl\_init()

```
_GSASL_API int gsasl_init (
 Gsasl ** ctx)
```

gsasl\_init:

##### Parameters

|            |                             |
|------------|-----------------------------|
| <i>ctx</i> | pointer to libgsasl handle. |
|------------|-----------------------------|

This functions initializes libgsasl. The handle pointed to by ctx is valid for use with other libgsasl functions iff this function is successful. It also register all builtin SASL mechanisms, using [gsasl\\_register\(\)](#).

Return value: GSASL\_OK iff successful, otherwise GSASL\_MALLOC\_ERROR.

Definition at line 158 of file init.c.

#### 5.38.4.21 `gsasl_mechanism_name()`

```
__GSASL_API const char* gsasl_mechanism_name (
 Gsasl_session * sctx)
```

`gsasl_mechanism_name`:

##### Parameters

|                   |                          |
|-------------------|--------------------------|
| <code>sctx</code> | libgsasl session handle. |
|-------------------|--------------------------|

This function returns the name of the SASL mechanism used in the session. The pointer must not be deallocated by the caller.

Return value: Returns a zero terminated character array with the name of the SASL mechanism, or NULL if not known.

Since: 0.2.28

Definition at line 39 of file mecname.c.

#### 5.38.4.22 `gsasl_mechanism_name_p()`

```
__GSASL_API int gsasl_mechanism_name_p (
 const char * mech)
```

`gsasl_mechanism_name_p`:

##### Parameters

|                   |                                            |
|-------------------|--------------------------------------------|
| <code>mech</code> | input variable with mechanism name string. |
|-------------------|--------------------------------------------|

Check if the mechanism name string `@mech` follows syntactical rules. It does not check that the name is registered with IANA. It does not check that the mechanism name is actually implemented and supported.

SASL mechanisms are named by strings, from 1 to 20 characters in length, consisting of upper-case letters, digits, hyphens, and/or underscores.

Returns: non-zero when mechanism name string `@mech` conforms to rules, zero when it does not meet the requirements.

Since: 2.0.0

Definition at line 53 of file suggest.c.



**5.38.4.23 gssapi\_nonce()**

```
_GSSAPI int gssapi_nonce (
 char * data,
 size_t datalen)
```

gssapi\_nonce:

**Parameters**

|                |                                                           |
|----------------|-----------------------------------------------------------|
| <i>data</i>    | output array to be filled with unpredictable random data. |
| <i>datalen</i> | size of output array.                                     |

Store unpredictable data of given size in the provided buffer.

Return value: Returns GSSAPI\_OK iff successful.

Definition at line 39 of file crypto.c.

**5.38.4.24 gssapi\_property\_fast()**

```
_GSSAPI const char* gssapi_property_fast (
 Gssapi_session * sctx,
 Gssapi_property prop)
```

gssapi\_property\_fast:

**Parameters**

|             |                                                                                 |
|-------------|---------------------------------------------------------------------------------|
| <i>sctx</i> | session handle.                                                                 |
| <i>prop</i> | enumerated value of Gssapi_property type, indicating the type of data in @data. |

Retrieve the data stored in the session handle for given property @prop.

The pointer is to live data, and must not be deallocated or modified in any way.

This function will not invoke the application callback.

Return value: Return property value, if known, or NULL if no value known.

Since: 0.2.0

Definition at line 262 of file property.c.

**5.38.4.25 gssapi\_property\_free()**

```
_GSSAPI void gssapi_property_free (
 Gssapi_session * sctx,
 Gssapi_property prop)
```

gssapi\_property\_free:

## Parameters

|             |                                                  |
|-------------|--------------------------------------------------|
| <i>sctx</i> | session handle.                                  |
| <i>prop</i> | enumerated value of Gsasl_property type to clear |

Deallocate associated data with property @prop in session handle. After this call, gsasl\_property\_fast(@sctx, @prop) will always return NULL.

Since: 2.0.0

Definition at line 159 of file property.c.

#### 5.38.4.26 gsasl\_property\_get()

```
__GSASL_API const char* gsasl_property_get (
 Gsasl_session * sctx,
 Gsasl_property prop)
```

gsasl\_property\_get:

## Parameters

|             |                                                                                |
|-------------|--------------------------------------------------------------------------------|
| <i>sctx</i> | session handle.                                                                |
| <i>prop</i> | enumerated value of Gsasl_property type, indicating the type of data in @data. |

Retrieve the data stored in the session handle for given property @prop, possibly invoking the application callback to get the value.

The pointer is to live data, and must not be deallocated or modified in any way.

This function will invoke the application callback, using [gsasl\\_callback\(\)](#), when a property value is not known.

Return value: Return data for property, or NULL if no value known.

Since: 0.2.0

Definition at line 292 of file property.c.

#### 5.38.4.27 gsasl\_property\_set()

```
__GSASL_API int gsasl_property_set (
 Gsasl_session * sctx,
 Gsasl_property prop,
 const char * data)
```

gsasl\_property\_set:

## Parameters

|             |                                                                                |
|-------------|--------------------------------------------------------------------------------|
| <i>sctx</i> | session handle.                                                                |
| <i>prop</i> | enumerated value of Gsasl_property type, indicating the type of data in @data. |
| <i>data</i> | zero terminated character string to store.                                     |

Make a copy of @data and store it in the session handle for the indicated property @prop.

You can immediately deallocate @data after calling this function, without affecting the data stored in the session handle.

Return value: GSASL\_OK iff successful, otherwise GSASL\_MALLOC\_ERROR.

Since: 0.2.0

Definition at line 189 of file property.c.

#### 5.38.4.28 gsasl\_property\_set\_raw()

```
_GSASL_API int gsasl_property_set_raw (
 Gsasl_session * sctx,
 Gsasl_property prop,
 const char * data,
 size_t len)
```

gsasl\_property\_set\_raw:

## Parameters

|             |                                                                                |
|-------------|--------------------------------------------------------------------------------|
| <i>sctx</i> | session handle.                                                                |
| <i>prop</i> | enumerated value of Gsasl_property type, indicating the type of data in @data. |
| <i>data</i> | character string to store.                                                     |
| <i>len</i>  | length of character string to store.                                           |

Make a copy of @len sized @data and store a zero terminated version of it in the session handle for the indicated property @prop.

You can immediately deallocate @data after calling this function, without affecting the data stored in the session handle.

Except for the length indicator, this function is identical to gsasl\_property\_set.

Return value: GSASL\_OK iff successful, otherwise GSASL\_MALLOC\_ERROR.

Since: 0.2.0

Definition at line 218 of file property.c.

### 5.38.4.29 gsasl\_random()

```
_GSASL_API int gsasl_random (
 char * data,
 size_t datalen)
```

gsasl\_random:

#### Parameters

|                |                                                    |
|----------------|----------------------------------------------------|
| <i>data</i>    | output array to be filled with strong random data. |
| <i>datalen</i> | size of output array.                              |

Store cryptographically strong random data of given size in the provided buffer.

Return value: Returns GSASL\_OK iff successful.

Definition at line 55 of file crypto.c.

### 5.38.4.30 gsasl\_saslprep()

```
_GSASL_API int gsasl_saslprep (
 const char * in,
 Gsasl_saslprep_flags flags,
 char ** out,
 int * stringpreprc)
```

### 5.38.4.31 gsasl\_scram\_secrets\_from\_password()

```
_GSASL_API int gsasl_scram_secrets_from_password (
 Gsasl_hash hash,
 const char * password,
 unsigned int iteration_count,
 const char * salt,
 size_t saltlen,
 char * salted_password,
 char * client_key,
 char * server_key,
 char * stored_key)
```

gsasl\_scram\_secrets\_from\_password:

#### Parameters

|                        |                                                                 |
|------------------------|-----------------------------------------------------------------|
| <i>hash</i>            | a Gsasl_hash element, e.g., <a href="#">GSASL_HASH_SHA256</a> . |
| <i>password</i>        | input parameter with password.                                  |
| <i>iteration_count</i> | number of PBKDF2 rounds to apply.                               |
| <i>salt</i>            | input character array of @saltlen length with salt for PBKDF2.  |
| <i>saltlen</i>         | length of @salt.                                                |
| <i>salted_password</i> | pre-allocated output array with derived salted password.        |
| <i>client_key</i>      | pre-allocated output array with derived client key.             |
| <i>server_key</i>      | pre-allocated output array with derived server key.             |
| <i>stored_key</i>      | pre-allocated output array with derived stored key.             |

Helper function to generate SCRAM secrets from a password. The @salted\_password, @client\_key, @server\_key, and @stored\_key buffers must have room to hold digest for given @hash, use [GSASL\\_HASH\\_MAX\\_SIZE](#) which is sufficient for all hashes.

Return value: Returns GSASL\_OK if successful, or error code.

Since: 1.10

Definition at line 156 of file crypto.c.

#### 5.38.4.32 gssapi\_scram\_secrets\_from\_salted\_password()

```
_GSASL_API int gssapi_scram_secrets_from_salted_password (
 Gssapi_hash hash,
 const char * salted_password,
 char * client_key,
 char * server_key,
 char * stored_key)
```

gssapi\_scram\_secrets\_from\_salted\_password:

##### Parameters

|                        |                                                                                  |
|------------------------|----------------------------------------------------------------------------------|
| <i>hash</i>            | a <a href="#">Gssapi_hash</a> element, e.g., <a href="#">GSASL_HASH_SHA256</a> . |
| <i>salted_password</i> | input array with salted password.                                                |
| <i>client_key</i>      | pre-allocated output array with derived client key.                              |
| <i>server_key</i>      | pre-allocated output array with derived server key.                              |
| <i>stored_key</i>      | pre-allocated output array with derived stored key.                              |

Helper function to derive SCRAM ClientKey/ServerKey/StoredKey. The @client\_key, @server\_key, and @stored\_key buffers must have room to hold digest for given @hash, use [GSASL\\_HASH\\_MAX\\_SIZE](#) which is sufficient for all hashes.

Return value: Returns GSASL\_OK if successful, or error code.

Since: 1.10

Definition at line 104 of file crypto.c.

#### 5.38.4.33 gssapi\_server\_mechlist()

```
_GSASL_API int gssapi_server_mechlist (
 Gssapi * ctx,
 char ** out)
```

gssapi\_server\_mechlist:

## Parameters

|            |                                         |
|------------|-----------------------------------------|
| <i>ctx</i> | libgsasl handle.                        |
| <i>out</i> | newly allocated output character array. |

Return a newly allocated string containing SASL names, separated by space, of mechanisms supported by the libgsasl server. @out is allocated by this function, and it is the responsibility of caller to deallocate it.

Return value: Returns GSASL\_OK if successful, or error code.

Definition at line 94 of file listmech.c.

#### 5.38.4.34 gsasl\_server\_start()

```
__GSASL_API int gsasl_server_start (
 Gsasl * ctx,
 const char * mech,
 Gsasl_session ** sctx)
```

gsasl\_server\_start:

## Parameters

|             |                           |
|-------------|---------------------------|
| <i>ctx</i>  | libgsasl handle.          |
| <i>mech</i> | name of SASL mechanism.   |
| <i>sctx</i> | pointer to server handle. |

This functions initiates a server SASL authentication. This function must be called before any other gsasl\_server\_\*( ) function is called.

Return value: Returns GSASL\_OK if successful, or error code.

Definition at line 138 of file xstart.c.

#### 5.38.4.35 gsasl\_server\_support\_p()

```
__GSASL_API int gsasl_server_support_p (
 Gsasl * ctx,
 const char * name)
```

gsasl\_server\_support\_p:

## Parameters

|             |                         |
|-------------|-------------------------|
| <i>ctx</i>  | libgsasl handle.        |
| <i>name</i> | name of SASL mechanism. |

Decide whether there is server-side support for a specified mechanism.

Return value: Returns 1 if the libgssapi server supports the named mechanism, otherwise 0.

Definition at line 67 of file supportp.c.

#### 5.38.4.36 gssapi\_session\_hook\_get()

```
__GSSAPI_API void* gssapi_session_hook_get (
 Gssapi_session * sctx)
```

gssapi\_session\_hook\_get:

##### Parameters

|             |                           |
|-------------|---------------------------|
| <i>sctx</i> | libgssapi session handle. |
|-------------|---------------------------|

Retrieve application specific data from libgssapi session handle.

The application data is set using [gssapi\\_callback\\_hook\\_set\(\)](#). This is normally used by the application to maintain a per-session state between the main program and callbacks.

Return value: Returns the application specific data, or NULL.

Since: 0.2.14

Definition at line 160 of file callback.c.

#### 5.38.4.37 gssapi\_session\_hook\_set()

```
__GSSAPI_API void gssapi_session_hook_set (
 Gssapi_session * sctx,
 void * hook)
```

gssapi\_session\_hook\_set:

##### Parameters

|             |                                              |
|-------------|----------------------------------------------|
| <i>sctx</i> | libgssapi session handle.                    |
| <i>hook</i> | opaque pointer to application specific data. |

Store application specific data in the libgssapi session handle.

The application data can be later (for instance, inside a callback) be retrieved by calling [gssapi\\_session\\_hook\\_get\(\)](#). This is normally used by the application to maintain a per-session state between the main program and callbacks.

Since: 0.2.14

Definition at line 140 of file callback.c.

#### 5.38.4.38 gsasl\_simple\_getpass()

```
_GSASL_API int gsasl_simple_getpass (
 const char * filename,
 const char * username,
 char ** key)
```

gsasl\_simple\_getpass:

##### Parameters

|                 |                                         |
|-----------------|-----------------------------------------|
| <i>filename</i> | filename of file containing passwords.  |
| <i>username</i> | username string.                        |
| <i>key</i>      | newly allocated output character array. |

Retrieve password for user from specified file. The buffer @key contain the password if this function is successful. The caller is responsible for deallocating it.

The file should be on the UoW "MD5 Based Authentication" format, which means it is in text format with comments denoted by # first on the line, with user entries looking as "usernameTABpassword". This function removes CR and LF at the end of lines before processing. TAB, CR, and LF denote ASCII values 9, 13, and 10, respectively.

Return value: Return GSASL\_OK if output buffer contains the password, GSASL\_AUTHENTICATION\_ERROR if the user could not be found, or other error code.

Definition at line 48 of file md5pwd.c.

#### 5.38.4.39 gsasl\_step()

```
_GSASL_API int gsasl_step (
 Gsasl_session * sctx,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

gsasl\_step:

##### Parameters

|                   |                                                            |
|-------------------|------------------------------------------------------------|
| <i>sctx</i>       | libgsasl session handle.                                   |
| <i>input</i>      | input byte array.                                          |
| <i>input_len</i>  | size of input byte array.                                  |
| <i>output</i>     | newly allocated output byte array.                         |
| <i>output_len</i> | pointer to output variable with size of output byte array. |



Perform one step of SASL authentication. This reads data from the other end (from @input and @input\_len), processes it (potentially invoking callbacks to the application), and writes data to server (into newly allocated variable @output and @output\_len that indicate the length of @output).

The contents of the @output buffer is unspecified if this functions returns anything other than GSASL\_OK or GSASL\_NEEDS\_MORE. If this function return GSASL\_OK or GSASL\_NEEDS\_MORE, however, the @output buffer is allocated by this function, and it is the responsibility of caller to deallocate it by calling gsasl\_free(@output).

Return value: Returns GSASL\_OK if authenticated terminated successfully, GSASL\_NEEDS\_MORE if more data is needed, or error code.

Definition at line 52 of file xstep.c.

#### 5.38.4.40 gsasl\_step64()

```
_GSASL_API int gsasl_step64 (
 Gsasl_session * sctx,
 const char * b64input,
 char ** b64output)
```

gsasl\_step64:

##### Parameters

|                  |                                                   |
|------------------|---------------------------------------------------|
| <i>sctx</i>      | libgsasl client handle.                           |
| <i>b64input</i>  | input base64 encoded byte array.                  |
| <i>b64output</i> | newly allocated output base64 encoded byte array. |

This is a simple wrapper around [gsasl\\_step\(\)](#) that base64 decodes the input and base64 encodes the output.

The contents of the @b64output buffer is unspecified if this functions returns anything other than GSASL\_OK or GSASL\_NEEDS\_MORE. If this function return GSASL\_OK or GSASL\_NEEDS\_MORE, however, the @b64output buffer is allocated by this function, and it is the responsibility of caller to deallocate it by calling gsasl\_free(@b64output).

Return value: Returns GSASL\_OK if authenticated terminated successfully, GSASL\_NEEDS\_MORE if more data is needed, or error code.

Definition at line 87 of file xstep.c.

#### 5.38.4.41 gsasl\_strerror()

```
_GSASL_API const char* gsasl_strerror (
 int err)
```

gsasl\_strerror:

## Parameters

|            |                     |
|------------|---------------------|
| <i>err</i> | libgsasl error code |
|------------|---------------------|

Convert return code to human readable string explanation of the reason for the particular error code.

This string can be used to output a diagnostic message to the user.

This function is one of few in the library that can be used without a successful call to [gsasl\\_init\(\)](#).

Return value: Returns a pointer to a statically allocated string containing an explanation of the error code @err.

Definition at line 185 of file error.c.

#### 5.38.4.42 gsasl\_strerror\_name()

```
__GSASL_API const char* gsasl_strerror_name (
 int err)
```

gsasl\_strerror\_name:

## Parameters

|            |                     |
|------------|---------------------|
| <i>err</i> | libgsasl error code |
|------------|---------------------|

Convert return code to human readable string representing the error code symbol itself. For example, `gsasl_strerror_name(GSASL_OK)` returns the string "GSASL\_OK".

This string can be used to output a diagnostic message to the user.

This function is one of few in the library that can be used without a successful call to [gsasl\\_init\(\)](#).

Return value: Returns a pointer to a statically allocated string containing a string version of the error code @err, or NULL if the error code is not known.

Since: 0.2.29

Definition at line 223 of file error.c.

## 5.39 init.c File Reference

```
#include <config.h>
#include "internal.h"
#include <gc.h>
#include "cram-md5/cram-md5.h"
#include "external/external.h"
#include "gssapi/x-gssapi.h"
#include "gs2/gs2.h"
#include "anonymous/anonymous.h"
```

```
#include "plain/plain.h"
#include "securid/securid.h"
#include "digest-md5/digest-md5.h"
#include "scram/scram.h"
#include "saml20/saml20.h"
#include "openid20/openid20.h"
#include "login/login.h"
#include "ntlm/x-ntlm.h"
```

## Functions

- int [gsasl\\_init](#) ([Gsasl](#) \*\*ctx)

### 5.39.1 Function Documentation

#### 5.39.1.1 [gsasl\\_init\(\)](#)

```
int gsasl_init (
 Gsasl ** ctx)
```

gsasl\_init:

##### Parameters

|            |                             |
|------------|-----------------------------|
| <i>ctx</i> | pointer to libgsasl handle. |
|------------|-----------------------------|

This functions initializes libgsasl. The handle pointed to by *ctx* is valid for use with other libgsasl functions iff this function is successful. It also register all builtin SASL mechanisms, using [gsasl\\_register\(\)](#).

Return value: GSASL\_OK iff successful, otherwise GSASL\_MALLOC\_ERROR.

Definition at line 158 of file init.c.

## 5.40 internal.h File Reference

```
#include "gsasl.h"
#include <stdlib.h>
#include <string.h>
```

## Data Structures

- struct [Gsasl](#)
- struct [Gsasl\\_session](#)

## 5.41 listmech.c File Reference

```
#include <config.h>
#include "internal.h"
```

### Functions

- int [gsasl\\_client\\_mechlist](#) ([Gsasl](#) \*ctx, char \*\*out)
- int [gsasl\\_server\\_mechlist](#) ([Gsasl](#) \*ctx, char \*\*out)

### 5.41.1 Function Documentation

#### 5.41.1.1 [gsasl\\_client\\_mechlist\(\)](#)

```
int gsasl_client_mechlist (
 Gsasl * ctx,
 char ** out)
```

[gsasl\\_client\\_mechlist](#):

#### Parameters

|            |                                         |
|------------|-----------------------------------------|
| <i>ctx</i> | libgsasl handle.                        |
| <i>out</i> | newly allocated output character array. |

Return a newly allocated string containing SASL names, separated by space, of mechanisms supported by the libgsasl client. @out is allocated by this function, and it is the responsibility of caller to deallocate it.

Return value: Returns GSASL\_OK if successful, or error code.

Definition at line 75 of file listmech.c.

#### 5.41.1.2 [gsasl\\_server\\_mechlist\(\)](#)

```
int gsasl_server_mechlist (
 Gsasl * ctx,
 char ** out)
```

[gsasl\\_server\\_mechlist](#):

#### Parameters

|            |                                         |
|------------|-----------------------------------------|
| <i>ctx</i> | libgsasl handle.                        |
| <i>out</i> | newly allocated output character array. |

Return a newly allocated string containing SASL names, separated by space, of mechanisms supported by the libgsasl server. @out is allocated by this function, and it is the responsibility of caller to deallocate it.

Return value: Returns GSASL\_OK if successful, or error code.

Definition at line 94 of file listmech.c.

## 5.42 login.h File Reference

```
#include <gsasl.h>
```

### Macros

- `#define GSASL_LOGIN_NAME "LOGIN"`

### Functions

- `int _gsasl_login_client_start (Gsasl_session *sctx, void **mech_data)`
- `int _gsasl_login_client_step (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)`
- `void _gsasl_login_client_finish (Gsasl_session *sctx, void *mech_data)`
- `int _gsasl_login_server_start (Gsasl_session *sctx, void **mech_data)`
- `int _gsasl_login_server_step (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)`
- `void _gsasl_login_server_finish (Gsasl_session *sctx, void *mech_data)`

### Variables

- `Gsasl_mechanism _gsasl_login_mechanism`

#### 5.42.1 Macro Definition Documentation

##### 5.42.1.1 GSASL\_LOGIN\_NAME

```
#define GSASL_LOGIN_NAME "LOGIN"
```

Definition at line 28 of file login.h.

#### 5.42.2 Function Documentation

#### 5.42.2.1 `_gsasl_login_client_finish()`

```
void _gsasl_login_client_finish (
 Gsasl_session * sctx,
 void * mech_data)
```

#### 5.42.2.2 `_gsasl_login_client_start()`

```
int _gsasl_login_client_start (
 Gsasl_session * sctx,
 void ** mech_data)
```

#### 5.42.2.3 `_gsasl_login_client_step()`

```
int _gsasl_login_client_step (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

#### 5.42.2.4 `_gsasl_login_server_finish()`

```
void _gsasl_login_server_finish (
 Gsasl_session * sctx,
 void * mech_data)
```

#### 5.42.2.5 `_gsasl_login_server_start()`

```
int _gsasl_login_server_start (
 Gsasl_session * sctx,
 void ** mech_data)
```

### 5.42.2.6 `_gsasl_login_server_step()`

```
int _gsasl_login_server_step (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 59 of file login/server.c.

## 5.42.3 Variable Documentation

### 5.42.3.1 `_gsasl_login_mechanism`

```
Gsasl_mechanism _gsasl_login_mechanism [extern]
```

Definition at line 28 of file login/mechinfo.c.

## 5.43 md5pwd.c File Reference

```
#include <config.h>
#include "internal.h"
```

### Functions

- int `gsasl_simple_getpass` (const char \*filename, const char \*username, char \*\*key)

### 5.43.1 Function Documentation

#### 5.43.1.1 `gsasl_simple_getpass()`

```
int gsasl_simple_getpass (
 const char * filename,
 const char * username,
 char ** key)
```

`gsasl_simple_getpass`:

**Parameters**

|                 |                                         |
|-----------------|-----------------------------------------|
| <i>filename</i> | filename of file containing passwords.  |
| <i>username</i> | username string.                        |
| <i>key</i>      | newly allocated output character array. |

Retrieve password for user from specified file. The buffer @key contain the password if this function is successful. The caller is responsible for deallocating it.

The file should be on the UoW "MD5 Based Authentication" format, which means it is in text format with comments denoted by # first on the line, with user entries looking as "usernameTABpassword". This function removes CR and LF at the end of lines before processing. TAB, CR, and LF denote ASCII values 9, 13, and 10, respectively.

Return value: Return GSASL\_OK if output buffer contains the password, GSASL\_AUTHENTICATION\_ERROR if the user could not be found, or other error code.

Definition at line 48 of file md5pwd.c.

**5.44 mechinfo.c File Reference**

```
#include <config.h>
#include "anonymous.h"
```

**Variables**

- [Gsasl\\_mechanism \\_\\_gsasl\\_anonymous\\_mechanism](#)

**5.44.1 Variable Documentation****5.44.1.1 \_\_gsasl\_anonymous\_mechanism**

[Gsasl\\_mechanism \\_\\_gsasl\\_anonymous\\_mechanism](#)

**Initial value:**

```
= {
 GSASL_ANONYMOUS_NAME,
 {
 NULL,
 NULL,
 NULL,
 NULL,
 NULL,
 NULL,
 NULL,
 NULL}
,
 {
 NULL,
 NULL,
 NULL,
 NULL,
 NULL,
 NULL,
 NULL}
}
```

Definition at line 28 of file anonymous/mechinfo.c.



## 5.45 mechinfo.c File Reference

```
#include <config.h>
#include "cram-md5.h"
```

### Variables

- [Gsasl\\_mechanism\\_gsasl\\_cram\\_md5\\_mechanism](#)

### 5.45.1 Variable Documentation

#### 5.45.1.1 `_gsasl_cram_md5_mechanism`

[Gsasl\\_mechanism\\_gsasl\\_cram\\_md5\\_mechanism](#)

Definition at line 28 of file cram-md5/mechinfo.c.

## 5.46 mechinfo.c File Reference

```
#include <config.h>
#include "digest-md5.h"
```

### Variables

- [Gsasl\\_mechanism\\_gsasl\\_digest\\_md5\\_mechanism](#)

### 5.46.1 Variable Documentation

#### 5.46.1.1 `_gsasl_digest_md5_mechanism`

[Gsasl\\_mechanism\\_gsasl\\_digest\\_md5\\_mechanism](#)

Definition at line 28 of file digest-md5/mechinfo.c.

## 5.47 mechinfo.c File Reference

```
#include <config.h>
#include "external.h"
```

### Variables

- [Gsasl\\_mechanism\\_gsasl\\_external\\_mechanism](#)

### 5.47.1 Variable Documentation

#### 5.47.1.1 \_gsasl\_external\_mechanism

`Gsasl_mechanism_gsasl_external_mechanism`

##### Initial value:

```
= {
 GSASL_EXTERNAL_NAME,
 {
 NULL,
 NULL,
 NULL,
 NULL,
 NULL,
 NULL,
 NULL,
 NULL,
 NULL,
 NULL,
 },
 {
 NULL,
 NULL,
 NULL,
 NULL,
 NULL,
 NULL,
 NULL,
 NULL,
 }
}
```

Definition at line 28 of file external/mechinfo.c.

## 5.48 mechinfo.c File Reference

```
#include <config.h>
#include "gs2.h"
```

### Variables

- [Gsasl\\_mechanism\\_gsasl\\_gs2\\_krb5\\_mechanism](#)

### 5.48.1 Variable Documentation

### 5.48.1.1 `_gsasl_gs2_krb5_mechanism`

`Gsasl_mechanism` `_gsasl_gs2_krb5_mechanism`

Definition at line 28 of file gs2/mechinfo.c.

## 5.49 mechinfo.c File Reference

```
#include <config.h>
#include "x-gssapi.h"
```

### Variables

- `Gsasl_mechanism` `_gsasl_gssapi_mechanism`

### 5.49.1 Variable Documentation

#### 5.49.1.1 `_gsasl_gssapi_mechanism`

`Gsasl_mechanism` `_gsasl_gssapi_mechanism`

Definition at line 28 of file gssapi/mechinfo.c.

## 5.50 mechinfo.c File Reference

```
#include <config.h>
#include "login.h"
```

### Variables

- `Gsasl_mechanism` `_gsasl_login_mechanism`

### 5.50.1 Variable Documentation

### 5.50.1.1 `_gsasl_login_mechanism`

`Gsasl_mechanism` `_gsasl_login_mechanism`

Definition at line 28 of file login/mechinfo.c.

## 5.51 `mechinfo.c` File Reference

```
#include <config.h>
#include "x-ntlm.h"
```

### Variables

- `Gsasl_mechanism` `_gsasl_ntlm_mechanism`

### 5.51.1 Variable Documentation

#### 5.51.1.1 `_gsasl_ntlm_mechanism`

`Gsasl_mechanism` `_gsasl_ntlm_mechanism`

Definition at line 28 of file ntlm/mechinfo.c.

## 5.52 `mechinfo.c` File Reference

```
#include <config.h>
#include "openid20.h"
```

### Variables

- `Gsasl_mechanism` `_gsasl_openid20_mechanism`

### 5.52.1 Variable Documentation

### 5.52.1.1 `_gsasl_openid20_mechanism`

`Gsasl_mechanism` `_gsasl_openid20_mechanism`

Definition at line 28 of file openid20/mechinfo.c.

## 5.53 mechinfo.c File Reference

```
#include <config.h>
#include "plain.h"
```

### Variables

- `Gsasl_mechanism` `_gsasl_plain_mechanism`

### 5.53.1 Variable Documentation

#### 5.53.1.1 `_gsasl_plain_mechanism`

`Gsasl_mechanism` `_gsasl_plain_mechanism`

##### Initial value:

```
= {
 GSASL_PLAIN_NAME,
 {
 NULL,
 NULL,
 NULL,
 NULL,
 NULL,
 NULL,
 NULL,
 NULL}
,
 {
 NULL,
 NULL,
 NULL,
 NULL,
 NULL,
 NULL,
 NULL}
}
```

Definition at line 28 of file plain/mechinfo.c.

## 5.54 mechinfo.c File Reference

```
#include <config.h>
#include "saml20.h"
```

## Variables

- [Gsasl\\_mechanism \\_gsasl\\_saml20\\_mechanism](#)

### 5.54.1 Variable Documentation

#### 5.54.1.1 `_gsasl_saml20_mechanism`

`Gsasl_mechanism _gsasl_saml20_mechanism`

Definition at line 28 of file saml20/mechinfo.c.

## 5.55 mechinfo.c File Reference

```
#include <config.h>
#include "scram.h"
```

## 5.56 mechinfo.c File Reference

```
#include <config.h>
#include "securid.h"
```

## Variables

- [Gsasl\\_mechanism \\_gsasl\\_securid\\_mechanism](#)

### 5.56.1 Variable Documentation

#### 5.56.1.1 `_gsasl_securid_mechanism`

`Gsasl_mechanism _gsasl_securid_mechanism`

Definition at line 28 of file securid/mechinfo.c.

## 5.57 mechname.c File Reference

```
#include <config.h>
#include "internal.h"
```

### Functions

- const char \* [gsasl\\_mechanism\\_name](#) ([Gsasl\\_session](#) \*sctx)

### 5.57.1 Function Documentation

#### 5.57.1.1 [gsasl\\_mechanism\\_name\(\)](#)

```
const char* gsasl_mechanism_name (
 Gsasl_session * sctx)
```

gsasl\_mechanism\_name:

#### Parameters

|             |                          |
|-------------|--------------------------|
| <i>sctx</i> | libgsasl session handle. |
|-------------|--------------------------|

This function returns the name of the SASL mechanism used in the session. The pointer must not be deallocated by the caller.

Return value: Returns a zero terminated character array with the name of the SASL mechanism, or NULL if not known.

Since: 0.2.28

Definition at line 39 of file mechname.c.

## 5.58 mechttools.c File Reference

```
#include <config.h>
#include "mechttools.h"
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <gsasl.h>
#include <gc.h>
```

## Functions

- `int _gsasl_parse_gs2_header` (const char \*data, size\_t len, char \*\*authzid, size\_t \*headerlen)
- `int _gsasl_gs2_generate_header` (bool nonstd, char cbflag, const char \*cbname, const char \*authzid, size\_t extralen, const char \*extra, char \*\*gs2h, size\_t \*gs2hlen)
- `void _gsasl_hex_encode` (const char \*in, size\_t inlen, char \*out)
- `void _gsasl_hex_decode` (const char \*hexstr, char \*bin)
- `bool _gsasl_hex_p` (const char \*hexstr)
- `int _gsasl_hash` (`Gsasl_hash` hash, const char \*in, size\_t inlen, char \*outhash)
- `int _gsasl_hmac` (`Gsasl_hash` hash, const char \*key, size\_t keylen, const char \*in, size\_t inlen, char \*outhash)
- `int _gsasl_pbkdf2` (`Gsasl_hash` hash, const char \*password, size\_t passwordlen, const char \*salt, size\_t saltlen, unsigned int c, char \*dk, size\_t dklen)

### 5.58.1 Function Documentation

#### 5.58.1.1 `_gsasl_gs2_generate_header()`

```
int _gsasl_gs2_generate_header (
 bool nonstd,
 char cbflag,
 const char * cbname,
 const char * authzid,
 size_t extralen,
 const char * extra,
 char ** gs2h,
 size_t * gs2hlen)
```

Definition at line 166 of file mechtools.c.

#### 5.58.1.2 `_gsasl_hash()`

```
int _gsasl_hash (
 Gsasl_hash hash,
 const char * in,
 size_t inlen,
 char * outhash)
```

Definition at line 296 of file mechtools.c.

#### 5.58.1.3 `_gsasl_hex_decode()`

```
void _gsasl_hex_decode (
 const char * hexstr,
 char * bin)
```

Definition at line 256 of file mechtools.c.



#### 5.58.1.4 `_gsasl_hex_encode()`

```
void _gsasl_hex_encode (
 const char * in,
 size_t inlen,
 char * out)
```

Definition at line 221 of file mechttools.c.

#### 5.58.1.5 `_gsasl_hex_p()`

```
bool _gsasl_hex_p (
 const char * hexstr)
```

Definition at line 268 of file mechttools.c.

#### 5.58.1.6 `_gsasl_hmac()`

```
int _gsasl_hmac (
 Gsasl_hash hash,
 const char * key,
 size_t keylen,
 const char * in,
 size_t inlen,
 char * outhash)
```

Definition at line 329 of file mechttools.c.

#### 5.58.1.7 `_gsasl_parse_gs2_header()`

```
int _gsasl_parse_gs2_header (
 const char * data,
 size_t len,
 char ** authzid,
 size_t * headerlen)
```

Definition at line 97 of file mechttools.c.

### 5.58.1.8 `_gsasl_pbkdf2()`

```
int _gsasl_pbkdf2 (
 Gsasl_hash hash,
 const char * password,
 size_t passwordlen,
 const char * salt,
 size_t saltlen,
 unsigned int c,
 char * dk,
 size_t dklen)
```

Definition at line 368 of file mechtools.c.

## 5.59 mechtools.h File Reference

```
#include <stddef.h>
#include <stdbool.h>
#include <gsasl.h>
```

### Functions

- int [\\_gsasl\\_parse\\_gs2\\_header](#) (const char \*data, size\_t len, char \*\*authzid, size\_t \*headerlen)
- int [\\_gsasl\\_gs2\\_generate\\_header](#) (bool nonstd, char cbflag, const char \*cbname, const char \*authzid, size\_t extralen, const char \*extra, char \*\*gs2h, size\_t \*gs2hlen)
- void [\\_gsasl\\_hex\\_encode](#) (const char \*in, size\_t inlen, char \*out)
- void [\\_gsasl\\_hex\\_decode](#) (const char \*hexstr, char \*bin)
- bool [\\_gsasl\\_hex\\_p](#) (const char \*hexstr)
- int [\\_gsasl\\_hash](#) (Gsasl\_hash hash, const char \*in, size\_t inlen, char \*out)
- int [\\_gsasl\\_hmac](#) (Gsasl\_hash hash, const char \*key, size\_t keylen, const char \*in, size\_t inlen, char \*outhash)
- int [\\_gsasl\\_pbkdf2](#) (Gsasl\_hash hash, const char \*password, size\_t passwordlen, const char \*salt, size\_t saltlen, unsigned int c, char \*dk, size\_t dklen)

### 5.59.1 Function Documentation

#### 5.59.1.1 `_gsasl_gs2_generate_header()`

```
int _gsasl_gs2_generate_header (
 bool nonstd,
 char cbflag,
 const char * cbname,
 const char * authzid,
 size_t extralen,
 const char * extra,
 char ** gs2h,
 size_t * gs2hlen)
```

Definition at line 166 of file mechtools.c.

### 5.59.1.2 `_gsasl_hash()`

```
int _gsasl_hash (
 Gsasl_hash hash,
 const char * in,
 size_t inlen,
 char * out)
```

Definition at line 296 of file mechttools.c.

### 5.59.1.3 `_gsasl_hex_decode()`

```
void _gsasl_hex_decode (
 const char * hexstr,
 char * bin)
```

Definition at line 256 of file mechttools.c.

### 5.59.1.4 `_gsasl_hex_encode()`

```
void _gsasl_hex_encode (
 const char * in,
 size_t inlen,
 char * out)
```

Definition at line 221 of file mechttools.c.

### 5.59.1.5 `_gsasl_hex_p()`

```
bool _gsasl_hex_p (
 const char * hexstr)
```

Definition at line 268 of file mechttools.c.

### 5.59.1.6 `_gsasl_hmac()`

```
int _gsasl_hmac (
 Gsasl_hash hash,
 const char * key,
 size_t keylen,
 const char * in,
 size_t inlen,
 char * outhash)
```

Definition at line 329 of file mechttools.c.

### 5.59.1.7 `_gsasl_parse_gs2_header()`

```
int _gsasl_parse_gs2_header (
 const char * data,
 size_t len,
 char ** authzid,
 size_t * headerlen)
```

Definition at line 97 of file mechtools.c.

### 5.59.1.8 `_gsasl_pbkdf2()`

```
int _gsasl_pbkdf2 (
 Gsasl_hash hash,
 const char * password,
 size_t passwordlen,
 const char * salt,
 size_t saltlen,
 unsigned int c,
 char * dk,
 size_t dklen)
```

Definition at line 368 of file mechtools.c.

## 5.60 nonascii.c File Reference

```
#include <config.h>
#include "nonascii.h"
#include <stdlib.h>
#include <string.h>
```

### Functions

- char \* [latin1toutf8](#) (const char \*str)
- char \* [utf8tolatin1ifpossible](#) (const char \*passwd)

### 5.60.1 Function Documentation

#### 5.60.1.1 `latin1toutf8()`

```
char* latin1toutf8 (
 const char * str)
```

Definition at line 39 of file nonascii.c.

### 5.60.1.2 utf8tolatin1ifpossible()

```
char* utf8tolatin1ifpossible (
 const char * passwd)
```

Definition at line 67 of file nonascii.c.

## 5.61 nonascii.h File Reference

### Functions

- char \* [latin1toutf8](#) (const char \*str)
- char \* [utf8tolatin1ifpossible](#) (const char \*passwd)

### 5.61.1 Function Documentation

#### 5.61.1.1 latin1toutf8()

```
char* latin1toutf8 (
 const char * str)
```

Definition at line 39 of file nonascii.c.

#### 5.61.1.2 utf8tolatin1ifpossible()

```
char* utf8tolatin1ifpossible (
 const char * passwd)
```

Definition at line 67 of file nonascii.c.

## 5.62 ntlm.c File Reference

```
#include <config.h>
#include <stdlib.h>
#include <string.h>
#include "x-ntlm.h"
#include <ntlm.h>
```

### Data Structures

- struct [\\_Gsasl\\_ntlm\\_state](#)

## Typedefs

- typedef struct [\\_Gssasl\\_ntlm\\_state](#) [\\_Gssasl\\_ntlm\\_state](#)

## Functions

- int [\\_gssasl\\_ntlm\\_client\\_start](#) ([Gssasl\\_session](#) \*sctx [\\_GL\\_UNUSED](#), void \*\*mech\_data)
- int [\\_gssasl\\_ntlm\\_client\\_step](#) ([Gssasl\\_session](#) \*sctx, void \*mech\_data, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len)
- void [\\_gssasl\\_ntlm\\_client\\_finish](#) ([Gssasl\\_session](#) \*sctx [\\_GL\\_UNUSED](#), void \*mech\_data)

### 5.62.1 Typedef Documentation

#### 5.62.1.1 [\\_Gssasl\\_ntlm\\_state](#)

```
typedef struct _Gssasl_ntlm_state _Gssasl_ntlm_state
```

Definition at line 1 of file ntlm.c.

### 5.62.2 Function Documentation

#### 5.62.2.1 [\\_gssasl\\_ntlm\\_client\\_finish\(\)](#)

```
void _gssasl_ntlm_client_finish (
 Gssasl_session *sctx _GL_UNUSED,
 void * mech_data)
```

Definition at line 165 of file ntlm.c.

#### 5.62.2.2 [\\_gssasl\\_ntlm\\_client\\_start\(\)](#)

```
int _gssasl_ntlm_client_start (
 Gssasl_session *sctx _GL_UNUSED,
 void ** mech_data)
```

Definition at line 43 of file ntlm.c.

### 5.62.2.3 `_gsasl_ntlm_client_step()`

```
int _gsasl_ntlm_client_step (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 59 of file ntlm.c.

## 5.63 `openid20.h` File Reference

```
#include <gsasl.h>
```

### Macros

- `#define GSASL_OPENID20_NAME "OPENID20"`

### Functions

- `int _gsasl_openid20_client_start(Gsasl_session *sctx, void **mech_data)`
- `int _gsasl_openid20_client_step(Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)`
- `void _gsasl_openid20_client_finish(Gsasl_session *sctx, void *mech_data)`
- `int _gsasl_openid20_server_start(Gsasl_session *sctx, void **mech_data)`
- `int _gsasl_openid20_server_step(Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)`
- `void _gsasl_openid20_server_finish(Gsasl_session *sctx, void *mech_data)`

### Variables

- `Gsasl_mechanism _gsasl_openid20_mechanism`

## 5.63.1 Macro Definition Documentation

### 5.63.1.1 `GSASL_OPENID20_NAME`

```
#define GSASL_OPENID20_NAME "OPENID20"
```

Definition at line 28 of file openid20.h.

## 5.63.2 Function Documentation

### 5.63.2.1 `__gsasl_openid20_client_finish()`

```
void __gsasl_openid20_client_finish (
 Gsasl_session * sctx,
 void * mech_data)
```

### 5.63.2.2 `__gsasl_openid20_client_start()`

```
int __gsasl_openid20_client_start (
 Gsasl_session * sctx,
 void ** mech_data)
```

### 5.63.2.3 `__gsasl_openid20_client_step()`

```
int __gsasl_openid20_client_step (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 61 of file openid20/client.c.

### 5.63.2.4 `__gsasl_openid20_server_finish()`

```
void __gsasl_openid20_server_finish (
 Gsasl_session * sctx,
 void * mech_data)
```

### 5.63.2.5 `__gsasl_openid20_server_start()`

```
int __gsasl_openid20_server_start (
 Gsasl_session * sctx,
 void ** mech_data)
```



### 5.63.2.6 `_gsasl_openid20_server_step()`

```
int _gsasl_openid20_server_step (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 59 of file openid20/server.c.

## 5.63.3 Variable Documentation

### 5.63.3.1 `_gsasl_openid20_mechanism`

```
Gsasl_mechanism _gsasl_openid20_mechanism [extern]
```

Definition at line 28 of file openid20/mechinfo.c.

## 5.64 parser.c File Reference

```
#include <config.h>
#include "parser.h"
#include <stdlib.h>
#include <string.h>
#include "validate.h"
```

### Macros

- #define `DEFAULT_CHARSET` "utf-8"
- #define `DEFAULT_ALGORITHM` "md5-sess"

### Enumerations

- enum {
  - `CHALLENGE_REALM = 0`, `CHALLENGE_NONCE`, `CHALLENGE_QOP`, `CHALLENGE_STALE`,
  - `CHALLENGE_MAXBUF`, `CHALLENGE_CHARSET`, `CHALLENGE_ALGORITHM`, `CHALLENGE_CIPHER`
- enum { `QOP_AUTH = 0`, `QOP_AUTH_INT`, `QOP_AUTH_CONF` }
- enum {
  - `CIPHER_DES = 0`, `CIPHER_3DES`, `CIPHER_RC4`, `CIPHER_RC4_40`,
  - `CIPHER_RC4_56`, `CIPHER_AES_CBC` }
- enum {
  - `RESPONSE_USERNAME = 0`, `RESPONSE_REALM`, `RESPONSE_NONCE`, `RESPONSE_CNONCE`,
  - `RESPONSE_NC`, `RESPONSE_QOP`, `RESPONSE_DIGEST_URI`, `RESPONSE_RESPONSE`,
  - `RESPONSE_MAXBUF`, `RESPONSE_CHARSET`, `RESPONSE_CIPHER`, `RESPONSE_AUTHZID` }
- enum { `RESPONSEAUTH_RSPAUTH = 0` }

## Functions

- int [digest\\_md5\\_parse\\_challenge](#) (const char \*challenge, size\_t len, [digest\\_md5\\_challenge](#) \*out)
- int [digest\\_md5\\_parse\\_response](#) (const char \*response, size\_t len, [digest\\_md5\\_response](#) \*out)
- int [digest\\_md5\\_parse\\_finish](#) (const char \*finish, size\_t len, [digest\\_md5\\_finish](#) \*out)

### 5.64.1 Macro Definition Documentation

#### 5.64.1.1 DEFAULT\_ALGORITHM

```
#define DEFAULT_ALGORITHM "md5-sess"
```

Definition at line 38 of file digest-md5/parser.c.

#### 5.64.1.2 DEFAULT\_CHARSET

```
#define DEFAULT_CHARSET "utf-8"
```

Definition at line 37 of file digest-md5/parser.c.

### 5.64.2 Enumeration Type Documentation

#### 5.64.2.1 anonymous enum

```
anonymous enum
```

##### Enumerator

|                     |  |
|---------------------|--|
| CHALLENGE_REALM     |  |
| CHALLENGE_NONCE     |  |
| CHALLENGE_QOP       |  |
| CHALLENGE_STALE     |  |
| CHALLENGE_MAXBUF    |  |
| CHALLENGE_CHARSET   |  |
| CHALLENGE_ALGORITHM |  |
| CHALLENGE_CIPHER    |  |

Definition at line 40 of file digest-md5/parser.c.

### 5.64.2.2 anonymous enum

anonymous enum

#### Enumerator

|               |  |
|---------------|--|
| QOP_AUTH      |  |
| QOP_AUTH_INT  |  |
| QOP_AUTH_CONF |  |

Definition at line 67 of file digest-md5/parser.c.

### 5.64.2.3 anonymous enum

anonymous enum

#### Enumerator

|                |  |
|----------------|--|
| CIPHER_DES     |  |
| CIPHER_3DES    |  |
| CIPHER_RC4     |  |
| CIPHER_RC4_40  |  |
| CIPHER_RC4_56  |  |
| CIPHER_AES_CBC |  |

Definition at line 87 of file digest-md5/parser.c.

### 5.64.2.4 anonymous enum

anonymous enum

#### Enumerator

|                     |  |
|---------------------|--|
| RESPONSE_USERNAME   |  |
| RESPONSE_REALM      |  |
| RESPONSE_NONCE      |  |
| RESPONSE_CNONCE     |  |
| RESPONSE_NC         |  |
| RESPONSE_QOP        |  |
| RESPONSE_DIGEST_URI |  |
| RESPONSE_RESPONSE   |  |
| RESPONSE_MAXBUF     |  |
| RESPONSE_CHARSET    |  |
| RESPONSE_CIPHER     |  |
| RESPONSE_AUTHZID    |  |

Definition at line 313 of file digest-md5/parser.c.

### 5.64.2.5 anonymous enum

anonymous enum

Enumerator

|                      |  |
|----------------------|--|
| RESPONSEAUTH_RSPAUTH |  |
|----------------------|--|

Definition at line 519 of file digest-md5/parser.c.

## 5.64.3 Function Documentation

### 5.64.3.1 digest\_md5\_parse\_challenge()

```
int digest_md5_parse_challenge (
 const char * challenge,
 size_t len,
 digest_md5_challenge * out)
```

Definition at line 567 of file digest-md5/parser.c.

### 5.64.3.2 digest\_md5\_parse\_finish()

```
int digest_md5_parse_finish (
 const char * finish,
 size_t len,
 digest_md5_finish * out)
```

Definition at line 601 of file digest-md5/parser.c.

### 5.64.3.3 digest\_md5\_parse\_response()

```
int digest_md5_parse_response (
 const char * response,
 size_t len,
 digest_md5_response * out)
```

Definition at line 584 of file digest-md5/parser.c.

## 5.65 parser.c File Reference

```
#include <config.h>
#include "parser.h"
#include <stdlib.h>
#include <string.h>
#include "validate.h"
#include "c-ctype.h"
```

### Functions

- int [scram\\_parse\\_client\\_first](#) (const char \*str, size\_t len, struct [scram\\_client\\_first](#) \*cf)
- int [scram\\_parse\\_server\\_first](#) (const char \*str, size\_t len, struct [scram\\_server\\_first](#) \*sf)
- int [scram\\_parse\\_client\\_final](#) (const char \*str, size\_t len, struct [scram\\_client\\_final](#) \*cl)
- int [scram\\_parse\\_server\\_final](#) (const char \*str, size\_t len, struct [scram\\_server\\_final](#) \*sl)

### 5.65.1 Function Documentation

#### 5.65.1.1 [scram\\_parse\\_client\\_final\(\)](#)

```
int scram_parse_client_final (
 const char * str,
 size_t len,
 struct scram_client_final * cl)
```

Definition at line 329 of file `scram/parser.c`.

#### 5.65.1.2 [scram\\_parse\\_client\\_first\(\)](#)

```
int scram_parse_client_first (
 const char * str,
 size_t len,
 struct scram_client_first * cf)
```

Definition at line 76 of file `scram/parser.c`.

#### 5.65.1.3 [scram\\_parse\\_server\\_final\(\)](#)

```
int scram_parse_server_final (
 const char * str,
 size_t len,
 struct scram_server_final * sl)
```

Definition at line 459 of file `scram/parser.c`.

#### 5.65.1.4 `scram_parse_server_first()`

```
int scram_parse_server_first (
 const char * str,
 size_t len,
 struct scram_server_first * sf)
```

Definition at line 218 of file `scram/parser.c`.

## 5.66 `parser.h` File Reference

```
#include "tokens.h"
```

### Functions

- int `digest_md5_getsubopt` (char \*\**optionp*, const char \**const* \**tokens*, char \*\**valuep*)
- int `digest_md5_parse_challenge` (const char \**challenge*, size\_t *len*, `digest_md5_challenge` \**out*)
- int `digest_md5_parse_response` (const char \**response*, size\_t *len*, `digest_md5_response` \**out*)
- int `digest_md5_parse_finish` (const char \**finish*, size\_t *len*, `digest_md5_finish` \**out*)

### 5.66.1 Function Documentation

#### 5.66.1.1 `digest_md5_getsubopt()`

```
int digest_md5_getsubopt (
 char ** optionp,
 const char *const * tokens,
 char ** valuep)
```

Definition at line 44 of file `getsubopt.c`.

#### 5.66.1.2 `digest_md5_parse_challenge()`

```
int digest_md5_parse_challenge (
 const char * challenge,
 size_t len,
 digest_md5_challenge * out)
```

Definition at line 567 of file `digest-md5/parser.c`.

### 5.66.1.3 `digest_md5_parse_finish()`

```
int digest_md5_parse_finish (
 const char * finish,
 size_t len,
 digest_md5_finish * out)
```

Definition at line 601 of file `digest-md5/parser.c`.

### 5.66.1.4 `digest_md5_parse_response()`

```
int digest_md5_parse_response (
 const char * response,
 size_t len,
 digest_md5_response * out)
```

Definition at line 584 of file `digest-md5/parser.c`.

## 5.67 parser.h File Reference

```
#include "tokens.h"
```

### Functions

- int [scram\\_parse\\_client\\_first](#) (const char \*str, size\_t len, struct [scram\\_client\\_first](#) \*cf)
- int [scram\\_parse\\_server\\_first](#) (const char \*str, size\_t len, struct [scram\\_server\\_first](#) \*cf)
- int [scram\\_parse\\_client\\_final](#) (const char \*str, size\_t len, struct [scram\\_client\\_final](#) \*cl)
- int [scram\\_parse\\_server\\_final](#) (const char \*str, size\_t len, struct [scram\\_server\\_final](#) \*sl)

### 5.67.1 Function Documentation

#### 5.67.1.1 `scram_parse_client_final()`

```
int scram_parse_client_final (
 const char * str,
 size_t len,
 struct scram_client_final * cl)
```

Definition at line 329 of file `scram/parser.c`.

### 5.67.1.2 `scram_parse_client_first()`

```
int scram_parse_client_first (
 const char * str,
 size_t len,
 struct scram_client_first * cf)
```

Definition at line 76 of file `scram/parser.c`.

### 5.67.1.3 `scram_parse_server_final()`

```
int scram_parse_server_final (
 const char * str,
 size_t len,
 struct scram_server_final * sl)
```

Definition at line 459 of file `scram/parser.c`.

### 5.67.1.4 `scram_parse_server_first()`

```
int scram_parse_server_first (
 const char * str,
 size_t len,
 struct scram_server_first * cf)
```

Definition at line 218 of file `scram/parser.c`.

## 5.68 `plain.h` File Reference

```
#include <gsasl.h>
```

### Macros

- `#define GSASL_PLAIN_NAME "PLAIN"`

### Functions

- `int _gsasl_plain_client_step` (`Gsasl_session` \*sctx, void \*mech\_data, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len)
- `int _gsasl_plain_server_step` (`Gsasl_session` \*sctx, void \*mech\_data, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len)



## Variables

- [Gsasl\\_mechanism\\_gsasl\\_plain\\_mechanism](#)

## 5.68.1 Macro Definition Documentation

### 5.68.1.1 GSASL\_PLAIN\_NAME

```
#define GSASL_PLAIN_NAME "PLAIN"
```

Definition at line 28 of file plain.h.

## 5.68.2 Function Documentation

### 5.68.2.1 \_\_gsasl\_plain\_client\_step()

```
int __gsasl_plain_client_step (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

### 5.68.2.2 \_\_gsasl\_plain\_server\_step()

```
int __gsasl_plain_server_step (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

## 5.68.3 Variable Documentation

### 5.68.3.1 `_gsasl_plain_mechanism`

`Gsasl_mechanism` `_gsasl_plain_mechanism` [extern]

Definition at line 28 of file `plain/mechinfo.c`.

## 5.69 printer.c File Reference

```
#include <config.h>
#include "printer.h"
#include <stdlib.h>
#include <stdio.h>
#include "validate.h"
```

### Functions

- char \* `digest_md5_print_challenge` (`digest_md5_challenge` \*c)
- char \* `digest_md5_print_response` (`digest_md5_response` \*r)
- char \* `digest_md5_print_finish` (`digest_md5_finish` \*finish)

### 5.69.1 Function Documentation

#### 5.69.1.1 `digest_md5_print_challenge()`

```
char* digest_md5_print_challenge (
 digest_md5_challenge * c)
```

Definition at line 72 of file `digest-md5/printer.c`.

#### 5.69.1.2 `digest_md5_print_finish()`

```
char* digest_md5_print_finish (
 digest_md5_finish * finish)
```

Definition at line 385 of file `digest-md5/printer.c`.

#### 5.69.1.3 `digest_md5_print_response()`

```
char* digest_md5_print_response (
 digest_md5_response * r)
```

Definition at line 241 of file `digest-md5/printer.c`.

## 5.70 printer.c File Reference

```
#include <config.h>
#include "printer.h"
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "validate.h"
```

### Functions

- int [scram\\_print\\_client\\_first](#) (struct [scram\\_client\\_first](#) \*cf, char \*\*out)
- int [scram\\_print\\_server\\_first](#) (struct [scram\\_server\\_first](#) \*sf, char \*\*out)
- int [scram\\_print\\_client\\_final](#) (struct [scram\\_client\\_final](#) \*cl, char \*\*out)
- int [scram\\_print\\_server\\_final](#) (struct [scram\\_server\\_final](#) \*sl, char \*\*out)

### 5.70.1 Function Documentation

#### 5.70.1.1 [scram\\_print\\_client\\_final\(\)](#)

```
int scram_print_client_final (
 struct scram_client_final * cl,
 char ** out)
```

Definition at line 145 of file `scram/printer.c`.

#### 5.70.1.2 [scram\\_print\\_client\\_first\(\)](#)

```
int scram_print_client_first (
 struct scram_client_first * cf,
 char ** out)
```

Definition at line 77 of file `scram/printer.c`.

#### 5.70.1.3 [scram\\_print\\_server\\_final\(\)](#)

```
int scram_print_server_final (
 struct scram_server_final * sl,
 char ** out)
```

Definition at line 165 of file `scram/printer.c`.

#### 5.70.1.4 `scram_print_server_first()`

```
int scram_print_server_first (
 struct scram_server_first * sf,
 char ** out)
```

Definition at line 124 of file `scram/printer.c`.

## 5.71 `printer.h` File Reference

```
#include "tokens.h"
```

### Functions

- char \* `digest_md5_print_challenge` (`digest_md5_challenge` \*challenge)
- char \* `digest_md5_print_response` (`digest_md5_response` \*response)
- char \* `digest_md5_print_finish` (`digest_md5_finish` \*out)

### 5.71.1 Function Documentation

#### 5.71.1.1 `digest_md5_print_challenge()`

```
char* digest_md5_print_challenge (
 digest_md5_challenge * challenge)
```

Definition at line 72 of file `digest-md5/printer.c`.

#### 5.71.1.2 `digest_md5_print_finish()`

```
char* digest_md5_print_finish (
 digest_md5_finish * out)
```

Definition at line 385 of file `digest-md5/printer.c`.

#### 5.71.1.3 `digest_md5_print_response()`

```
char* digest_md5_print_response (
 digest_md5_response * response)
```

Definition at line 241 of file `digest-md5/printer.c`.

## 5.72 printer.h File Reference

```
#include "tokens.h"
```

### Functions

- int [scram\\_print\\_client\\_first](#) (struct [scram\\_client\\_first](#) \*cf, char \*\*out)
- int [scram\\_print\\_server\\_first](#) (struct [scram\\_server\\_first](#) \*cf, char \*\*out)
- int [scram\\_print\\_client\\_final](#) (struct [scram\\_client\\_final](#) \*cl, char \*\*out)
- int [scram\\_print\\_server\\_final](#) (struct [scram\\_server\\_final](#) \*sl, char \*\*out)

### 5.72.1 Function Documentation

#### 5.72.1.1 [scram\\_print\\_client\\_final\(\)](#)

```
int scram_print_client_final (
 struct scram_client_final * cl,
 char ** out)
```

Definition at line 145 of file `scram/printer.c`.

#### 5.72.1.2 [scram\\_print\\_client\\_first\(\)](#)

```
int scram_print_client_first (
 struct scram_client_first * cf,
 char ** out)
```

Definition at line 77 of file `scram/printer.c`.

#### 5.72.1.3 [scram\\_print\\_server\\_final\(\)](#)

```
int scram_print_server_final (
 struct scram_server_final * sl,
 char ** out)
```

Definition at line 165 of file `scram/printer.c`.

### 5.72.1.4 `scram_print_server_first()`

```
int scram_print_server_first (
 struct scram_server_first * cf,
 char ** out)
```

Definition at line 124 of file `scram/printer.c`.

## 5.73 `property.c` File Reference

```
#include <config.h>
#include "internal.h"
```

### Functions

- void `gsasl_property_free` (`Gsasl_session` \**sctx*, `Gsasl_property` *prop*)
- int `gsasl_property_set` (`Gsasl_session` \**sctx*, `Gsasl_property` *prop*, const char \**data*)
- int `gsasl_property_set_raw` (`Gsasl_session` \**sctx*, `Gsasl_property` *prop*, const char \**data*, size\_t *len*)
- const char \* `gsasl_property_fast` (`Gsasl_session` \**sctx*, `Gsasl_property` *prop*)
- const char \* `gsasl_property_get` (`Gsasl_session` \**sctx*, `Gsasl_property` *prop*)

### 5.73.1 Function Documentation

#### 5.73.1.1 `gsasl_property_fast()`

```
const char* gsasl_property_fast (
 Gsasl_session * sctx,
 Gsasl_property prop)
```

`gsasl_property_fast`:

#### Parameters

|             |                                                                                             |
|-------------|---------------------------------------------------------------------------------------------|
| <i>sctx</i> | session handle.                                                                             |
| <i>prop</i> | enumerated value of <code>Gsasl_property</code> type, indicating the type of data in @data. |

Retrieve the data stored in the session handle for given property @*prop*.

The pointer is to live data, and must not be deallocated or modified in any way.

This function will not invoke the application callback.

Return value: Return property value, if known, or NULL if no value known.

Since: 0.2.0

Definition at line 262 of file `property.c`.

### 5.73.1.2 gsasl\_property\_free()

```
void gsasl_property_free (
 Gsasl_session * sctx,
 Gsasl_property prop)
```

gsasl\_property\_free:

#### Parameters

|             |                                                  |
|-------------|--------------------------------------------------|
| <i>sctx</i> | session handle.                                  |
| <i>prop</i> | enumerated value of Gsasl_property type to clear |

Deallocate associated data with property @prop in session handle. After this call, gsasl\_property\_fast(@sctx, @prop) will always return NULL.

Since: 2.0.0

Definition at line 159 of file property.c.

### 5.73.1.3 gsasl\_property\_get()

```
const char* gsasl_property_get (
 Gsasl_session * sctx,
 Gsasl_property prop)
```

gsasl\_property\_get:

#### Parameters

|             |                                                                                |
|-------------|--------------------------------------------------------------------------------|
| <i>sctx</i> | session handle.                                                                |
| <i>prop</i> | enumerated value of Gsasl_property type, indicating the type of data in @data. |

Retrieve the data stored in the session handle for given property @prop, possibly invoking the application callback to get the value.

The pointer is to live data, and must not be deallocated or modified in any way.

This function will invoke the application callback, using [gsasl\\_callback\(\)](#), when a property value is not known.

Return value: Return data for property, or NULL if no value known.

Since: 0.2.0

Definition at line 292 of file property.c.

#### 5.73.1.4 `gsasl_property_set()`

```
int gsasl_property_set (
 Gsasl_session * sctx,
 Gsasl_property prop,
 const char * data)
```

`gsasl_property_set`:

##### Parameters

|             |                                                                                                           |
|-------------|-----------------------------------------------------------------------------------------------------------|
| <i>sctx</i> | session handle.                                                                                           |
| <i>prop</i> | enumerated value of <code>Gsasl_property</code> type, indicating the type of data in <code>@data</code> . |
| <i>data</i> | zero terminated character string to store.                                                                |

Make a copy of `@data` and store it in the session handle for the indicated property `@prop`.

You can immediately deallocate `@data` after calling this function, without affecting the data stored in the session handle.

Return value: `GSASL_OK` iff successful, otherwise `GSASL_MALLOC_ERROR`.

Since: 0.2.0

Definition at line 189 of file `property.c`.

#### 5.73.1.5 `gsasl_property_set_raw()`

```
int gsasl_property_set_raw (
 Gsasl_session * sctx,
 Gsasl_property prop,
 const char * data,
 size_t len)
```

`gsasl_property_set_raw`:

##### Parameters

|             |                                                                                                           |
|-------------|-----------------------------------------------------------------------------------------------------------|
| <i>sctx</i> | session handle.                                                                                           |
| <i>prop</i> | enumerated value of <code>Gsasl_property</code> type, indicating the type of data in <code>@data</code> . |
| <i>data</i> | character string to store.                                                                                |
| <i>len</i>  | length of character string to store.                                                                      |

Make a copy of `@len` sized `@data` and store a zero terminated version of it in the session handle for the indicated property `@prop`.

You can immediately deallocate `@data` after calling this function, without affecting the data stored in the session handle.

Except for the length indicator, this function is identical to `gsasl_property_set`.



Return value: GSASL\_OK iff successful, otherwise GSASL\_MALLOC\_ERROR.

Since: 0.2.0

Definition at line 218 of file property.c.

## 5.74 qop.c File Reference

```
#include <config.h>
#include "qop.h"
#include "tokens.h"
#include "parser.h"
#include <string.h>
#include <stdlib.h>
```

### Functions

- int [digest\\_md5\\_qopstr2qops](#) (const char \*qopstr)
- const char \* [digest\\_md5\\_qops2qopstr](#) (int qops)

### 5.74.1 Function Documentation

#### 5.74.1.1 [digest\\_md5\\_qops2qopstr\(\)](#)

```
const char* digest_md5_qops2qopstr (
 int qops)
```

Definition at line 90 of file qop.c.

#### 5.74.1.2 [digest\\_md5\\_qopstr2qops\(\)](#)

```
int digest_md5_qopstr2qops (
 const char * qopstr)
```

Definition at line 35 of file qop.c.

## 5.75 qop.h File Reference

### Functions

- int [digest\\_md5\\_qopstr2qops](#) (const char \*qopstr)
- const char \* [digest\\_md5\\_qops2qopstr](#) (int qops)

## 5.75.1 Function Documentation

### 5.75.1.1 digest\_md5\_qops2qopstr()

```
const char* digest_md5_qops2qopstr (
 int qops)
```

Definition at line 90 of file qop.c.

### 5.75.1.2 digest\_md5\_qopstr2qops()

```
int digest_md5_qopstr2qops (
 const char * qopstr)
```

Definition at line 35 of file qop.c.

## 5.76 register.c File Reference

```
#include <config.h>
#include "internal.h"
```

### Functions

- int [gsasl\\_register](#) ([Gsasl](#) \*ctx, const [Gsasl\\_mechanism](#) \*mech)

## 5.76.1 Function Documentation

### 5.76.1.1 gsasl\_register()

```
int gsasl_register (
 Gsasl * ctx,
 const Gsasl_mechanism * mech)
```

gsasl\_register:

#### Parameters

|             |                                                 |
|-------------|-------------------------------------------------|
| <i>ctx</i>  | pointer to libgsasl handle.                     |
| <i>mech</i> | plugin structure with information about plugin. |

This function initialize given mechanism, and if successful, add it to the list of plugins that is used by the library.

Return value: GSASL\_OK iff successful, otherwise GSASL\_MALLOC\_ERROR.

Since: 0.2.0

Definition at line 39 of file register.c.

## 5.77 saml20.h File Reference

```
#include <gsasl.h>
```

### Macros

- #define [GSASL\\_SAML20\\_NAME](#) "SAML20"

### Functions

- int [\\_\\_gsasl\\_saml20\\_client\\_start](#) ([Gsasl\\_session](#) \*sctx, void \*\*mech\_data)
- int [\\_\\_gsasl\\_saml20\\_client\\_step](#) ([Gsasl\\_session](#) \*sctx, void \*mech\_data, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len)
- void [\\_\\_gsasl\\_saml20\\_client\\_finish](#) ([Gsasl\\_session](#) \*sctx, void \*mech\_data)
- int [\\_\\_gsasl\\_saml20\\_server\\_start](#) ([Gsasl\\_session](#) \*sctx, void \*\*mech\_data)
- int [\\_\\_gsasl\\_saml20\\_server\\_step](#) ([Gsasl\\_session](#) \*sctx, void \*mech\_data, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len)
- void [\\_\\_gsasl\\_saml20\\_server\\_finish](#) ([Gsasl\\_session](#) \*sctx, void \*mech\_data)

### Variables

- [Gsasl\\_mechanism\\_gsasl\\_saml20\\_mechanism](#)

## 5.77.1 Macro Definition Documentation

### 5.77.1.1 GSASL\_SAML20\_NAME

```
#define GSASL_SAML20_NAME "SAML20"
```

Definition at line 28 of file saml20.h.

## 5.77.2 Function Documentation

### 5.77.2.1 `_gsasl_saml20_client_finish()`

```
void _gsasl_saml20_client_finish (
 Gsasl_session * sctx,
 void * mech_data)
```

### 5.77.2.2 `_gsasl_saml20_client_start()`

```
int _gsasl_saml20_client_start (
 Gsasl_session * sctx,
 void ** mech_data)
```

### 5.77.2.3 `_gsasl_saml20_client_step()`

```
int _gsasl_saml20_client_step (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 60 of file `saml20/client.c`.

### 5.77.2.4 `_gsasl_saml20_server_finish()`

```
void _gsasl_saml20_server_finish (
 Gsasl_session * sctx,
 void * mech_data)
```

### 5.77.2.5 `_gsasl_saml20_server_start()`

```
int _gsasl_saml20_server_start (
 Gsasl_session * sctx,
 void ** mech_data)
```

### 5.77.2.6 `_gsasl_saml20_server_step()`

```
int _gsasl_saml20_server_step (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 57 of file saml20/server.c.

## 5.77.3 Variable Documentation

### 5.77.3.1 `_gsasl_saml20_mechanism`

```
Gsasl_mechanism _gsasl_saml20_mechanism [extern]
```

Definition at line 28 of file saml20/mechinfo.c.

## 5.78 saslprep.c File Reference

```
#include <config.h>
#include "internal.h"
```

### Functions

- int `gsasl_saslprep` (const char \*in, `Gsasl_saslprep_flags` flags `_GL_UNUSED`, char \*\*out, int \*stringpreprc `_GL_UNUSED`)

### 5.78.1 Function Documentation

#### 5.78.1.1 `gsasl_saslprep()`

```
int gsasl_saslprep (
 const char * in,
 Gsasl_saslprep_flags flags _GL_UNUSED,
 char ** out,
 int *stringpreprc _GL_UNUSED)
```

Definition at line 87 of file saslprep.c.

## 5.79 scram.h File Reference

```
#include <gsasl.h>
```

## 5.80 securid.h File Reference

```
#include <gsasl.h>
```

### Macros

- `#define GSASL_SECURID_NAME "SECURID"`

### Functions

- `int _gsasl_securig_client_start (Gsasl_session *sctx, void **mech_data)`
- `int _gsasl_securig_client_step (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)`
- `void _gsasl_securig_client_finish (Gsasl_session *sctx, void *mech_data)`
- `int _gsasl_securig_server_step (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)`

### Variables

- `Gsasl_mechanism_gsasl_securig_mechanism`

### 5.80.1 Macro Definition Documentation

#### 5.80.1.1 GSASL\_SECURID\_NAME

```
#define GSASL_SECURID_NAME "SECURID"
```

Definition at line 28 of file securid.h.

### 5.80.2 Function Documentation

### 5.80.2.1 `_gsasl_secured_client_finish()`

```
void _gsasl_secured_client_finish (
 Gsasl_session * sctx,
 void * mech_data)
```

### 5.80.2.2 `_gsasl_secured_client_start()`

```
int _gsasl_secured_client_start (
 Gsasl_session * sctx,
 void ** mech_data)
```

### 5.80.2.3 `_gsasl_secured_client_step()`

```
int _gsasl_secured_client_step (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 54 of file securid/client.c.

### 5.80.2.4 `_gsasl_secured_server_step()`

```
int _gsasl_secured_server_step (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

## 5.80.3 Variable Documentation

### 5.80.3.1 `_gsasl_secured_mechanism`

`Gsasl_mechanism` `_gsasl_secured_mechanism` [extern]

Definition at line 28 of file securid/mechinfo.c.

## 5.81 server.c File Reference

```
#include <config.h>
#include "anonymous.h"
```

### Functions

- int [\\_gsasl\\_anonymous\\_server\\_step](#) ([Gsasl\\_session](#) \*sctx, void \*mech\_data \_GL\_UNUSED, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len)

### 5.81.1 Function Documentation

#### 5.81.1.1 [\\_gsasl\\_anonymous\\_server\\_step\(\)](#)

```
int _gsasl_anonymous_server_step (
 Gsasl_session * sctx,
 void *mech_data _GL_UNUSED,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 29 of file anonymous/server.c.

## 5.82 server.c File Reference

```
#include <config.h>
#include "cram-md5.h"
#include <stdlib.h>
#include <string.h>
#include "challenge.h"
#include "digest.h"
```

### Macros

- #define [MD5LEN](#) 16

### Functions

- int [\\_gsasl\\_cram\\_md5\\_server\\_start](#) ([Gsasl\\_session](#) \*sctx \_GL\_UNUSED, void \*\*mech\_data)
- int [\\_gsasl\\_cram\\_md5\\_server\\_step](#) ([Gsasl\\_session](#) \*sctx, void \*mech\_data, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len)
- void [\\_gsasl\\_cram\\_md5\\_server\\_finish](#) ([Gsasl\\_session](#) \*sctx \_GL\_UNUSED, void \*mech\_data)



## 5.82.1 Macro Definition Documentation

### 5.82.1.1 MD5LEN

```
#define MD5LEN 16
```

Definition at line 40 of file cram-md5/server.c.

## 5.82.2 Function Documentation

### 5.82.2.1 `_gsasl_cram_md5_server_finish()`

```
void _gsasl_cram_md5_server_finish (
 Gsasl_session *sctx _GL_UNUSED,
 void * mech_data)
```

Definition at line 130 of file cram-md5/server.c.

### 5.82.2.2 `_gsasl_cram_md5_server_start()`

```
int _gsasl_cram_md5_server_start (
 Gsasl_session *sctx _GL_UNUSED,
 void ** mech_data)
```

Definition at line 43 of file cram-md5/server.c.

### 5.82.2.3 `_gsasl_cram_md5_server_step()`

```
int _gsasl_cram_md5_server_step (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 66 of file cram-md5/server.c.

## 5.83 server.c File Reference

```
#include <config.h>
#include "digest-md5.h"
#include <stdlib.h>
#include <string.h>
#include "gc.h"
#include "nonascii.h"
#include "tokens.h"
#include "parser.h"
#include "printer.h"
#include "free.h"
#include "session.h"
#include "digesthmac.h"
#include "validate.h"
#include "qop.h"
#include "mechtools.h"
```

### Data Structures

- struct [\\_Gsasl\\_digest\\_md5\\_server\\_state](#)

### Macros

- #define [NONCE\\_ENTROPY\\_BYTES](#) 16

### Typedefs

- typedef struct [\\_Gsasl\\_digest\\_md5\\_server\\_state](#) [\\_Gsasl\\_digest\\_md5\\_server\\_state](#)

### Functions

- int [\\_gsasl\\_digest\\_md5\\_server\\_start](#) ([Gsasl\\_session](#) \*sctx [\\_GL\\_UNUSED](#), void \*\*mech\_data)
- int [\\_gsasl\\_digest\\_md5\\_server\\_step](#) ([Gsasl\\_session](#) \*sctx, void \*mech\_data, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len)
- void [\\_gsasl\\_digest\\_md5\\_server\\_finish](#) ([Gsasl\\_session](#) \*sctx [\\_GL\\_UNUSED](#), void \*mech\_data)
- int [\\_gsasl\\_digest\\_md5\\_server\\_encode](#) ([Gsasl\\_session](#) \*sctx [\\_GL\\_UNUSED](#), void \*mech\_data, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len)
- int [\\_gsasl\\_digest\\_md5\\_server\\_decode](#) ([Gsasl\\_session](#) \*sctx [\\_GL\\_UNUSED](#), void \*mech\_data, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len)

#### 5.83.1 Macro Definition Documentation

### 5.83.1.1 NONCE\_ENTROPY\_BYTES

```
#define NONCE_ENTROPY_BYTES 16
```

Definition at line 49 of file digest-md5/server.c.

## 5.83.2 Typedef Documentation

### 5.83.2.1 \_Gsasl\_digest\_md5\_server\_state

```
typedef struct _Gsasl_digest_md5_server_state _Gsasl_digest_md5_server_state
```

Definition at line 1 of file digest-md5/server.c.

## 5.83.3 Function Documentation

### 5.83.3.1 \_\_gsasl\_digest\_md5\_server\_decode()

```
int __gsasl_digest_md5_server_decode (
 Gsasl_session *sctx _GL_UNUSED,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 389 of file digest-md5/server.c.

### 5.83.3.2 \_\_gsasl\_digest\_md5\_server\_encode()

```
int __gsasl_digest_md5_server_encode (
 Gsasl_session *sctx _GL_UNUSED,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 365 of file digest-md5/server.c.

### 5.83.3.3 `_gsasl_digest_md5_server_finish()`

```
void _gsasl_digest_md5_server_finish (
 Gsasl_session *sctx _GL_UNUSED,
 void * mech_data)
```

Definition at line 349 of file digest-md5/server.c.

### 5.83.3.4 `_gsasl_digest_md5_server_start()`

```
int _gsasl_digest_md5_server_start (
 Gsasl_session *sctx _GL_UNUSED,
 void ** mech_data)
```

Definition at line 67 of file digest-md5/server.c.

### 5.83.3.5 `_gsasl_digest_md5_server_step()`

```
int _gsasl_digest_md5_server_step (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 146 of file digest-md5/server.c.

## 5.84 server.c File Reference

```
#include <config.h>
#include "external.h"
#include <string.h>
```

### Functions

- `int _gsasl_external_server_step(Gsasl_session *sctx, void *mech_data _GL_UNUSED, const char *input, size_t input_len, char **output, size_t *output_len)`

#### 5.84.1 Function Documentation

### 5.84.1.1 `_gsasl_external_server_step()`

```
int _gsasl_external_server_step (
 Gsasl_session * sctx,
 void *mech_data _GL_UNUSED,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 32 of file external/server.c.

## 5.85 server.c File Reference

```
#include <config.h>
#include "gs2.h"
#include <stdlib.h>
#include <string.h>
#include <attribute.h>
#include "gss-extra.h"
#include "gs2helper.h"
#include "mechtools.h"
```

### Data Structures

- struct [\\_Gsasl\\_gs2\\_server\\_state](#)

### Typedefs

- typedef struct [\\_Gsasl\\_gs2\\_server\\_state](#) [\\_Gsasl\\_gs2\\_server\\_state](#)

### Functions

- int [\\_gsasl\\_gs2\\_server\\_start](#) ([Gsasl\\_session](#) \*sctx, void \*\*mech\_data)
- int [\\_gsasl\\_gs2\\_server\\_step](#) ([Gsasl\\_session](#) \*sctx, void \*mech\_data, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len)
- void [\\_gsasl\\_gs2\\_server\\_finish](#) ([Gsasl\\_session](#) \*sctx, void \*mech\_data)

## 5.85.1 Typedef Documentation

### 5.85.1.1 `_Gsasl_gs2_server_state`

```
typedef struct _Gsasl_gs2_server_state _Gsasl_gs2_server_state
```

Definition at line 1 of file gs2/server.c.

## 5.85.2 Function Documentation

### 5.85.2.1 `__gsasl_gs2_server_finish()`

```
void __gsasl_gs2_server_finish (
 Gsasl_session * sctx,
 void * mech_data)
```

Definition at line 299 of file gs2/server.c.

### 5.85.2.2 `__gsasl_gs2_server_start()`

```
int __gsasl_gs2_server_start (
 Gsasl_session * sctx,
 void ** mech_data)
```

Definition at line 119 of file gs2/server.c.

### 5.85.2.3 `__gsasl_gs2_server_step()`

```
int __gsasl_gs2_server_step (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 162 of file gs2/server.c.

## 5.86 server.c File Reference

```
#include <config.h>
#include <stdlib.h>
#include <string.h>
#include "x-gssapi.h"
#include "gss-extra.h"
```

### Data Structures

- struct [\\_Gsasl\\_gssapi\\_server\\_state](#)

## Typedefs

- typedef struct [\\_Gssapi\\_server\\_state](#) [\\_Gssapi\\_server\\_state](#)

## Functions

- int [\\_gssapi\\_server\\_start](#) ([Gssapi\\_session](#) \*sctx, void \*\*mech\_data)
- int [\\_gssapi\\_server\\_step](#) ([Gssapi\\_session](#) \*sctx, void \*mech\_data, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len)
- void [\\_gssapi\\_server\\_finish](#) ([Gssapi\\_session](#) \*sctx, void \*mech\_data)

### 5.86.1 Typedef Documentation

#### 5.86.1.1 [\\_Gssapi\\_server\\_state](#)

```
typedef struct _Gssapi_server_state _Gssapi_server_state
```

Definition at line 1 of file gssapi/server.c.

### 5.86.2 Function Documentation

#### 5.86.2.1 [\\_gssapi\\_server\\_finish\(\)](#)

```
void _gssapi_server_finish (
 Gssapi_session * sctx,
 void * mech_data)
```

Definition at line 266 of file gssapi/server.c.

#### 5.86.2.2 [\\_gssapi\\_server\\_start\(\)](#)

```
int _gssapi_server_start (
 Gssapi_session * sctx,
 void ** mech_data)
```

Definition at line 47 of file gssapi/server.c.

### 5.86.2.3 `_gsasl_gssapi_server_step()`

```
int _gsasl_gssapi_server_step (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 65 of file gssapi/server.c.

## 5.87 server.c File Reference

```
#include <config.h>
#include <stdlib.h>
#include <string.h>
#include "login.h"
```

### Data Structures

- struct [\\_Gsasl\\_login\\_server\\_state](#)

### Macros

- #define [CHALLENGE\\_USERNAME](#) "User Name"
- #define [CHALLENGE\\_PASSWORD](#) "Password"

### Functions

- int [\\_gsasl\\_login\\_server\\_start](#) ([Gsasl\\_session](#) \*sctx, [\\_GL\\_UNUSED](#), void \*\*mech\_data)
- int [\\_gsasl\\_login\\_server\\_step](#) ([Gsasl\\_session](#) \*sctx, void \*mech\_data, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len)
- void [\\_gsasl\\_login\\_server\\_finish](#) ([Gsasl\\_session](#) \*sctx, [\\_GL\\_UNUSED](#), void \*mech\_data)

## 5.87.1 Macro Definition Documentation

### 5.87.1.1 [CHALLENGE\\_PASSWORD](#)

```
#define CHALLENGE_PASSWORD "Password"
```

Definition at line 42 of file login/server.c.



### 5.87.1.2 CHALLENGE\_USERNAME

```
#define CHALLENGE_USERNAME "User Name"
```

Definition at line 41 of file login/server.c.

## 5.87.2 Function Documentation

### 5.87.2.1 \_\_gsasl\_login\_server\_finish()

```
void __gsasl_login_server_finish (
 Gsasl_session *sctx _GL_UNUSED,
 void * mech_data)
```

Definition at line 148 of file login/server.c.

### 5.87.2.2 \_\_gsasl\_login\_server\_start()

```
int __gsasl_login_server_start (
 Gsasl_session *sctx _GL_UNUSED,
 void ** mech_data)
```

Definition at line 45 of file login/server.c.

### 5.87.2.3 \_\_gsasl\_login\_server\_step()

```
int __gsasl_login_server_step (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 59 of file login/server.c.

## 5.88 server.c File Reference

```
#include <config.h>
#include "openid20.h"
#include <string.h>
#include <stdlib.h>
#include "mechtools.h"
```

## Data Structures

- struct [openid20\\_server\\_state](#)

## Functions

- int [\\_gsasl\\_openid20\\_server\\_start](#) ([Gsasl\\_session](#) \*sctx [\\_GL\\_UNUSED](#), void \*\*mech\_data)
- int [\\_gsasl\\_openid20\\_server\\_step](#) ([Gsasl\\_session](#) \*sctx, void \*mech\_data, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len)
- void [\\_gsasl\\_openid20\\_server\\_finish](#) ([Gsasl\\_session](#) \*sctx [\\_GL\\_UNUSED](#), void \*mech\_data)

### 5.88.1 Function Documentation

#### 5.88.1.1 [\\_gsasl\\_openid20\\_server\\_finish\(\)](#)

```
void _gsasl_openid20_server_finish (
 Gsasl_session *sctx _GL_UNUSED,
 void * mech_data)
```

Definition at line 191 of file openid20/server.c.

#### 5.88.1.2 [\\_gsasl\\_openid20\\_server\\_start\(\)](#)

```
int _gsasl_openid20_server_start (
 Gsasl_session *sctx _GL_UNUSED,
 void ** mech_data)
```

Definition at line 44 of file openid20/server.c.

#### 5.88.1.3 [\\_gsasl\\_openid20\\_server\\_step\(\)](#)

```
int _gsasl_openid20_server_step (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 59 of file openid20/server.c.

## 5.89 server.c File Reference

```
#include <config.h>
#include "plain.h"
#include <string.h>
#include <stdlib.h>
```

### Functions

- int [\\_gsasl\\_plain\\_server\\_step](#) ([Gsasl\\_session](#) \*sctx, void \*mech\_data \_GL\_UNUSED, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len)

### 5.89.1 Function Documentation

#### 5.89.1.1 [\\_gsasl\\_plain\\_server\\_step\(\)](#)

```
int _gsasl_plain_server_step (
 Gsasl_session * sctx,
 void *mech_data _GL_UNUSED,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 35 of file plain/server.c.

## 5.90 server.c File Reference

```
#include <config.h>
#include "saml20.h"
#include <string.h>
#include <stdlib.h>
#include "mechtools.h"
```

### Data Structures

- struct [saml20\\_server\\_state](#)

### Functions

- int [\\_gsasl\\_saml20\\_server\\_start](#) ([Gsasl\\_session](#) \*sctx \_GL\_UNUSED, void \*\*mech\_data)
- int [\\_gsasl\\_saml20\\_server\\_step](#) ([Gsasl\\_session](#) \*sctx, void \*mech\_data, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len)
- void [\\_gsasl\\_saml20\\_server\\_finish](#) ([Gsasl\\_session](#) \*sctx \_GL\_UNUSED, void \*mech\_data)

## 5.90.1 Function Documentation

### 5.90.1.1 `__gsasl_saml20_server_finish()`

```
void __gsasl_saml20_server_finish (
 Gsasl_session *sctx _GL_UNUSED,
 void * mech_data)
```

Definition at line 141 of file `saml20/server.c`.

### 5.90.1.2 `__gsasl_saml20_server_start()`

```
int __gsasl_saml20_server_start (
 Gsasl_session *sctx _GL_UNUSED,
 void ** mech_data)
```

Definition at line 43 of file `saml20/server.c`.

### 5.90.1.3 `__gsasl_saml20_server_step()`

```
int __gsasl_saml20_server_step (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 57 of file `saml20/server.c`.

## 5.91 `server.c` File Reference

```
#include <config.h>
#include "scram.h"
#include <stdlib.h>
#include <limits.h>
#include <string.h>
#include "minmax.h"
#include "tokens.h"
#include "parser.h"
#include "printer.h"
#include "gc.h"
#include "memxor.h"
#include "tools.h"
#include "mechtools.h"
```

## Data Structures

- struct [scram\\_server\\_state](#)

## Macros

- #define [DEFAULT\\_SALT\\_BYTES](#) 12
- #define [SNONCE\\_ENTROPY\\_BYTES](#) 18

## Functions

- int [\\_gsasl\\_scram\\_server\\_step](#) ([Gsasl\\_session](#) \*sctx, void \*mech\_data, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len)
- void [\\_gsasl\\_scram\\_server\\_finish](#) ([Gsasl\\_session](#) \*sctx [\\_GL\\_UNUSED](#), void \*mech\_data)

### 5.91.1 Macro Definition Documentation

#### 5.91.1.1 DEFAULT\_SALT\_BYTES

```
#define DEFAULT_SALT_BYTES 12
```

Definition at line 48 of file `scram/server.c`.

#### 5.91.1.2 SNONCE\_ENTROPY\_BYTES

```
#define SNONCE_ENTROPY_BYTES 18
```

Definition at line 49 of file `scram/server.c`.

### 5.91.2 Function Documentation

#### 5.91.2.1 \_gsasl\_scram\_server\_finish()

```
void _gsasl_scram_server_finish (
 Gsasl_session *sctx _GL_UNUSED,
 void * mech_data)
```

Definition at line 581 of file `scram/server.c`.

### 5.91.2.2 `_gsasl_scram_server_step()`

```
int _gsasl_scram_server_step (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 169 of file `scram/server.c`.

## 5.92 server.c File Reference

```
#include <config.h>
#include "securid.h"
#include <stdlib.h>
#include <string.h>
```

### Macros

- `#define PASSCODE "passcode"`
- `#define PIN "pin"`

### Functions

- `int _gsasl_securid_server_step (Gsasl_session *sctx, void *mech_data _GL_UNUSED, const char *input, size_t input_len, char **output, size_t *output_len)`

## 5.92.1 Macro Definition Documentation

### 5.92.1.1 PASSCODE

```
#define PASSCODE "passcode"
```

Definition at line 34 of file `securid/server.c`.

### 5.92.1.2 PIN

```
#define PIN "pin"
```

Definition at line 35 of file `securid/server.c`.

## 5.92.2 Function Documentation

### 5.92.2.1 `_gsasl_securid_server_step()`

```
int _gsasl_securid_server_step (
 Gsasl_session * sctx,
 void *mech_data _GL_UNUSED,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 38 of file securid/server.c.

## 5.93 session.c File Reference

```
#include <config.h>
#include "session.h"
#include <stdlib.h>
#include <string.h>
#include <gc.h>
```

### Macros

- `#define MD5LEN` 16
- `#define SASL_INTEGRITY_PREFIX_LENGTH` 4
- `#define MAC_DATA_LEN` 4
- `#define MAC_HMAC_LEN` 10
- `#define MAC_MSG_TYPE` "\x00\x01"
- `#define MAC_MSG_TYPE_LEN` 2
- `#define MAC_SEQNUM_LEN` 4
- `#define C2I`(buf)

### Functions

- int `digest_md5_encode` (const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len, `digest_md5_qop` qop, unsigned long sendseqnum, char key[`DIGEST_MD5_LENGTH`])
- int `digest_md5_decode` (const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len, `digest_md5_qop` qop, unsigned long readseqnum, char key[`DIGEST_MD5_LENGTH`])

### 5.93.1 Macro Definition Documentation

### 5.93.1.1 C2I

```
#define C2I(
 buf)
```

**Value:**

```
(buf[3] & 0xFF) |
(buf[2] & 0xFF) << 8) |
(buf[1] & 0xFF) << 16) |
(buf[0] & 0xFF) << 24))
```

Definition at line 114 of file session.c.

### 5.93.1.2 MAC\_DATA\_LEN

```
#define MAC_DATA_LEN 4
```

Definition at line 39 of file session.c.

### 5.93.1.3 MAC\_HMAC\_LEN

```
#define MAC_HMAC_LEN 10
```

Definition at line 40 of file session.c.

### 5.93.1.4 MAC\_MSG\_TYPE

```
#define MAC_MSG_TYPE "\x00\x01"
```

Definition at line 41 of file session.c.

### 5.93.1.5 MAC\_MSG\_TYPE\_LEN

```
#define MAC_MSG_TYPE_LEN 2
```

Definition at line 42 of file session.c.



### 5.93.1.6 MAC\_SEQNUM\_LEN

```
#define MAC_SEQNUM_LEN 4
```

Definition at line 43 of file session.c.

### 5.93.1.7 MD5LEN

```
#define MD5LEN 16
```

Definition at line 37 of file session.c.

### 5.93.1.8 SASL\_INTEGRITY\_PREFIX\_LENGTH

```
#define SASL_INTEGRITY_PREFIX_LENGTH 4
```

Definition at line 38 of file session.c.

## 5.93.2 Function Documentation

### 5.93.2.1 digest\_md5\_decode()

```
int digest_md5_decode (
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len,
 digest_md5_qop qop,
 unsigned long readseqnum,
 char key[DIGEST_MD5_LENGTH])
```

Definition at line 120 of file session.c.

### 5.93.2.2 digest\_md5\_encode()

```
int digest_md5_encode (
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len,
 digest_md5_qop qop,
 unsigned long sendseqnum,
 char key[DIGEST_MD5_LENGTH])
```

Definition at line 46 of file session.c.

## 5.94 session.h File Reference

```
#include "tokens.h"
```

### Functions

- int [digest\\_md5\\_encode](#) (const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len, [digest\\_md5\\_qop](#) qop, unsigned long sendseqnum, char key[[DIGEST\\_MD5\\_LENGTH](#)])
- int [digest\\_md5\\_decode](#) (const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len, [digest\\_md5\\_qop](#) qop, unsigned long readseqnum, char key[[DIGEST\\_MD5\\_LENGTH](#)])

### 5.94.1 Function Documentation

#### 5.94.1.1 [digest\\_md5\\_decode\(\)](#)

```
int digest_md5_decode (
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len,
 digest_md5_qop qop,
 unsigned long readseqnum,
 char key[DIGEST_MD5_LENGTH])
```

Definition at line 120 of file session.c.

#### 5.94.1.2 [digest\\_md5\\_encode\(\)](#)

```
int digest_md5_encode (
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len,
 digest_md5_qop qop,
 unsigned long sendseqnum,
 char key[DIGEST_MD5_LENGTH])
```

Definition at line 46 of file session.c.

## 5.95 suggest.c File Reference

```
#include <config.h>
#include "internal.h"
```

## Functions

- int `gsasl_mechanism_name_p` (const char \*mech)
- const char \* `gsasl_client_suggest_mechanism` (Gsasl \*ctx, const char \*mechlist)

## Variables

- const char \* `_GSASL_VALID_MECHANISM_CHARACTERS`

### 5.95.1 Function Documentation

#### 5.95.1.1 `gsasl_client_suggest_mechanism()`

```
const char* gsasl_client_suggest_mechanism (
 Gsasl * ctx,
 const char * mechlist)
```

`gsasl_client_suggest_mechanism`:

##### Parameters

|                 |                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------|
| <i>ctx</i>      | libgsasl handle.                                                                             |
| <i>mechlist</i> | input character array with SASL mechanism names, separated by invalid characters (e.g. SPC). |

Given a list of mechanisms, suggest which to use.

Return value: Returns name of "best" SASL mechanism supported by the libgsasl client which is present in the input string, or NULL if no supported mechanism is found.

Definition at line 88 of file suggest.c.

#### 5.95.1.2 `gsasl_mechanism_name_p()`

```
int gsasl_mechanism_name_p (
 const char * mech)
```

`gsasl_mechanism_name_p`:

##### Parameters

|             |                                            |
|-------------|--------------------------------------------|
| <i>mech</i> | input variable with mechanism name string. |
|-------------|--------------------------------------------|

Check if the mechanism name string @mech follows syntactical rules. It does not check that the name is registered with IANA. It does not check that the mechanism name is actually implemented and supported.

SASL mechanisms are named by strings, from 1 to 20 characters in length, consisting of upper-case letters, digits, hyphens, and/or underscores.

Returns: non-zero when mechanism name string @mech conforms to rules, zero when it does not meet the requirements.

Since: 2.0.0

Definition at line 53 of file suggest.c.

## 5.95.2 Variable Documentation

### 5.95.2.1 \_GSASL\_VALID\_MECHANISM\_CHARACTERS

```
const char* _GSASL_VALID_MECHANISM_CHARACTERS
```

**Initial value:**

```
=
"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-_"
```

Definition at line 32 of file suggest.c.

## 5.96 supportp.c File Reference

```
#include <config.h>
#include "internal.h"
```

### Functions

- int [gsasl\\_client\\_support\\_p](#)(Gsasl \*ctx, const char \*name)
- int [gsasl\\_server\\_support\\_p](#)(Gsasl \*ctx, const char \*name)

### 5.96.1 Function Documentation

#### 5.96.1.1 gsasl\_client\_support\_p()

```
int gsasl_client_support_p (
 Gsasl * ctx,
 const char * name)
```

gsasl\_client\_support\_p:

## Parameters

|             |                         |
|-------------|-------------------------|
| <i>ctx</i>  | libgsasl handle.        |
| <i>name</i> | name of SASL mechanism. |

Decide whether there is client-side support for a specified mechanism.

Return value: Returns 1 if the libgsasl client supports the named mechanism, otherwise 0.

Definition at line 50 of file supportp.c.

### 5.96.1.2 gsasl\_server\_support\_p()

```
int gsasl_server_support_p (
 Gsasl * ctx,
 const char * name)
```

gsasl\_server\_support\_p:

## Parameters

|             |                         |
|-------------|-------------------------|
| <i>ctx</i>  | libgsasl handle.        |
| <i>name</i> | name of SASL mechanism. |

Decide whether there is server-side support for a specified mechanism.

Return value: Returns 1 if the libgsasl server supports the named mechanism, otherwise 0.

Definition at line 67 of file supportp.c.

## 5.97 test-parser.c File Reference

```
#include <config.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "free.h"
#include "parser.h"
#include "printer.h"
#include "digesthmac.h"
#include "gc.h"
```

### Functions

- int [main](#) (void)

## 5.97.1 Function Documentation

### 5.97.1.1 main()

```
int main (
 void)
```

Definition at line 37 of file test-parser.c.

## 5.98 tokens.c File Reference

```
#include <config.h>
#include "tokens.h"
#include <stdlib.h>
#include <string.h>
```

### Functions

- void [scram\\_free\\_client\\_first](#) (struct [scram\\_client\\_first](#) \*cf)
- void [scram\\_free\\_server\\_first](#) (struct [scram\\_server\\_first](#) \*sf)
- void [scram\\_free\\_client\\_final](#) (struct [scram\\_client\\_final](#) \*cl)
- void [scram\\_free\\_server\\_final](#) (struct [scram\\_server\\_final](#) \*sl)

## 5.98.1 Function Documentation

### 5.98.1.1 [scram\\_free\\_client\\_final\(\)](#)

```
void scram_free_client_final (
 struct scram_client_final * cl)
```

Definition at line 55 of file tokens.c.

### 5.98.1.2 [scram\\_free\\_client\\_first\(\)](#)

```
void scram_free_client_first (
 struct scram_client_first * cf)
```

Definition at line 35 of file tokens.c.

### 5.98.1.3 `scram_free_server_final()`

```
void scram_free_server_final (
 struct scram_server_final * sl)
```

Definition at line 65 of file `tokens.c`.

### 5.98.1.4 `scram_free_server_first()`

```
void scram_free_server_first (
 struct scram_server_first * sf)
```

Definition at line 46 of file `tokens.c`.

## 5.99 tokens.h File Reference

```
#include <stddef.h>
```

### Data Structures

- struct `digest_md5_challenge`
- struct `digest_md5_response`
- struct `digest_md5_finish`

### Macros

- #define `DIGEST_MD5_LENGTH` 16
- #define `DIGEST_MD5_RESPONSE_LENGTH` 32

### Typedefs

- typedef enum `digest_md5_qop` `digest_md5_qop`
- typedef enum `digest_md5_cipher` `digest_md5_cipher`
- typedef struct `digest_md5_challenge` `digest_md5_challenge`
- typedef struct `digest_md5_response` `digest_md5_response`
- typedef struct `digest_md5_finish` `digest_md5_finish`

### Enumerations

- enum `digest_md5_qop` { `DIGEST_MD5_QOP_AUTH` = 1 , `DIGEST_MD5_QOP_AUTH_INT` = 2 , `DIGEST_MD5_QOP_AUTH_CONF` = 4 }
- enum `digest_md5_cipher` { `DIGEST_MD5_CIPHER_DES` = 1 , `DIGEST_MD5_CIPHER_3DES` = 2 , `DIGEST_MD5_CIPHER_RC4` = 4 , `DIGEST_MD5_CIPHER_RC4_40` = 8 , `DIGEST_MD5_CIPHER_RC4_56` = 16 , `DIGEST_MD5_CIPHER_AES_CBC` = 32 }

## 5.99.1 Macro Definition Documentation

### 5.99.1.1 DIGEST\_MD5\_LENGTH

```
#define DIGEST_MD5_LENGTH 16
```

Definition at line 30 of file digest-md5/tokens.h.

### 5.99.1.2 DIGEST\_MD5\_RESPONSE\_LENGTH

```
#define DIGEST_MD5_RESPONSE_LENGTH 32
```

Definition at line 95 of file digest-md5/tokens.h.

## 5.99.2 Typedef Documentation

### 5.99.2.1 digest\_md5\_challenge

```
typedef struct digest_md5_challenge digest_md5_challenge
```

Definition at line 1 of file digest-md5/tokens.h.

### 5.99.2.2 digest\_md5\_cipher

```
typedef enum digest_md5_cipher digest_md5_cipher
```

Definition at line 1 of file digest-md5/tokens.h.

### 5.99.2.3 digest\_md5\_finish

```
typedef struct digest_md5_finish digest_md5_finish
```

Definition at line 1 of file digest-md5/tokens.h.



#### 5.99.2.4 digest\_md5\_qop

```
typedef enum digest_md5_qop digest_md5_qop
```

Definition at line 1 of file digest-md5/tokens.h.

#### 5.99.2.5 digest\_md5\_response

```
typedef struct digest_md5_response digest_md5_response
```

Definition at line 1 of file digest-md5/tokens.h.

### 5.99.3 Enumeration Type Documentation

#### 5.99.3.1 digest\_md5\_cipher

```
enum digest_md5_cipher
```

Enumerator

|                           |  |
|---------------------------|--|
| DIGEST_MD5_CIPHER_DES     |  |
| DIGEST_MD5_CIPHER_3DES    |  |
| DIGEST_MD5_CIPHER_RC4     |  |
| DIGEST_MD5_CIPHER_RC4_40  |  |
| DIGEST_MD5_CIPHER_RC4_56  |  |
| DIGEST_MD5_CIPHER_AES_CBC |  |

Definition at line 42 of file digest-md5/tokens.h.

#### 5.99.3.2 digest\_md5\_qop

```
enum digest_md5_qop
```

Enumerator

|                          |  |
|--------------------------|--|
| DIGEST_MD5_QOP_AUTH      |  |
| DIGEST_MD5_QOP_AUTH_INT  |  |
| DIGEST_MD5_QOP_AUTH_CONF |  |

Definition at line 33 of file digest-md5/tokens.h.

## 5.100 tokens.h File Reference

```
#include <stddef.h>
```

### Data Structures

- struct [scram\\_client\\_first](#)
- struct [scram\\_server\\_first](#)
- struct [scram\\_client\\_final](#)
- struct [scram\\_server\\_final](#)

### Functions

- void [scram\\_free\\_client\\_first](#) (struct [scram\\_client\\_first](#) \*cf)
- void [scram\\_free\\_server\\_first](#) (struct [scram\\_server\\_first](#) \*sf)
- void [scram\\_free\\_client\\_final](#) (struct [scram\\_client\\_final](#) \*cl)
- void [scram\\_free\\_server\\_final](#) (struct [scram\\_server\\_final](#) \*sl)

### 5.100.1 Function Documentation

#### 5.100.1.1 [scram\\_free\\_client\\_final\(\)](#)

```
void scram_free_client_final (
 struct scram_client_final * cl)
```

Definition at line 55 of file tokens.c.

#### 5.100.1.2 [scram\\_free\\_client\\_first\(\)](#)

```
void scram_free_client_first (
 struct scram_client_first * cf)
```

Definition at line 35 of file tokens.c.

#### 5.100.1.3 [scram\\_free\\_server\\_final\(\)](#)

```
void scram_free_server_final (
 struct scram_server_final * sl)
```

Definition at line 65 of file tokens.c.

### 5.100.1.4 `scram_free_server_first()`

```
void scram_free_server_first (
 struct scram_server_first * sf)
```

Definition at line 46 of file `tokens.c`.

## 5.101 tools.c File Reference

```
#include <config.h>
#include "tools.h"
#include "mechtools.h"
```

### Functions

- int `set_saltedpassword` (`Gsasl_session` \**sctx*, `Gsasl_hash` *hash*, const char \**hashbuf*)

### 5.101.1 Function Documentation

#### 5.101.1.1 `set_saltedpassword()`

```
int set_saltedpassword (
 Gsasl_session * sctx,
 Gsasl_hash hash,
 const char * hashbuf)
```

Definition at line 31 of file `tools.c`.

## 5.102 tools.h File Reference

```
#include <gsasl.h>
```

### Functions

- int `set_saltedpassword` (`Gsasl_session` \**sctx*, `Gsasl_hash` *hash*, const char \**hashbuf*)

### 5.102.1 Function Documentation

### 5.102.1.1 `set_saltedpassword()`

```
int set_saltedpassword (
 Gsasl_session * sctx,
 Gsasl_hash hash,
 const char * hashbuf)
```

Definition at line 31 of file tools.c.

## 5.103 `validate.c` File Reference

```
#include <config.h>
#include "validate.h"
#include <string.h>
```

### Functions

- int `digest_md5_validate_challenge` (`digest_md5_challenge *c`)
- int `digest_md5_validate_response` (`digest_md5_response *r`)
- int `digest_md5_validate_finish` (`digest_md5_finish *f`)
- int `digest_md5_validate` (`digest_md5_challenge *c`, `digest_md5_response *r`)

### 5.103.1 Function Documentation

#### 5.103.1.1 `digest_md5_validate()`

```
int digest_md5_validate (
 digest_md5_challenge * c,
 digest_md5_response * r)
```

Definition at line 114 of file digest-md5/validate.c.

#### 5.103.1.2 `digest_md5_validate_challenge()`

```
int digest_md5_validate_challenge (
 digest_md5_challenge * c)
```

Definition at line 32 of file digest-md5/validate.c.

### 5.103.1.3 digest\_md5\_validate\_finish()

```
int digest_md5_validate_finish (
 digest_md5_finish * f)
```

Definition at line 101 of file digest-md5/validate.c.

### 5.103.1.4 digest\_md5\_validate\_response()

```
int digest_md5_validate_response (
 digest_md5_response * r)
```

Definition at line 51 of file digest-md5/validate.c.

## 5.104 validate.c File Reference

```
#include <config.h>
#include "validate.h"
#include <string.h>
```

### Functions

- bool [scram\\_valid\\_client\\_first](#) (struct [scram\\_client\\_first](#) \*cf)
- bool [scram\\_valid\\_server\\_first](#) (struct [scram\\_server\\_first](#) \*sf)
- bool [scram\\_valid\\_client\\_final](#) (struct [scram\\_client\\_final](#) \*cl)
- bool [scram\\_valid\\_server\\_final](#) (struct [scram\\_server\\_final](#) \*sl)

## 5.104.1 Function Documentation

### 5.104.1.1 scram\_valid\_client\_final()

```
bool scram_valid_client_final (
 struct scram_client_final * cl)
```

Definition at line 104 of file scram/validate.c.

### 5.104.1.2 `scram_valid_client_first()`

```
bool scram_valid_client_first (
 struct scram_client_first * cf)
```

Definition at line 32 of file `scram/validate.c`.

### 5.104.1.3 `scram_valid_server_final()`

```
bool scram_valid_server_final (
 struct scram_server_final * sl)
```

Definition at line 134 of file `scram/validate.c`.

### 5.104.1.4 `scram_valid_server_first()`

```
bool scram_valid_server_first (
 struct scram_server_first * sf)
```

Definition at line 79 of file `scram/validate.c`.

## 5.105 `validate.h` File Reference

```
#include "tokens.h"
```

### Functions

- int `digest_md5_validate_challenge` (`digest_md5_challenge` \**c*)
- int `digest_md5_validate_response` (`digest_md5_response` \**r*)
- int `digest_md5_validate_finish` (`digest_md5_finish` \**f*)
- int `digest_md5_validate` (`digest_md5_challenge` \**c*, `digest_md5_response` \**r*)

### 5.105.1 Function Documentation

#### 5.105.1.1 `digest_md5_validate()`

```
int digest_md5_validate (
 digest_md5_challenge * c,
 digest_md5_response * r)
```

Definition at line 114 of file `digest-md5/validate.c`.

### 5.105.1.2 digest\_md5\_validate\_challenge()

```
int digest_md5_validate_challenge (
 digest_md5_challenge * c)
```

Definition at line 32 of file digest-md5/validate.c.

### 5.105.1.3 digest\_md5\_validate\_finish()

```
int digest_md5_validate_finish (
 digest_md5_finish * f)
```

Definition at line 101 of file digest-md5/validate.c.

### 5.105.1.4 digest\_md5\_validate\_response()

```
int digest_md5_validate_response (
 digest_md5_response * r)
```

Definition at line 51 of file digest-md5/validate.c.

## 5.106 validate.h File Reference

```
#include "tokens.h"
#include <stdbool.h>
```

### Functions

- bool [scram\\_valid\\_client\\_first](#) (struct [scram\\_client\\_first](#) \*cf)
- bool [scram\\_valid\\_server\\_first](#) (struct [scram\\_server\\_first](#) \*sf)
- bool [scram\\_valid\\_client\\_final](#) (struct [scram\\_client\\_final](#) \*cl)
- bool [scram\\_valid\\_server\\_final](#) (struct [scram\\_server\\_final](#) \*sl)

### 5.106.1 Function Documentation

#### 5.106.1.1 scram\_valid\_client\_final()

```
bool scram_valid_client_final (
 struct scram_client_final * cl)
```

Definition at line 104 of file scram/validate.c.

### 5.106.1.2 `scram_valid_client_first()`

```
bool scram_valid_client_first (
 struct scram_client_first * cf)
```

Definition at line 32 of file `scram/validate.c`.

### 5.106.1.3 `scram_valid_server_final()`

```
bool scram_valid_server_final (
 struct scram_server_final * sf)
```

Definition at line 134 of file `scram/validate.c`.

### 5.106.1.4 `scram_valid_server_first()`

```
bool scram_valid_server_first (
 struct scram_server_first * sf)
```

Definition at line 79 of file `scram/validate.c`.

## 5.107 `version.c` File Reference

```
#include <config.h>
#include "internal.h"
#include <string.h>
```

### Functions

- `const char * gsasl\_check\_version` (`const char *req_version`)

### 5.107.1 Function Documentation

#### 5.107.1.1 `gsasl_check_version()`

```
const char* gsasl_check_version (
 const char * req_version)
```

`gsasl_check_version`:



## Parameters

|                    |                                          |
|--------------------|------------------------------------------|
| <i>req_version</i> | version string to compare with, or NULL. |
|--------------------|------------------------------------------|

Check GNU SASL Library version.

See GSASL\_VERSION for a suitable @req\_version string.

This function is one of few in the library that can be used without a successful call to [gsasl\\_init\(\)](#).

Return value: Check that the version of the library is at minimum the one given as a string in @req\_version and return the actual version string of the library; return NULL if the condition is not met. If NULL is passed to this function no check is done and only the version string is returned.

Definition at line 46 of file version.c.

## 5.108 x-gssapi.h File Reference

```
#include <gsasl.h>
```

### Macros

- #define [GSASL\\_GSSAPI\\_NAME](#) "GSSAPI"

### Functions

- int [\\_gsasl\\_gssapi\\_client\\_start](#) ([Gsasl\\_session](#) \*sctx, void \*\*mech\_data)
- int [\\_gsasl\\_gssapi\\_client\\_step](#) ([Gsasl\\_session](#) \*sctx, void \*mech\_data, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len)
- void [\\_gsasl\\_gssapi\\_client\\_finish](#) ([Gsasl\\_session](#) \*sctx, void \*mech\_data)
- int [\\_gsasl\\_gssapi\\_client\\_encode](#) ([Gsasl\\_session](#) \*sctx, void \*mech\_data, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len)
- int [\\_gsasl\\_gssapi\\_client\\_decode](#) ([Gsasl\\_session](#) \*sctx, void \*mech\_data, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len)
- int [\\_gsasl\\_gssapi\\_server\\_start](#) ([Gsasl\\_session](#) \*sctx, void \*\*mech\_data)
- int [\\_gsasl\\_gssapi\\_server\\_step](#) ([Gsasl\\_session](#) \*sctx, void \*mech\_data, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_len)
- void [\\_gsasl\\_gssapi\\_server\\_finish](#) ([Gsasl\\_session](#) \*sctx, void \*mech\_data)

### Variables

- [Gsasl\\_mechanism\\_gsasl\\_gssapi\\_mechanism](#)

### 5.108.1 Macro Definition Documentation

### 5.108.1.1 GSASL\_GSSAPI\_NAME

```
#define GSASL_GSSAPI_NAME "GSSAPI"
```

Definition at line 28 of file x-gssapi.h.

## 5.108.2 Function Documentation

### 5.108.2.1 \_gsasl\_gssapi\_client\_decode()

```
int _gsasl_gssapi_client_decode (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 322 of file gssapi/client.c.

### 5.108.2.2 \_gsasl\_gssapi\_client\_encode()

```
int _gsasl_gssapi_client_encode (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 267 of file gssapi/client.c.

### 5.108.2.3 \_gsasl\_gssapi\_client\_finish()

```
void _gsasl_gssapi_client_finish (
 Gsasl_session * sctx,
 void * mech_data)
```

Definition at line 249 of file gssapi/client.c.

#### 5.108.2.4 `_gsasl_gssapi_client_start()`

```
int _gsasl_gssapi_client_start (
 Gsasl_session * sctx,
 void ** mech_data)
```

Definition at line 47 of file gssapi/client.c.

#### 5.108.2.5 `_gsasl_gssapi_client_step()`

```
int _gsasl_gssapi_client_step (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 66 of file gssapi/client.c.

#### 5.108.2.6 `_gsasl_gssapi_server_finish()`

```
void _gsasl_gssapi_server_finish (
 Gsasl_session * sctx,
 void * mech_data)
```

Definition at line 266 of file gssapi/server.c.

#### 5.108.2.7 `_gsasl_gssapi_server_start()`

```
int _gsasl_gssapi_server_start (
 Gsasl_session * sctx,
 void ** mech_data)
```

Definition at line 47 of file gssapi/server.c.

#### 5.108.2.8 `_gsasl_gssapi_server_step()`

```
int _gsasl_gssapi_server_step (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 65 of file gssapi/server.c.

### 5.108.3 Variable Documentation

#### 5.108.3.1 `_gsasl_gssapi_mechanism`

`Gsasl_mechanism` `_gsasl_gssapi_mechanism` [extern]

Definition at line 28 of file `gssapi/mechinfo.c`.

## 5.109 x-ntlm.h File Reference

```
#include <gsasl.h>
```

### Macros

- `#define GSASL_NTLM_NAME "NTLM"`

### Functions

- `int _gsasl_ntlm_client_start (Gsasl_session *sctx, void **mech_data)`
- `int _gsasl_ntlm_client_step (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)`
- `void _gsasl_ntlm_client_finish (Gsasl_session *sctx, void *mech_data)`

### Variables

- `Gsasl_mechanism _gsasl_ntlm_mechanism`

### 5.109.1 Macro Definition Documentation

#### 5.109.1.1 `GSASL_NTLM_NAME`

```
#define GSASL_NTLM_NAME "NTLM"
```

Definition at line 28 of file `x-ntlm.h`.

### 5.109.2 Function Documentation

### 5.109.2.1 `_gsasl_ntlm_client_finish()`

```
void _gsasl_ntlm_client_finish (
 Gsasl_session * sctx,
 void * mech_data)
```

### 5.109.2.2 `_gsasl_ntlm_client_start()`

```
int _gsasl_ntlm_client_start (
 Gsasl_session * sctx,
 void ** mech_data)
```

### 5.109.2.3 `_gsasl_ntlm_client_step()`

```
int _gsasl_ntlm_client_step (
 Gsasl_session * sctx,
 void * mech_data,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

Definition at line 59 of file ntlm.c.

## 5.109.3 Variable Documentation

### 5.109.3.1 `_gsasl_ntlm_mechanism`

```
Gsasl_mechanism _gsasl_ntlm_mechanism [extern]
```

Definition at line 28 of file ntlm/mechinfo.c.

## 5.110 xcode.c File Reference

```
#include <config.h>
#include "internal.h"
```

## Functions

- int `gsasl_encode` (`Gsasl_session` \*sctx, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_↵len)
- int `gsasl_decode` (`Gsasl_session` \*sctx, const char \*input, size\_t input\_len, char \*\*output, size\_t \*output\_↵len)

### 5.110.1 Function Documentation

#### 5.110.1.1 `gsasl_decode()`

```
int gsasl_decode (
 Gsasl_session * sctx,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

`gsasl_decode`:

##### Parameters

|                   |                                                            |
|-------------------|------------------------------------------------------------|
| <i>sctx</i>       | libgsasl session handle.                                   |
| <i>input</i>      | input byte array.                                          |
| <i>input_len</i>  | size of input byte array.                                  |
| <i>output</i>     | newly allocated output byte array.                         |
| <i>output_len</i> | pointer to output variable with size of output byte array. |

Decode data according to negotiated SASL mechanism. This might mean that data is integrity or privacy protected.

The @output buffer is allocated by this function, and it is the responsibility of caller to deallocate it by calling `gsasl_↵_free(@output)`.

Return value: Returns GSASL\_OK if encoding was successful, otherwise an error code.

Definition at line 99 of file xcode.c.

#### 5.110.1.2 `gsasl_encode()`

```
int gsasl_encode (
 Gsasl_session * sctx,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

`gsasl_encode`:

**Parameters**

|                   |                                                            |
|-------------------|------------------------------------------------------------|
| <i>sctx</i>       | libgsasl session handle.                                   |
| <i>input</i>      | input byte array.                                          |
| <i>input_len</i>  | size of input byte array.                                  |
| <i>output</i>     | newly allocated output byte array.                         |
| <i>output_len</i> | pointer to output variable with size of output byte array. |

Encode data according to negotiated SASL mechanism. This might mean that data is integrity or privacy protected.

The @output buffer is allocated by this function, and it is the responsibility of caller to deallocate it by calling `gsasl_free(@output)`.

Return value: Returns GSASL\_OK if encoding was successful, otherwise an error code.

Definition at line 66 of file xcode.c.

**5.111 xfinish.c File Reference**

```
#include <config.h>
#include "internal.h"
```

**Functions**

- void `gsasl_finish` (`Gsasl_session *sctx`)

**5.111.1 Function Documentation****5.111.1.1 gsasl\_finish()**

```
void gsasl_finish (
 Gsasl_session * sctx)
```

gsasl\_finish:

**Parameters**

|             |                          |
|-------------|--------------------------|
| <i>sctx</i> | libgsasl session handle. |
|-------------|--------------------------|

Destroy a libgsasl client or server handle. The handle must not be used with other libgsasl functions after this call.

Definition at line 34 of file xfinish.c.

## 5.112 xstart.c File Reference

```
#include <config.h>
#include "internal.h"
```

### Functions

- int [gsasl\\_client\\_start](#) ([Gsasl](#) \*ctx, const char \*mech, [Gsasl\\_session](#) \*\*sctx)
- int [gsasl\\_server\\_start](#) ([Gsasl](#) \*ctx, const char \*mech, [Gsasl\\_session](#) \*\*sctx)

### 5.112.1 Function Documentation

#### 5.112.1.1 [gsasl\\_client\\_start\(\)](#)

```
int gsasl_client_start (
 Gsasl * ctx,
 const char * mech,
 Gsasl_session ** sctx)
```

[gsasl\\_client\\_start](#):

#### Parameters

|             |                           |
|-------------|---------------------------|
| <i>ctx</i>  | libgsasl handle.          |
| <i>mech</i> | name of SASL mechanism.   |
| <i>sctx</i> | pointer to client handle. |

This functions initiates a client SASL authentication. This function must be called before any other [gsasl\\_client\\_\\*](#)(*)* function is called.

Return value: Returns GSASL\_OK if successful, or error code.

Definition at line 120 of file xstart.c.

#### 5.112.1.2 [gsasl\\_server\\_start\(\)](#)

```
int gsasl_server_start (
 Gsasl * ctx,
 const char * mech,
 Gsasl_session ** sctx)
```

[gsasl\\_server\\_start](#):



## Parameters

|             |                           |
|-------------|---------------------------|
| <i>ctx</i>  | libgsasl handle.          |
| <i>mech</i> | name of SASL mechanism.   |
| <i>sctx</i> | pointer to server handle. |

This functions initiates a server SASL authentication. This function must be called before any other `gsasl_server_*`() function is called.

Return value: Returns `GSASL_OK` if successful, or error code.

Definition at line 138 of file `xstart.c`.

## 5.113 xstep.c File Reference

```
#include <config.h>
#include "internal.h"
```

### Functions

- int `gsasl_step` (`Gsasl_session` \**sctx*, const char \**input*, size\_t *input\_len*, char \*\**output*, size\_t \**output\_len*)
- int `gsasl_step64` (`Gsasl_session` \**sctx*, const char \**b64input*, char \*\**b64output*)

### 5.113.1 Function Documentation

#### 5.113.1.1 `gsasl_step()`

```
int gsasl_step (
 Gsasl_session * sctx,
 const char * input,
 size_t input_len,
 char ** output,
 size_t * output_len)
```

`gsasl_step`:

## Parameters

|                   |                                                            |
|-------------------|------------------------------------------------------------|
| <i>sctx</i>       | libgsasl session handle.                                   |
| <i>input</i>      | input byte array.                                          |
| <i>input_len</i>  | size of input byte array.                                  |
| <i>output</i>     | newly allocated output byte array.                         |
| <i>output_len</i> | pointer to output variable with size of output byte array. |

Perform one step of SASL authentication. This reads data from the other end (from @input and @input\_len), processes it (potentially invoking callbacks to the application), and writes data to server (into newly allocated variable @output and @output\_len that indicate the length of @output).

The contents of the @output buffer is unspecified if this functions returns anything other than GSASL\_OK or GSASL\_NEEDS\_MORE. If this function return GSASL\_OK or GSASL\_NEEDS\_MORE, however, the @output buffer is allocated by this function, and it is the responsibility of caller to deallocate it by calling gsasl\_free(@output).

Return value: Returns GSASL\_OK if authenticated terminated successfully, GSASL\_NEEDS\_MORE if more data is needed, or error code.

Definition at line 52 of file xstep.c.

### 5.113.1.2 gsasl\_step64()

```
int gsasl_step64 (
 Gsasl_session * sctx,
 const char * b64input,
 char ** b64output)
```

gsasl\_step64:

#### Parameters

|                  |                                                   |
|------------------|---------------------------------------------------|
| <i>sctx</i>      | libgsasl client handle.                           |
| <i>b64input</i>  | input base64 encoded byte array.                  |
| <i>b64output</i> | newly allocated output base64 encoded byte array. |

This is a simple wrapper around [gsasl\\_step\(\)](#) that base64 decodes the input and base64 encodes the output.

The contents of the @b64output buffer is unspecified if this functions returns anything other than GSASL\_OK or GSASL\_NEEDS\_MORE. If this function return GSASL\_OK or GSASL\_NEEDS\_MORE, however, the @b64output buffer is allocated by this function, and it is the responsibility of caller to deallocate it by calling gsasl\_free(@b64output).

Return value: Returns GSASL\_OK if authenticated terminated successfully, GSASL\_NEEDS\_MORE if more data is needed, or error code.

Definition at line 87 of file xstep.c.

# Index

- - error.c, [97](#)
- [\\_GSASL\\_API](#)
  - gsasl.h, [116](#)
- [\\_GSASL\\_VALID\\_MECHANISM\\_CHARACTERS](#)
  - suggest.c, [210](#)
- [\\_Gsasl\\_digest\\_md5\\_client\\_state](#), [17](#)
  - challenge, [17](#)
  - digest-md5/client.c, [69](#)
  - finish, [17](#)
  - kcc, [18](#)
  - kcs, [18](#)
  - kic, [18](#)
  - kis, [18](#)
  - readseqnum, [18](#)
  - response, [18](#)
  - secret, [19](#)
  - sendseqnum, [19](#)
  - step, [19](#)
- [\\_Gsasl\\_digest\\_md5\\_server\\_state](#), [19](#)
  - challenge, [20](#)
  - digest-md5/server.c, [193](#)
  - finish, [20](#)
  - kcc, [20](#)
  - kcs, [20](#)
  - kic, [20](#)
  - kis, [20](#)
  - readseqnum, [21](#)
  - response, [21](#)
  - secret, [21](#)
  - sendseqnum, [21](#)
  - step, [21](#)
- [\\_Gsasl\\_gs2\\_server\\_state](#), [23](#)
  - cb, [23](#)
  - client, [24](#)
  - context, [24](#)
  - cred, [24](#)
  - gs2/server.c, [195](#)
  - mech\_oid, [24](#)
  - step, [24](#)
- [\\_Gsasl\\_gssapi\\_client\\_state](#), [25](#)
  - context, [25](#)
  - gssapi/client.c, [73](#)
  - qop, [25](#)
  - service, [25](#)
  - step, [25](#)
- [\\_Gsasl\\_gssapi\\_server\\_state](#), [26](#)
  - client, [26](#)
  - context, [26](#)
  - cred, [26](#)
  - gssapi/server.c, [197](#)
  - step, [26](#)
- [\\_Gsasl\\_login\\_client\\_state](#), [27](#)
  - step, [27](#)
- [\\_Gsasl\\_login\\_server\\_state](#), [27](#)
  - password, [27](#)
  - step, [28](#)
  - username, [28](#)
- [\\_Gsasl\\_ntlm\\_state](#), [28](#)
  - ntlm.c, [164](#)
  - step, [28](#)
- [\\_gsasl\\_anonymous\\_client\\_step](#)
  - anonymous.h, [58](#)
  - anonymous/client.c, [66](#)
- [\\_gsasl\\_anonymous\\_mechanism](#)
  - anonymous.h, [58](#)
  - anonymous/mechinfo.c, [150](#)
- [\\_gsasl\\_anonymous\\_server\\_step](#)
  - anonymous.h, [58](#)
  - anonymous/server.c, [190](#)
- [\\_gsasl\\_cram\\_md5\\_client\\_step](#)
  - cram-md5.h, [82](#)
  - cram-md5/client.c, [67](#)
- [\\_gsasl\\_cram\\_md5\\_mechanism](#)
  - cram-md5.h, [83](#)
  - cram-md5/mechinfo.c, [151](#)
- [\\_gsasl\\_cram\\_md5\\_server\\_finish](#)
  - cram-md5.h, [82](#)
  - cram-md5/server.c, [191](#)
- [\\_gsasl\\_cram\\_md5\\_server\\_start](#)
  - cram-md5.h, [82](#)
  - cram-md5/server.c, [191](#)
- [\\_gsasl\\_cram\\_md5\\_server\\_step](#)
  - cram-md5.h, [82](#)
  - cram-md5/server.c, [191](#)
- [\\_gsasl\\_digest\\_md5\\_client\\_decode](#)
  - digest-md5.h, [87](#)
  - digest-md5/client.c, [69](#)
- [\\_gsasl\\_digest\\_md5\\_client\\_encode](#)
  - digest-md5.h, [88](#)
  - digest-md5/client.c, [69](#)
- [\\_gsasl\\_digest\\_md5\\_client\\_finish](#)
  - digest-md5.h, [88](#)
  - digest-md5/client.c, [69](#)
- [\\_gsasl\\_digest\\_md5\\_client\\_start](#)
  - digest-md5.h, [88](#)
  - digest-md5/client.c, [69](#)
- [\\_gsasl\\_digest\\_md5\\_client\\_step](#)

- digest-md5.h, 88
- digest-md5/client.c, 70
- `_gsasl_digest_md5_mechanism`
  - digest-md5.h, 90
  - digest-md5/mechinfo.c, 151
- `_gsasl_digest_md5_server_decode`
  - digest-md5.h, 88
  - digest-md5/server.c, 193
- `_gsasl_digest_md5_server_encode`
  - digest-md5.h, 89
  - digest-md5/server.c, 193
- `_gsasl_digest_md5_server_finish`
  - digest-md5.h, 89
  - digest-md5/server.c, 193
- `_gsasl_digest_md5_server_start`
  - digest-md5.h, 89
  - digest-md5/server.c, 194
- `_gsasl_digest_md5_server_step`
  - digest-md5.h, 89
  - digest-md5/server.c, 194
- `_gsasl_external_client_step`
  - external.h, 100
  - external/client.c, 70
- `_gsasl_external_mechanism`
  - external.h, 101
  - external/mechinfo.c, 152
- `_gsasl_external_server_step`
  - external.h, 100
  - external/server.c, 194
- `_gsasl_gs2_client_finish`
  - gs2.h, 105
  - gs2/client.c, 71
- `_gsasl_gs2_client_start`
  - gs2.h, 105
  - gs2/client.c, 71
- `_gsasl_gs2_client_state`, 22
  - cb, 22
  - context, 22
  - gs2/client.c, 71
  - mech\_oid, 22
  - service, 22
  - step, 23
  - token, 23
- `_gsasl_gs2_client_step`
  - gs2.h, 105
  - gs2/client.c, 72
- `_gsasl_gs2_generate_header`
  - mechtools.c, 158
  - mechtools.h, 160
- `_gsasl_gs2_krb5_mechanism`
  - gs2.h, 106
  - gs2/mechinfo.c, 152
- `_gsasl_gs2_server_finish`
  - gs2.h, 105
  - gs2/server.c, 196
- `_gsasl_gs2_server_start`
  - gs2.h, 106
  - gs2/server.c, 196
- `_gsasl_gs2_server_step`
  - gs2.h, 106
  - gs2/server.c, 196
- `_gsasl_gssapi_client_decode`
  - gssapi/client.c, 73
  - x-gssapi.h, 224
- `_gsasl_gssapi_client_encode`
  - gssapi/client.c, 73
  - x-gssapi.h, 224
- `_gsasl_gssapi_client_finish`
  - gssapi/client.c, 73
  - x-gssapi.h, 224
- `_gsasl_gssapi_client_start`
  - gssapi/client.c, 73
  - x-gssapi.h, 224
- `_gsasl_gssapi_client_step`
  - gssapi/client.c, 74
  - x-gssapi.h, 225
- `_gsasl_gssapi_mechanism`
  - gssapi/mechinfo.c, 153
  - x-gssapi.h, 226
- `_gsasl_gssapi_server_finish`
  - gssapi/server.c, 197
  - x-gssapi.h, 225
- `_gsasl_gssapi_server_start`
  - gssapi/server.c, 197
  - x-gssapi.h, 225
- `_gsasl_gssapi_server_step`
  - gssapi/server.c, 197
  - x-gssapi.h, 225
- `_gsasl_hash`
  - mechtools.c, 158
  - mechtools.h, 160
- `_gsasl_hex_decode`
  - mechtools.c, 158
  - mechtools.h, 161
- `_gsasl_hex_encode`
  - mechtools.c, 158
  - mechtools.h, 161
- `_gsasl_hex_p`
  - mechtools.c, 159
  - mechtools.h, 161
- `_gsasl_hmac`
  - mechtools.c, 159
  - mechtools.h, 161
- `_gsasl_login_client_finish`
  - login.h, 147
  - login/client.c, 74
- `_gsasl_login_client_start`
  - login.h, 148
  - login/client.c, 75
- `_gsasl_login_client_step`
  - login.h, 148
  - login/client.c, 75
- `_gsasl_login_mechanism`
  - login.h, 149
  - login/mechinfo.c, 153
- `_gsasl_login_server_finish`

- login.h, 148
- login/server.c, 199
- \_\_gsasl\_login\_server\_start
  - login.h, 148
  - login/server.c, 199
- \_\_gsasl\_login\_server\_step
  - login.h, 148
  - login/server.c, 199
- \_\_gsasl\_ntlm\_client\_finish
  - ntlm.c, 164
  - x-ntlm.h, 226
- \_\_gsasl\_ntlm\_client\_start
  - ntlm.c, 164
  - x-ntlm.h, 227
- \_\_gsasl\_ntlm\_client\_step
  - ntlm.c, 164
  - x-ntlm.h, 227
- \_\_gsasl\_ntlm\_mechanism
  - ntlm/mechinfo.c, 154
  - x-ntlm.h, 227
- \_\_gsasl\_openid20\_client\_finish
  - openid20.h, 166
  - openid20/client.c, 76
- \_\_gsasl\_openid20\_client\_start
  - openid20.h, 166
  - openid20/client.c, 76
- \_\_gsasl\_openid20\_client\_step
  - openid20.h, 166
  - openid20/client.c, 76
- \_\_gsasl\_openid20\_mechanism
  - openid20.h, 167
  - openid20/mechinfo.c, 154
- \_\_gsasl\_openid20\_server\_finish
  - openid20.h, 166
  - openid20/server.c, 200
- \_\_gsasl\_openid20\_server\_start
  - openid20.h, 166
  - openid20/server.c, 200
- \_\_gsasl\_openid20\_server\_step
  - openid20.h, 166
  - openid20/server.c, 200
- \_\_gsasl\_parse\_gs2\_header
  - mechtools.c, 159
  - mechtools.h, 161
- \_\_gsasl\_pbkdf2
  - mechtools.c, 159
  - mechtools.h, 162
- \_\_gsasl\_plain\_client\_step
  - plain.h, 175
  - plain/client.c, 77
- \_\_gsasl\_plain\_mechanism
  - plain.h, 175
  - plain/mechinfo.c, 155
- \_\_gsasl\_plain\_server\_step
  - plain.h, 175
  - plain/server.c, 201
- \_\_gsasl\_saml20\_client\_finish
  - saml20.h, 185
  - saml20/client.c, 78
- \_\_gsasl\_saml20\_client\_start
  - saml20.h, 186
  - saml20/client.c, 78
- \_\_gsasl\_saml20\_client\_step
  - saml20.h, 186
  - saml20/client.c, 78
- \_\_gsasl\_saml20\_mechanism
  - saml20.h, 187
  - saml20/mechinfo.c, 156
- \_\_gsasl\_saml20\_server\_finish
  - saml20.h, 186
  - saml20/server.c, 202
- \_\_gsasl\_saml20\_server\_start
  - saml20.h, 186
  - saml20/server.c, 202
- \_\_gsasl\_saml20\_server\_step
  - saml20.h, 186
  - saml20/server.c, 202
- \_\_gsasl\_scram\_client\_finish
  - scram/client.c, 79
- \_\_gsasl\_scram\_client\_step
  - scram/client.c, 79
- \_\_gsasl\_scram\_server\_finish
  - scram/server.c, 203
- \_\_gsasl\_scram\_server\_step
  - scram/server.c, 203
- \_\_gsasl\_secured\_client\_finish
  - secured.h, 188
  - secured/client.c, 80
- \_\_gsasl\_secured\_client\_start
  - secured.h, 189
  - secured/client.c, 81
- \_\_gsasl\_secured\_client\_step
  - secured.h, 189
  - secured/client.c, 81
- \_\_gsasl\_secured\_mechanism
  - secured.h, 189
  - secured/mechinfo.c, 156
- \_\_gsasl\_secured\_server\_step
  - secured.h, 189
  - secured/server.c, 205
- A2\_POST
  - digesthmac.c, 92
- A2\_PRE
  - digesthmac.c, 92
- allow\_error\_step
  - openid20\_server\_state, 45
- anonymous.h, 57
  - \_\_gsasl\_anonymous\_client\_step, 58
  - \_\_gsasl\_anonymous\_mechanism, 58
  - \_\_gsasl\_anonymous\_server\_step, 58
  - GSASL\_ANONYMOUS\_NAME, 57
- anonymous/client.c
  - \_\_gsasl\_anonymous\_client\_step, 66
- anonymous/mechinfo.c
  - \_\_gsasl\_anonymous\_mechanism, 150
- anonymous/server.c

- [\\_gsasl\\_anonymous\\_server\\_step](#), 190
- anonymous\_token
  - [Gsasl\\_session](#), 39
- application\_hook
  - [Gsasl](#), 34
  - [Gsasl\\_session](#), 39
- authid
  - [Gsasl\\_session](#), 39
- authmessage
  - [scram\\_client\\_state](#), 49
  - [scram\\_server\\_state](#), 53
- authzid
  - [digest\\_md5\\_response](#), 32
  - [Gsasl\\_session](#), 39
  - [scram\\_client\\_first](#), 48
- base64.c, 58
  - [gsasl\\_base64\\_from](#), 59
  - [gsasl\\_base64\\_to](#), 59
  - [gsasl\\_hex\\_from](#), 60
  - [gsasl\\_hex\\_to](#), 60
- C2I
  - [session.c](#), 205
- callback.c, 61
  - [gsasl\\_callback](#), 61
  - [gsasl\\_callback\\_hook\\_get](#), 62
  - [gsasl\\_callback\\_hook\\_set](#), 62
  - [gsasl\\_callback\\_set](#), 63
  - [gsasl\\_session\\_hook\\_get](#), 63
  - [gsasl\\_session\\_hook\\_set](#), 64
- cb
  - [\\_Gsasl\\_gs2\\_server\\_state](#), 23
  - [\\_gsasl\\_gs2\\_client\\_state](#), 22
  - [Gsasl](#), 34
  - [scram\\_server\\_state](#), 53
- cb\_tls\_exporter
  - [Gsasl\\_session](#), 40
- cb\_tls\_unique
  - [Gsasl\\_session](#), 40
- cbflag
  - [scram\\_client\\_first](#), 48
- cbind
  - [scram\\_client\\_final](#), 47
  - [scram\\_server\\_state](#), 53
- cblen
  - [scram\\_server\\_state](#), 54
- cbname
  - [scram\\_client\\_first](#), 48
- cf
  - [scram\\_client\\_state](#), 49
  - [scram\\_server\\_state](#), 54
- cfmb
  - [scram\\_client\\_state](#), 49
- cfmb\_str
  - [scram\\_server\\_state](#), 54
- challenge
  - [\\_Gsasl\\_digest\\_md5\\_client\\_state](#), 17
  - [\\_Gsasl\\_digest\\_md5\\_server\\_state](#), 20
- challenge.c, 64
  - [cram\\_md5\\_challenge](#), 65
  - [DIGIT](#), 65
  - [NONCELEN](#), 65
  - [TEMPLATE](#), 65
- challenge.h, 65
  - [cram\\_md5\\_challenge](#), 66
  - [CRAM\\_MD5\\_CHALLENGE\\_LEN](#), 66
- CHALLENGE\_ALGORITHM
  - [digest-md5/parser.c](#), 168
- CHALLENGE\_CHARSET
  - [digest-md5/parser.c](#), 168
- CHALLENGE\_CIPHER
  - [digest-md5/parser.c](#), 168
- CHALLENGE\_MAXBUF
  - [digest-md5/parser.c](#), 168
- CHALLENGE\_NONCE
  - [digest-md5/parser.c](#), 168
- CHALLENGE\_PASSWORD
  - [login/server.c](#), 198
- CHALLENGE\_QOP
  - [digest-md5/parser.c](#), 168
- CHALLENGE\_REALM
  - [digest-md5/parser.c](#), 168
- CHALLENGE\_STALE
  - [digest-md5/parser.c](#), 168
- CHALLENGE\_USERNAME
  - [login/server.c](#), 198
- cipher
  - [digest\\_md5\\_response](#), 32
- CIPHER\_3DES
  - [digest-md5/parser.c](#), 169
- CIPHER\_AES\_CBC
  - [digest-md5/parser.c](#), 169
- CIPHER\_DES
  - [digest-md5/parser.c](#), 169
- CIPHER\_RC4
  - [digest-md5/parser.c](#), 169
- CIPHER\_RC4\_40
  - [digest-md5/parser.c](#), 169
- CIPHER\_RC4\_56
  - [digest-md5/parser.c](#), 169
- ciphers
  - [digest\\_md5\\_challenge](#), 29
- cl
  - [scram\\_client\\_state](#), 50
  - [scram\\_server\\_state](#), 54
- client
  - [\\_Gsasl\\_gs2\\_server\\_state](#), 24
  - [\\_Gsasl\\_gssapi\\_server\\_state](#), 26
  - [Gsasl\\_mechanism](#), 36
- client.c, 66–68, 70–72, 74, 75, 77, 78, 80
- CLIENT\_KEY
  - [crypto.c](#), 84
- client\_mechs
  - [Gsasl](#), 34
- client\_nonce
  - [scram\\_client\\_first](#), 48

- clientmaxbuf
  - digest\_md5\_response, 32
- clientp
  - Gsasl\_session, 40
- clientproof
  - scram\_server\_state, 54
- cnonce
  - digest\_md5\_response, 32
- CNONCE\_ENTROPY\_BYTES
  - digest-md5/client.c, 68
  - scram/client.c, 79
- COLON
  - digesthmac.c, 93
- context
  - \_Gsasl\_gs2\_server\_state, 24
  - \_Gsasl\_gssapi\_client\_state, 25
  - \_Gsasl\_gssapi\_server\_state, 26
  - \_gsasl\_gs2\_client\_state, 22
- cram-md5.h, 81
  - \_gsasl\_cram\_md5\_client\_step, 82
  - \_gsasl\_cram\_md5\_mechanism, 83
  - \_gsasl\_cram\_md5\_server\_finish, 82
  - \_gsasl\_cram\_md5\_server\_start, 82
  - \_gsasl\_cram\_md5\_server\_step, 82
  - GSASL\_CRAM\_MD5\_NAME, 82
- cram-md5/client.c
  - \_gsasl\_cram\_md5\_client\_step, 67
- cram-md5/mechinfo.c
  - \_gsasl\_cram\_md5\_mechanism, 151
- cram-md5/server.c
  - \_gsasl\_cram\_md5\_server\_finish, 191
  - \_gsasl\_cram\_md5\_server\_start, 191
  - \_gsasl\_cram\_md5\_server\_step, 191
  - MD5LEN, 191
- cram\_md5\_challenge
  - challenge.c, 65
  - challenge.h, 66
- CRAM\_MD5\_CHALLENGE\_LEN
  - challenge.h, 66
- cram\_md5\_digest
  - digest.c, 90
  - digest.h, 91
- CRAM\_MD5\_DIGEST\_LEN
  - digest.h, 91
- cred
  - \_Gsasl\_gs2\_server\_state, 24
  - \_Gsasl\_gssapi\_server\_state, 26
- crypto.c, 83
  - CLIENT\_KEY, 84
  - gsasl\_hash\_length, 84
  - gsasl\_nonce, 84
  - gsasl\_random, 85
  - gsasl\_scram\_secrets\_from\_password, 85
  - gsasl\_scram\_secrets\_from\_salted\_password, 86
  - SERVER\_KEY, 84
- ctx
  - Gsasl\_session, 40
- decode
  - Gsasl\_mechanism\_functions, 37
- DEFAULT\_ALGORITHM
  - digest-md5/parser.c, 168
- DEFAULT\_CHARSET
  - digest-md5/parser.c, 168
- DEFAULT\_SALT\_BYTES
  - scram/server.c, 203
- DERIVE\_CLIENT\_CONFIDENTIALITY\_KEY\_STRING
  - digesthmac.c, 93
- DERIVE\_CLIENT\_CONFIDENTIALITY\_KEY\_STRING\_LEN
  - digesthmac.c, 93
- DERIVE\_CLIENT\_INTEGRITY\_KEY\_STRING
  - digesthmac.c, 93
- DERIVE\_CLIENT\_INTEGRITY\_KEY\_STRING\_LEN
  - digesthmac.c, 93
- DERIVE\_SERVER\_CONFIDENTIALITY\_KEY\_STRING
  - digesthmac.c, 93
- DERIVE\_SERVER\_CONFIDENTIALITY\_KEY\_STRING\_LEN
  - digesthmac.c, 94
- DERIVE\_SERVER\_INTEGRITY\_KEY\_STRING
  - digesthmac.c, 94
- DERIVE\_SERVER\_INTEGRITY\_KEY\_STRING\_LEN
  - digesthmac.c, 94
- description
  - error.c, 99
- digest-md5.h, 87
  - \_gsasl\_digest\_md5\_client\_decode, 87
  - \_gsasl\_digest\_md5\_client\_encode, 88
  - \_gsasl\_digest\_md5\_client\_finish, 88
  - \_gsasl\_digest\_md5\_client\_start, 88
  - \_gsasl\_digest\_md5\_client\_step, 88
  - \_gsasl\_digest\_md5\_mechanism, 90
  - \_gsasl\_digest\_md5\_server\_decode, 88
  - \_gsasl\_digest\_md5\_server\_encode, 89
  - \_gsasl\_digest\_md5\_server\_finish, 89
  - \_gsasl\_digest\_md5\_server\_start, 89
  - \_gsasl\_digest\_md5\_server\_step, 89
  - GSASL\_DIGEST\_MD5\_NAME, 87
- digest-md5/client.c
  - \_Gsasl\_digest\_md5\_client\_state, 69
  - \_gsasl\_digest\_md5\_client\_decode, 69
  - \_gsasl\_digest\_md5\_client\_encode, 69
  - \_gsasl\_digest\_md5\_client\_finish, 69
  - \_gsasl\_digest\_md5\_client\_start, 69
  - \_gsasl\_digest\_md5\_client\_step, 70
  - CNONCE\_ENTROPY\_BYTES, 68
- digest-md5/free.c
  - digest\_md5\_free\_challenge, 101
  - digest\_md5\_free\_finish, 101
  - digest\_md5\_free\_response, 102
- digest-md5/mechinfo.c
  - \_gsasl\_digest\_md5\_mechanism, 151
- digest-md5/parser.c
  - CHALLENGE\_ALGORITHM, 168
  - CHALLENGE\_CHARSET, 168
  - CHALLENGE\_CIPHER, 168
  - CHALLENGE\_MAXBUF, 168
  - CHALLENGE\_NONCE, 168

- CHALLENGE\_QOP, 168
- CHALLENGE\_REALM, 168
- CHALLENGE\_STALE, 168
- CIPHER\_3DES, 169
- CIPHER\_AES\_CBC, 169
- CIPHER\_DES, 169
- CIPHER\_RC4, 169
- CIPHER\_RC4\_40, 169
- CIPHER\_RC4\_56, 169
- DEFAULT\_ALGORITHM, 168
- DEFAULT\_CHARSET, 168
- digest\_md5\_parse\_challenge, 170
- digest\_md5\_parse\_finish, 170
- digest\_md5\_parse\_response, 170
- QOP\_AUTH, 169
- QOP\_AUTH\_CONF, 169
- QOP\_AUTH\_INT, 169
- RESPONSE\_AUTHZID, 169
- RESPONSE\_CHARSET, 169
- RESPONSE\_CIPHER, 169
- RESPONSE\_CNONCE, 169
- RESPONSE\_DIGEST\_URI, 169
- RESPONSE\_MAXBUF, 169
- RESPONSE\_NC, 169
- RESPONSE\_NONCE, 169
- RESPONSE\_QOP, 169
- RESPONSE\_REALM, 169
- RESPONSE\_RESPONSE, 169
- RESPONSE\_USERNAME, 169
- RESPONSEAUTH\_RSPAUTH, 170
- digest-md5/parser.h
  - digest\_md5\_getsubopt, 172
  - digest\_md5\_parse\_challenge, 172
  - digest\_md5\_parse\_finish, 172
  - digest\_md5\_parse\_response, 173
- digest-md5/printer.c
  - digest\_md5\_print\_challenge, 176
  - digest\_md5\_print\_finish, 176
  - digest\_md5\_print\_response, 176
- digest-md5/printer.h
  - digest\_md5\_print\_challenge, 178
  - digest\_md5\_print\_finish, 178
  - digest\_md5\_print\_response, 178
- digest-md5/server.c
  - \_Gsassl\_digest\_md5\_server\_state, 193
  - \_gsasl\_digest\_md5\_server\_decode, 193
  - \_gsasl\_digest\_md5\_server\_encode, 193
  - \_gsasl\_digest\_md5\_server\_finish, 193
  - \_gsasl\_digest\_md5\_server\_start, 194
  - \_gsasl\_digest\_md5\_server\_step, 194
  - NONCE\_ENTROPY\_BYTES, 192
- digest-md5/tokens.h
  - digest\_md5\_challenge, 214
  - digest\_md5\_cipher, 214, 215
  - DIGEST\_MD5\_CIPHER\_3DES, 215
  - DIGEST\_MD5\_CIPHER\_AES\_CBC, 215
  - DIGEST\_MD5\_CIPHER\_DES, 215
  - DIGEST\_MD5\_CIPHER\_RC4, 215
- DIGEST\_MD5\_CIPHER\_RC4\_40, 215
- DIGEST\_MD5\_CIPHER\_RC4\_56, 215
- digest\_md5\_finish, 214
- DIGEST\_MD5\_LENGTH, 214
- digest\_md5\_qop, 214, 215
- DIGEST\_MD5\_QOP\_AUTH, 215
- DIGEST\_MD5\_QOP\_AUTH\_CONF, 215
- DIGEST\_MD5\_QOP\_AUTH\_INT, 215
- digest\_md5\_response, 215
- DIGEST\_MD5\_RESPONSE\_LENGTH, 214
- digest-md5/validate.c
  - digest\_md5\_validate, 218
  - digest\_md5\_validate\_challenge, 218
  - digest\_md5\_validate\_finish, 218
  - digest\_md5\_validate\_response, 219
- digest-md5/validate.h
  - digest\_md5\_validate, 220
  - digest\_md5\_validate\_challenge, 220
  - digest\_md5\_validate\_finish, 221
  - digest\_md5\_validate\_response, 221
- digest.c, 90
  - cram\_md5\_digest, 90
  - HEXCHAR, 90
- digest.h, 91
  - cram\_md5\_digest, 91
  - CRAM\_MD5\_DIGEST\_LEN, 91
- digest\_md5\_challenge, 29
  - ciphers, 29
  - digest-md5/tokens.h, 214
  - nonce, 29
  - nrealms, 29
  - qops, 29
  - realms, 30
  - servermaxbuf, 30
  - stale, 30
  - utf8, 30
- digest\_md5\_cipher
  - digest-md5/tokens.h, 214, 215
- DIGEST\_MD5\_CIPHER\_3DES
  - digest-md5/tokens.h, 215
- DIGEST\_MD5\_CIPHER\_AES\_CBC
  - digest-md5/tokens.h, 215
- DIGEST\_MD5\_CIPHER\_DES
  - digest-md5/tokens.h, 215
- DIGEST\_MD5\_CIPHER\_RC4
  - digest-md5/tokens.h, 215
- DIGEST\_MD5\_CIPHER\_RC4\_40
  - digest-md5/tokens.h, 215
- DIGEST\_MD5\_CIPHER\_RC4\_56
  - digest-md5/tokens.h, 215
- digest\_md5\_decode
  - session.c, 207
  - session.h, 208
- digest\_md5\_encode
  - session.c, 207
  - session.h, 208
- digest\_md5\_finish, 30
  - digest-md5/tokens.h, 214



- rspauth, 31
- digest\_md5\_free\_challenge
  - digest-md5/free.c, 101
  - free.h, 103
- digest\_md5\_free\_finish
  - digest-md5/free.c, 101
  - free.h, 103
- digest\_md5\_free\_response
  - digest-md5/free.c, 102
  - free.h, 103
- digest\_md5\_getsubopt
  - digest-md5/parser.h, 172
  - getsubopt.c, 104
- digest\_md5\_hashed\_password
  - Gsasl\_session, 40
- digest\_md5\_hmac
  - digesthmac.c, 95
  - digesthmac.h, 96
- DIGEST\_MD5\_LENGTH
  - digest-md5/tokens.h, 214
- digest\_md5\_parse\_challenge
  - digest-md5/parser.c, 170
  - digest-md5/parser.h, 172
- digest\_md5\_parse\_finish
  - digest-md5/parser.c, 170
  - digest-md5/parser.h, 172
- digest\_md5\_parse\_response
  - digest-md5/parser.c, 170
  - digest-md5/parser.h, 173
- digest\_md5\_print\_challenge
  - digest-md5/printer.c, 176
  - digest-md5/printer.h, 178
- digest\_md5\_print\_finish
  - digest-md5/printer.c, 176
  - digest-md5/printer.h, 178
- digest\_md5\_print\_response
  - digest-md5/printer.c, 176
  - digest-md5/printer.h, 178
- digest\_md5\_qop
  - digest-md5/tokens.h, 214, 215
- DIGEST\_MD5\_QOP\_AUTH
  - digest-md5/tokens.h, 215
- DIGEST\_MD5\_QOP\_AUTH\_CONF
  - digest-md5/tokens.h, 215
- DIGEST\_MD5\_QOP\_AUTH\_INT
  - digest-md5/tokens.h, 215
- digest\_md5\_qops2qopstr
  - qop.c, 183
  - qop.h, 184
- digest\_md5\_qopstr2qops
  - qop.c, 183
  - qop.h, 184
- digest\_md5\_response, 31
  - authzid, 32
  - cipher, 32
  - clientmaxbuf, 32
  - cnonce, 32
  - digest-md5/tokens.h, 215
  - digesturi, 32
  - nc, 32
  - nonce, 33
  - qop, 33
  - realm, 33
  - response, 33
  - username, 33
  - utf8, 33
- DIGEST\_MD5\_RESPONSE\_LENGTH
  - digest-md5/tokens.h, 214
- digest\_md5\_validate
  - digest-md5/validate.c, 218
  - digest-md5/validate.h, 220
- digest\_md5\_validate\_challenge
  - digest-md5/validate.c, 218
  - digest-md5/validate.h, 220
- digest\_md5\_validate\_finish
  - digest-md5/validate.c, 218
  - digest-md5/validate.h, 221
- digest\_md5\_validate\_response
  - digest-md5/validate.c, 219
  - digest-md5/validate.h, 221
- digesthmac.c, 92
  - A2\_POST, 92
  - A2\_PRE, 92
  - COLON, 93
  - DERIVE\_CLIENT\_CONFIDENTIALITY\_KEY\_STRING, 93
  - DERIVE\_CLIENT\_CONFIDENTIALITY\_KEY\_STRING\_LEN, 93
  - DERIVE\_CLIENT\_INTEGRITY\_KEY\_STRING, 93
  - DERIVE\_CLIENT\_INTEGRITY\_KEY\_STRING\_LEN, 93
  - DERIVE\_SERVER\_CONFIDENTIALITY\_KEY\_STRING, 93
  - DERIVE\_SERVER\_CONFIDENTIALITY\_KEY\_STRING\_LEN, 94
  - DERIVE\_SERVER\_INTEGRITY\_KEY\_STRING, 94
  - DERIVE\_SERVER\_INTEGRITY\_KEY\_STRING\_LEN, 94
  - digest\_md5\_hmac, 95
  - HEXCHAR, 94
  - MD5LEN, 94
  - QOP\_AUTH, 94
  - QOP\_AUTH\_CONF, 95
  - QOP\_AUTH\_INT, 95
- digesthmac.h, 95
  - digest\_md5\_hmac, 96
- digesturi
  - digest\_md5\_response, 32
- DIGIT
  - challenge.c, 65
- done
  - Gsasl\_mechanism\_functions, 37
- done.c, 96
  - gsasl\_done, 96
- doxygen.c, 97

- encode
  - Gsasl\_mechanism\_functions, 37
- ERR
  - error.c, 97
- ERR\_PREFIX
  - openid20/client.c, 76
- error.c, 97
  - \_, 97
  - description, 99
  - ERR, 97
  - gettext\_noop, 98
  - gsasl\_strerror, 98
  - gsasl\_strerror\_name, 98
  - N\_, 98
  - name, 99
  - rc, 99
- external.h, 100
  - \_gsasl\_external\_client\_step, 100
  - \_gsasl\_external\_mechanism, 101
  - \_gsasl\_external\_server\_step, 100
  - GSASL\_EXTERNAL\_NAME, 100
- external/client.c
  - \_gsasl\_external\_client\_step, 70
- external/mechinfo.c
  - \_gsasl\_external\_mechanism, 152
- external/server.c
  - \_gsasl\_external\_server\_step, 194
- finish
  - \_Gsasl\_digest\_md5\_client\_state, 17
  - \_Gsasl\_digest\_md5\_server\_state, 20
  - Gsasl\_mechanism\_functions, 37
- free.c, 101, 102
- free.h, 103
  - digest\_md5\_free\_challenge, 103
  - digest\_md5\_free\_finish, 103
  - digest\_md5\_free\_response, 103
- getsubopt.c, 104
  - digest\_md5\_getsubopt, 104
- gettext\_noop
  - error.c, 98
- gs2.h, 104
  - \_gsasl\_gs2\_client\_finish, 105
  - \_gsasl\_gs2\_client\_start, 105
  - \_gsasl\_gs2\_client\_step, 105
  - \_gsasl\_gs2\_krb5\_mechanism, 106
  - \_gsasl\_gs2\_server\_finish, 105
  - \_gsasl\_gs2\_server\_start, 106
  - \_gsasl\_gs2\_server\_step, 106
  - GSASL\_GS2\_KRB5\_NAME, 105
- gs2/client.c
  - \_gsasl\_gs2\_client\_finish, 71
  - \_gsasl\_gs2\_client\_start, 71
  - \_gsasl\_gs2\_client\_state, 71
  - \_gsasl\_gs2\_client\_step, 72
- gs2/mechinfo.c
  - \_gsasl\_gs2\_krb5\_mechanism, 152
- gs2/server.c
  - \_Gsasl\_gs2\_server\_state, 195
  - \_gsasl\_gs2\_server\_finish, 196
  - \_gsasl\_gs2\_server\_start, 196
  - \_gsasl\_gs2\_server\_step, 196
- gs2\_get\_oid
  - gs2helper.c, 107
  - gs2helper.h, 107
- gs2header
  - scram\_server\_state, 54
- gs2helper.c, 106
  - gs2\_get\_oid, 107
- gs2helper.h, 107
  - gs2\_get\_oid, 107
- Gsasl, 34
  - application\_hook, 34
  - cb, 34
  - client\_mechs, 34
  - gsasl.h, 116
  - n\_client\_mechs, 35
  - n\_server\_mechs, 35
  - server\_mechs, 35
- gsasl-mech.h, 107
  - Gsasl\_code\_function, 108
  - Gsasl\_done\_function, 108
  - Gsasl\_finish\_function, 109
  - Gsasl\_init\_function, 109
  - Gsasl\_mechanism, 110
  - Gsasl\_mechanism\_functions, 110
  - gsasl\_register, 111
  - Gsasl\_start\_function, 110
  - Gsasl\_step\_function, 110
- gsasl-version.h, 112
  - GSASL\_VERSION, 112
  - GSASL\_VERSION\_MAJOR, 112
  - GSASL\_VERSION\_MINOR, 113
  - GSASL\_VERSION\_NUMBER, 113
  - GSASL\_VERSION\_PATCH, 113
- gsasl.h, 114
  - \_GSASL\_API, 116
  - Gsasl, 116
  - GSASL\_ALLOW\_UNASSIGNED, 124
  - GSASL\_ANONYMOUS\_TOKEN, 120
  - GSASL\_AUTHENTICATION\_ERROR, 123
  - GSASL\_AUTHID, 120
  - GSASL\_AUTHZID, 120
  - GSASL\_BASE64\_ERROR, 123
  - gsasl\_base64\_from, 124
  - gsasl\_base64\_to, 125
  - gsasl\_callback, 125
  - Gsasl\_callback\_function, 116
  - gsasl\_callback\_hook\_get, 126
  - gsasl\_callback\_hook\_set, 126
  - gsasl\_callback\_set, 127
  - GSASL\_CB\_TLS\_EXPORTER, 121
  - GSASL\_CB\_TLS\_UNIQUE, 120
  - gsasl\_check\_version, 127
  - gsasl\_client\_mechlist, 128
  - gsasl\_client\_start, 128

- gsasl\_client\_suggest\_mechanism, [129](#)
- gsasl\_client\_support\_p, [129](#)
- GSASL\_CRYPTO\_ERROR, [123](#)
- gsasl\_decode, [129](#)
- GSASL\_DIGEST\_MD5\_HASHED\_PASSWORD, [120](#)
- gsasl\_done, [130](#)
- gsasl\_encode, [130](#)
- gsasl\_finish, [131](#)
- gsasl\_free, [131](#)
- GSASL\_GSSAPI\_ACCEPT\_SEC\_CONTEXT\_ERROR, [123](#)
- GSASL\_GSSAPI\_ACQUIRE\_CRED\_ERROR, [123](#)
- GSASL\_GSSAPI\_DECAPSULATE\_TOKEN\_ERROR, [123](#)
- GSASL\_GSSAPI\_DISPLAY\_NAME, [120](#)
- GSASL\_GSSAPI\_DISPLAY\_NAME\_ERROR, [123](#)
- GSASL\_GSSAPI\_ENCAPSULATE\_TOKEN\_ERROR, [123](#)
- GSASL\_GSSAPI\_IMPORT\_NAME\_ERROR, [123](#)
- GSASL\_GSSAPI\_INIT\_SEC\_CONTEXT\_ERROR, [123](#)
- GSASL\_GSSAPI\_INQUIRE\_MECH\_FOR\_SASLNAME\_ERROR, [123](#)
- GSASL\_GSSAPI\_RELEASE\_BUFFER\_ERROR, [123](#)
- GSASL\_GSSAPI\_RELEASE\_OID\_SET\_ERROR, [124](#)
- GSASL\_GSSAPI\_TEST\_OID\_SET\_MEMBER\_ERROR, [124](#)
- GSASL\_GSSAPI\_UNSUPPORTED\_PROTECTION\_ERROR, [123](#)
- GSASL\_GSSAPI\_UNWRAP\_ERROR, [123](#)
- GSASL\_GSSAPI\_WRAP\_ERROR, [123](#)
- Gsasl\_hash, [117](#)
- Gsasl\_hash\_length, [118](#)
- gsasl\_hash\_length, [132](#)
- GSASL\_HASH\_MAX\_SIZE, [118](#)
- GSASL\_HASH\_SHA1, [118](#)
- GSASL\_HASH\_SHA1\_SIZE, [118](#)
- GSASL\_HASH\_SHA256, [118](#)
- GSASL\_HASH\_SHA256\_SIZE, [118](#)
- gsasl\_hex\_from, [132](#)
- gsasl\_hex\_to, [133](#)
- GSASL\_HOSTNAME, [120](#)
- gsasl\_init, [133](#)
- GSASL\_INTEGRITY\_ERROR, [123](#)
- GSASL\_MALLOC\_ERROR, [123](#)
- GSASL\_MAX\_MECHANISM\_SIZE, [119](#)
- GSASL\_MECHANISM\_CALLED\_TOO\_MANY\_TIMES, [123](#)
- gsasl\_mechanism\_name, [134](#)
- gsasl\_mechanism\_name\_p, [134](#)
- GSASL\_MECHANISM\_PARSE\_ERROR, [123](#)
- Gsasl\_mechname\_limits, [118](#)
- GSASL\_MIN\_MECHANISM\_SIZE, [119](#)
- GSASL\_NEEDS\_MORE, [123](#)
- GSASL\_NO\_ANONYMOUS\_TOKEN, [123](#)
- GSASL\_NO\_AUTHID, [123](#)
- GSASL\_NO\_AUTHZID, [123](#)
- GSASL\_NO\_CALLBACK, [123](#)
- GSASL\_NO\_CB\_TLS\_EXPORTER, [123](#)
- GSASL\_NO\_CB\_TLS\_UNIQUE, [123](#)
- GSASL\_NO\_CLIENT\_CODE, [123](#)
- GSASL\_NO\_HOSTNAME, [123](#)
- GSASL\_NO\_OPENID20\_REDIRECT\_URL, [123](#)
- GSASL\_NO\_PASSCODE, [123](#)
- GSASL\_NO\_PASSWORD, [123](#)
- GSASL\_NO\_PIN, [123](#)
- GSASL\_NO\_SAML20\_IDP\_IDENTIFIER, [123](#)
- GSASL\_NO\_SAML20\_REDIRECT\_URL, [123](#)
- GSASL\_NO\_SERVER\_CODE, [123](#)
- GSASL\_NO\_SERVICE, [123](#)
- gsasl\_nonce, [134](#)
- GSASL\_OK, [123](#)
- GSASL\_OPENID20\_AUTHENTICATE\_IN\_BROWSER, [121](#)
- GSASL\_OPENID20\_OUTCOME\_DATA, [121](#)
- GSASL\_OPENID20\_REDIRECT\_URL, [120](#)
- GSASL\_PASSCODE, [120](#)
- GSASL\_PASSWORD, [120](#)
- GSASL\_PIN, [120](#)
- Gsasl\_property, [119](#)
- gsasl\_property\_fast, [135](#)
- gsasl\_property\_free, [135](#)
- gsasl\_property\_get, [136](#)
- gsasl\_property\_set, [136](#)
- gsasl\_property\_set\_raw, [137](#)
- GSASL\_QOP, [120](#)
- Gsasl\_qop, [121](#)
- GSASL\_QOP\_AUTH, [121](#)
- GSASL\_QOP\_AUTH\_CONF, [121](#)
- GSASL\_QOP\_AUTH\_INT, [121](#)
- GSASL\_QOPS, [120](#)
- gsasl\_random, [137](#)
- Gsasl\_rc, [121](#)
- GSASL\_REALM, [120](#)
- GSASL\_SAML20\_AUTHENTICATE\_IN\_BROWSER, [121](#)
- GSASL\_SAML20\_IDP\_IDENTIFIER, [120](#)
- GSASL\_SAML20\_REDIRECT\_URL, [120](#)
- gsasl\_saslprep, [138](#)
- GSASL\_SASLPREP\_ERROR, [123](#)
- Gsasl\_saslprep\_flags, [124](#)
- GSASL\_SCRAM\_ITER, [120](#)
- GSASL\_SCRAM\_SALT, [120](#)
- GSASL\_SCRAM\_SALTED\_PASSWORD, [120](#)
- gsasl\_scram\_secrets\_from\_password, [138](#)
- gsasl\_scram\_secrets\_from\_salted\_password, [139](#)
- GSASL\_SCRAM\_SERVERKEY, [120](#)
- GSASL\_SCRAM\_STOREDKEY, [120](#)
- GSASL\_SECURID\_SERVER\_NEED\_ADDITIONAL\_PASSCODE, [123](#)
- GSASL\_SECURID\_SERVER\_NEED\_NEW\_PIN, [123](#)
- gsasl\_server\_mechlist, [139](#)

- gsasl\_server\_start, [140](#)
- gsasl\_server\_support\_p, [140](#)
- GSASL\_SERVICE, [120](#)
- Gsasl\_session, [117](#)
- gsasl\_session\_hook\_get, [141](#)
- gsasl\_session\_hook\_set, [141](#)
- gsasl\_simple\_getpass, [142](#)
- gsasl\_step, [142](#)
- gsasl\_step64, [143](#)
- gsasl\_strerror, [143](#)
- gsasl\_strerror\_name, [144](#)
- GSASL\_SUGGESTED\_PIN, [120](#)
- GSASL\_UNKNOWN\_MECHANISM, [123](#)
- GSASL\_VALIDATE\_ANONYMOUS, [121](#)
- GSASL\_VALIDATE\_EXTERNAL, [121](#)
- GSASL\_VALIDATE\_GSSAPI, [121](#)
- GSASL\_VALIDATE\_OPENID20, [121](#)
- GSASL\_VALIDATE\_SAML20, [121](#)
- GSASL\_VALIDATE\_SECURID, [121](#)
- GSASL\_VALIDATE\_SIMPLE, [121](#)
- GSASL\_ALLOW\_UNASSIGNED
  - gsasl.h, [124](#)
- GSASL\_ANONYMOUS\_NAME
  - anonymous.h, [57](#)
- GSASL\_ANONYMOUS\_TOKEN
  - gsasl.h, [120](#)
- GSASL\_AUTHENTICATION\_ERROR
  - gsasl.h, [123](#)
- GSASL\_AUTHID
  - gsasl.h, [120](#)
- GSASL\_AUTHZID
  - gsasl.h, [120](#)
- GSASL\_BASE64\_ERROR
  - gsasl.h, [123](#)
- gsasl\_base64\_from
  - base64.c, [59](#)
  - gsasl.h, [124](#)
- gsasl\_base64\_to
  - base64.c, [59](#)
  - gsasl.h, [125](#)
- gsasl\_callback
  - callback.c, [61](#)
  - gsasl.h, [125](#)
- Gsasl\_callback\_function
  - gsasl.h, [116](#)
- gsasl\_callback\_hook\_get
  - callback.c, [62](#)
  - gsasl.h, [126](#)
- gsasl\_callback\_hook\_set
  - callback.c, [62](#)
  - gsasl.h, [126](#)
- gsasl\_callback\_set
  - callback.c, [63](#)
  - gsasl.h, [127](#)
- GSASL\_CB\_TLS\_EXPORTER
  - gsasl.h, [121](#)
- GSASL\_CB\_TLS\_UNIQUE
  - gsasl.h, [120](#)
- gsasl\_check\_version
  - gsasl.h, [127](#)
  - version.c, [222](#)
- gsasl\_client\_mechlist
  - gsasl.h, [128](#)
  - listmech.c, [146](#)
- gsasl\_client\_start
  - gsasl.h, [128](#)
  - xstart.c, [230](#)
- gsasl\_client\_suggest\_mechanism
  - gsasl.h, [129](#)
  - suggest.c, [209](#)
- gsasl\_client\_support\_p
  - gsasl.h, [129](#)
  - supportp.c, [210](#)
- Gsasl\_code\_function
  - gsasl-mech.h, [108](#)
- GSASL\_CRAM\_MD5\_NAME
  - cram-md5.h, [82](#)
- GSASL\_CRYPT0\_ERROR
  - gsasl.h, [123](#)
- gsasl\_decode
  - gsasl.h, [129](#)
  - xcode.c, [228](#)
- GSASL\_DIGEST\_MD5\_HASHED\_PASSWORD
  - gsasl.h, [120](#)
- GSASL\_DIGEST\_MD5\_NAME
  - digest-md5.h, [87](#)
- gsasl\_done
  - done.c, [96](#)
  - gsasl.h, [130](#)
- Gsasl\_done\_function
  - gsasl-mech.h, [108](#)
- gsasl\_encode
  - gsasl.h, [130](#)
  - xcode.c, [228](#)
- GSASL\_EXTERNAL\_NAME
  - external.h, [100](#)
- gsasl\_finish
  - gsasl.h, [131](#)
  - xfinish.c, [229](#)
- Gsasl\_finish\_function
  - gsasl-mech.h, [109](#)
- gsasl\_free
  - gsasl.h, [131](#)
  - src/free.c, [102](#)
- GSASL\_GS2\_KRB5\_NAME
  - gs2.h, [105](#)
- GSASL\_GSSAPI\_ACCEPT\_SEC\_CONTEXT\_ERROR
  - gsasl.h, [123](#)
- GSASL\_GSSAPI\_ACQUIRE\_CRED\_ERROR
  - gsasl.h, [123](#)
- GSASL\_GSSAPI\_DECAPSULATE\_TOKEN\_ERROR
  - gsasl.h, [123](#)
- GSASL\_GSSAPI\_DISPLAY\_NAME
  - gsasl.h, [120](#)
- GSASL\_GSSAPI\_DISPLAY\_NAME\_ERROR
  - gsasl.h, [123](#)

- GSASL\_GSSAPI\_ENCAPSULATE\_TOKEN\_ERROR
  - gsasl.h, [123](#)
- GSASL\_GSSAPI\_IMPORT\_NAME\_ERROR
  - gsasl.h, [123](#)
- GSASL\_GSSAPI\_INIT\_SEC\_CONTEXT\_ERROR
  - gsasl.h, [123](#)
- GSASL\_GSSAPI\_INQUIRE\_MECH\_FOR\_SASLNAME\_ERROR
  - gsasl.h, [123](#)
- GSASL\_GSSAPI\_NAME
  - x-gssapi.h, [223](#)
- GSASL\_GSSAPI\_RELEASE\_BUFFER\_ERROR
  - gsasl.h, [123](#)
- GSASL\_GSSAPI\_RELEASE\_OID\_SET\_ERROR
  - gsasl.h, [124](#)
- GSASL\_GSSAPI\_TEST\_OID\_SET\_MEMBER\_ERROR
  - gsasl.h, [124](#)
- GSASL\_GSSAPI\_UNSUPPORTED\_PROTECTION\_ERROR
  - gsasl.h, [123](#)
- GSASL\_GSSAPI\_UNWRAP\_ERROR
  - gsasl.h, [123](#)
- GSASL\_GSSAPI\_WRAP\_ERROR
  - gsasl.h, [123](#)
- Gsasl\_hash
  - gsasl.h, [117](#)
- Gsasl\_hash\_length
  - gsasl.h, [118](#)
- gsasl\_hash\_length
  - crypto.c, [84](#)
  - gsasl.h, [132](#)
- GSASL\_HASH\_MAX\_SIZE
  - gsasl.h, [118](#)
- GSASL\_HASH\_SHA1
  - gsasl.h, [118](#)
- GSASL\_HASH\_SHA1\_SIZE
  - gsasl.h, [118](#)
- GSASL\_HASH\_SHA256
  - gsasl.h, [118](#)
- GSASL\_HASH\_SHA256\_SIZE
  - gsasl.h, [118](#)
- gsasl\_hex\_from
  - base64.c, [60](#)
  - gsasl.h, [132](#)
- gsasl\_hex\_to
  - base64.c, [60](#)
  - gsasl.h, [133](#)
- GSASL\_HOSTNAME
  - gsasl.h, [120](#)
- gsasl\_init
  - gsasl.h, [133](#)
  - init.c, [145](#)
- Gsasl\_init\_function
  - gsasl-mech.h, [109](#)
- GSASL\_INTEGRITY\_ERROR
  - gsasl.h, [123](#)
- GSASL\_LOGIN\_NAME
  - login.h, [147](#)
- GSASL\_MALLOC\_ERROR
  - gsasl.h, [123](#)
- GSASL\_MAX\_MECHANISM\_SIZE
  - gsasl.h, [119](#)
- Gsasl\_mechanism
  - 35
  - client, [36](#)
  - gsasl-mech.h, [110](#)
  - name, [36](#)
  - server, [36](#)
- GSASL\_MECHANISM\_CALLED\_TOO\_MANY\_TIMES
  - gsasl.h, [123](#)
- Gsasl\_mechanism\_functions
  - 36
  - decode, [37](#)
  - done, [37](#)
  - encode, [37](#)
  - finish, [37](#)
  - gsasl-mech.h, [110](#)
  - init, [37](#)
  - start, [38](#)
  - step, [38](#)
- gsasl\_mechanism\_name
  - gsasl.h, [134](#)
  - mechname.c, [157](#)
- gsasl\_mechanism\_name\_p
  - gsasl.h, [134](#)
  - suggest.c, [209](#)
- GSASL\_MECHANISM\_PARSE\_ERROR
  - gsasl.h, [123](#)
- Gsasl\_mechname\_limits
  - gsasl.h, [118](#)
- GSASL\_MIN\_MECHANISM\_SIZE
  - gsasl.h, [119](#)
- GSASL\_NEEDS\_MORE
  - gsasl.h, [123](#)
- GSASL\_NO\_ANONYMOUS\_TOKEN
  - gsasl.h, [123](#)
- GSASL\_NO\_AUTHID
  - gsasl.h, [123](#)
- GSASL\_NO\_AUTHZID
  - gsasl.h, [123](#)
- GSASL\_NO\_CALLBACK
  - gsasl.h, [123](#)
- GSASL\_NO\_CB\_TLS\_EXPORTER
  - gsasl.h, [123](#)
- GSASL\_NO\_CB\_TLS\_UNIQUE
  - gsasl.h, [123](#)
- GSASL\_NO\_CLIENT\_CODE
  - gsasl.h, [123](#)
- GSASL\_NO\_HOSTNAME
  - gsasl.h, [123](#)
- GSASL\_NO\_OPENID20\_REDIRECT\_URL
  - gsasl.h, [123](#)
- GSASL\_NO\_PASSCODE
  - gsasl.h, [123](#)
- GSASL\_NO\_PASSWORD
  - gsasl.h, [123](#)
- GSASL\_NO\_PIN
  - gsasl.h, [123](#)
- GSASL\_NO\_SAML20\_IDP\_IDENTIFIER
  - gsasl.h, [123](#)

- GSASL\_NO\_SAML20\_REDIRECT\_URL
  - gsasl.h, [123](#)
- GSASL\_NO\_SERVER\_CODE
  - gsasl.h, [123](#)
- GSASL\_NO\_SERVICE
  - gsasl.h, [123](#)
- gsasl\_nonce
  - crypto.c, [84](#)
  - gsasl.h, [134](#)
- GSASL\_NTLM\_NAME
  - x-ntlm.h, [226](#)
- GSASL\_OK
  - gsasl.h, [123](#)
- GSASL\_OPENID20\_AUTHENTICATE\_IN\_BROWSER
  - gsasl.h, [121](#)
- GSASL\_OPENID20\_NAME
  - openid20.h, [165](#)
- GSASL\_OPENID20\_OUTCOME\_DATA
  - gsasl.h, [121](#)
- GSASL\_OPENID20\_REDIRECT\_URL
  - gsasl.h, [120](#)
- GSASL\_PASSCODE
  - gsasl.h, [120](#)
- GSASL\_PASSWORD
  - gsasl.h, [120](#)
- GSASL\_PIN
  - gsasl.h, [120](#)
- GSASL\_PLAIN\_NAME
  - plain.h, [175](#)
- Gsasl\_property
  - gsasl.h, [119](#)
- gsasl\_property\_fast
  - gsasl.h, [135](#)
  - property.c, [180](#)
- gsasl\_property\_free
  - gsasl.h, [135](#)
  - property.c, [180](#)
- gsasl\_property\_get
  - gsasl.h, [136](#)
  - property.c, [181](#)
- gsasl\_property\_set
  - gsasl.h, [136](#)
  - property.c, [181](#)
- gsasl\_property\_set\_raw
  - gsasl.h, [137](#)
  - property.c, [182](#)
- GSASL\_QOP
  - gsasl.h, [120](#)
- Gsasl\_qop
  - gsasl.h, [121](#)
- GSASL\_QOP\_AUTH
  - gsasl.h, [121](#)
- GSASL\_QOP\_AUTH\_CONF
  - gsasl.h, [121](#)
- GSASL\_QOP\_AUTH\_INT
  - gsasl.h, [121](#)
- GSASL\_QOPS
  - gsasl.h, [120](#)
- gsasl\_random
  - crypto.c, [85](#)
  - gsasl.h, [137](#)
- Gsasl\_rc
  - gsasl.h, [121](#)
- GSASL\_REALM
  - gsasl.h, [120](#)
- gsasl\_register
  - gsasl-mech.h, [111](#)
  - register.c, [184](#)
- GSASL\_SAML20\_AUTHENTICATE\_IN\_BROWSER
  - gsasl.h, [121](#)
- GSASL\_SAML20\_IDP\_IDENTIFIER
  - gsasl.h, [120](#)
- GSASL\_SAML20\_NAME
  - saml20.h, [185](#)
- GSASL\_SAML20\_REDIRECT\_URL
  - gsasl.h, [120](#)
- gsasl\_saslprep
  - gsasl.h, [138](#)
  - saslprep.c, [187](#)
- GSASL\_SASLPREP\_ERROR
  - gsasl.h, [123](#)
- Gsasl\_saslprep\_flags
  - gsasl.h, [124](#)
- GSASL\_SCRAM\_ITER
  - gsasl.h, [120](#)
- GSASL\_SCRAM\_SALT
  - gsasl.h, [120](#)
- GSASL\_SCRAM\_SALTED\_PASSWORD
  - gsasl.h, [120](#)
- gsasl\_scram\_secrets\_from\_password
  - crypto.c, [85](#)
  - gsasl.h, [138](#)
- gsasl\_scram\_secrets\_from\_salted\_password
  - crypto.c, [86](#)
  - gsasl.h, [139](#)
- GSASL\_SCRAM\_SERVERKEY
  - gsasl.h, [120](#)
- GSASL\_SCRAM\_STOREDKEY
  - gsasl.h, [120](#)
- GSASL\_SECURID\_NAME
  - securid.h, [188](#)
- GSASL\_SECURID\_SERVER\_NEED\_ADDITIONAL\_PASSCODE
  - gsasl.h, [123](#)
- GSASL\_SECURID\_SERVER\_NEED\_NEW\_PIN
  - gsasl.h, [123](#)
- gsasl\_server\_mechlist
  - gsasl.h, [139](#)
  - listmech.c, [146](#)
- gsasl\_server\_start
  - gsasl.h, [140](#)
  - xstart.c, [230](#)
- gsasl\_server\_support\_p
  - gsasl.h, [140](#)
  - supportp.c, [211](#)
- GSASL\_SERVICE
  - gsasl.h, [120](#)

- Gsasl\_session, 38
  - anonymous\_token, 39
  - application\_hook, 39
  - authid, 39
  - authzid, 39
  - cb\_tls\_exporter, 40
  - cb\_tls\_unique, 40
  - clientp, 40
  - ctx, 40
  - digest\_md5\_hashed\_password, 40
  - gsasl.h, 117
  - gssapi\_display\_name, 40
  - hostname, 41
  - mech, 41
  - mech\_data, 41
  - openid20\_outcome\_data, 41
  - openid20\_redirect\_url, 41
  - passcode, 41
  - password, 42
  - pin, 42
  - qop, 42
  - qops, 42
  - realm, 42
  - saml20\_idp\_identifier, 42
  - saml20\_redirect\_url, 43
  - scram\_iter, 43
  - scram\_salt, 43
  - scram\_salted\_password, 43
  - scram\_serverkey, 43
  - scram\_storedkey, 43
  - service, 44
  - suggestedpin, 44
- gsasl\_session\_hook\_get
  - callback.c, 63
  - gsasl.h, 141
- gsasl\_session\_hook\_set
  - callback.c, 64
  - gsasl.h, 141
- gsasl\_simple\_getpass
  - gsasl.h, 142
  - md5pwd.c, 149
- Gsasl\_start\_function
  - gsasl-mech.h, 110
- gsasl\_step
  - gsasl.h, 142
  - xstep.c, 231
- gsasl\_step64
  - gsasl.h, 143
  - xstep.c, 232
- Gsasl\_step\_function
  - gsasl-mech.h, 110
- gsasl\_strerror
  - error.c, 98
  - gsasl.h, 143
- gsasl\_strerror\_name
  - error.c, 98
  - gsasl.h, 144
- GSASL\_SUGGESTED\_PIN
  - gsasl.h, 120
- GSASL\_UNKNOWN\_MECHANISM
  - gsasl.h, 123
- GSASL\_VALIDATE\_ANONYMOUS
  - gsasl.h, 121
- GSASL\_VALIDATE\_EXTERNAL
  - gsasl.h, 121
- GSASL\_VALIDATE\_GSSAPI
  - gsasl.h, 121
- GSASL\_VALIDATE\_OPENID20
  - gsasl.h, 121
- GSASL\_VALIDATE\_SAML20
  - gsasl.h, 121
- GSASL\_VALIDATE\_SECURID
  - gsasl.h, 121
- GSASL\_VALIDATE\_SIMPLE
  - gsasl.h, 121
- GSASL\_VERSION
  - gsasl-version.h, 112
- GSASL\_VERSION\_MAJOR
  - gsasl-version.h, 112
- GSASL\_VERSION\_MINOR
  - gsasl-version.h, 113
- GSASL\_VERSION\_NUMBER
  - gsasl-version.h, 113
- GSASL\_VERSION\_PATCH
  - gsasl-version.h, 113
- gssapi/client.c
  - \_Gsasl\_gssapi\_client\_state, 73
  - \_gsasl\_gssapi\_client\_decode, 73
  - \_gsasl\_gssapi\_client\_encode, 73
  - \_gsasl\_gssapi\_client\_finish, 73
  - \_gsasl\_gssapi\_client\_start, 73
  - \_gsasl\_gssapi\_client\_step, 74
- gssapi/mechinfo.c
  - \_gsasl\_gssapi\_mechanism, 153
- gssapi/server.c
  - \_Gsasl\_gssapi\_server\_state, 197
  - \_gsasl\_gssapi\_server\_finish, 197
  - \_gsasl\_gssapi\_server\_start, 197
  - \_gsasl\_gssapi\_server\_step, 197
- gssapi\_display\_name
  - Gsasl\_session, 40
- hash
  - scram\_client\_state, 50
  - scram\_server\_state, 55
- HEXCHAR
  - digest.c, 90
  - digesthmac.c, 94
- hostname
  - Gsasl\_session, 41
- init
  - Gsasl\_mechanism\_functions, 37
- init.c, 144
  - gsasl\_init, 145
- internal.h, 145
- iter



- scram\_server\_first, 52
- kcc
  - \_Gsasl\_digest\_md5\_client\_state, 18
  - \_Gsasl\_digest\_md5\_server\_state, 20
- kcs
  - \_Gsasl\_digest\_md5\_client\_state, 18
  - \_Gsasl\_digest\_md5\_server\_state, 20
- kic
  - \_Gsasl\_digest\_md5\_client\_state, 18
  - \_Gsasl\_digest\_md5\_server\_state, 20
- kis
  - \_Gsasl\_digest\_md5\_client\_state, 18
  - \_Gsasl\_digest\_md5\_server\_state, 20
- latin1toutf8
  - nonascii.c, 162
  - nonascii.h, 163
- listmech.c, 146
  - gsasl\_client\_mechlist, 146
  - gsasl\_server\_mechlist, 146
- login.h, 147
  - \_gsasl\_login\_client\_finish, 147
  - \_gsasl\_login\_client\_start, 148
  - \_gsasl\_login\_client\_step, 148
  - \_gsasl\_login\_mechanism, 149
  - \_gsasl\_login\_server\_finish, 148
  - \_gsasl\_login\_server\_start, 148
  - \_gsasl\_login\_server\_step, 148
  - GSASL\_LOGIN\_NAME, 147
- login/client.c
  - \_gsasl\_login\_client\_finish, 74
  - \_gsasl\_login\_client\_start, 75
  - \_gsasl\_login\_client\_step, 75
- login/mechinfo.c
  - \_gsasl\_login\_mechanism, 153
- login/server.c
  - \_gsasl\_login\_server\_finish, 199
  - \_gsasl\_login\_server\_start, 199
  - \_gsasl\_login\_server\_step, 199
  - CHALLENGE\_PASSWORD, 198
  - CHALLENGE\_USERNAME, 198
- MAC\_DATA\_LEN
  - session.c, 206
- MAC\_HMAC\_LEN
  - session.c, 206
- MAC\_MSG\_TYPE
  - session.c, 206
- MAC\_MSG\_TYPE\_LEN
  - session.c, 206
- MAC\_SEQNUM\_LEN
  - session.c, 206
- main
  - test-parser.c, 212
- MD5LEN
  - cram-md5/server.c, 191
  - digesthmac.c, 94
  - session.c, 207
- md5pwd.c, 149
  - gsasl\_simple\_getpass, 149
- mech
  - Gsasl\_session, 41
- mech\_data
  - Gsasl\_session, 41
- mech\_oid
  - \_Gsasl\_gs2\_server\_state, 24
  - \_gsasl\_gs2\_client\_state, 22
- mechinfo.c, 150–156
- mechname.c, 157
  - gsasl\_mechanism\_name, 157
- mechtools.c, 157
  - \_gsasl\_gs2\_generate\_header, 158
  - \_gsasl\_hash, 158
  - \_gsasl\_hex\_decode, 158
  - \_gsasl\_hex\_encode, 158
  - \_gsasl\_hex\_p, 159
  - \_gsasl\_hmac, 159
  - \_gsasl\_parse\_gs2\_header, 159
  - \_gsasl\_pbkdf2, 159
- mechtools.h, 160
  - \_gsasl\_gs2\_generate\_header, 160
  - \_gsasl\_hash, 160
  - \_gsasl\_hex\_decode, 161
  - \_gsasl\_hex\_encode, 161
  - \_gsasl\_hex\_p, 161
  - \_gsasl\_hmac, 161
  - \_gsasl\_parse\_gs2\_header, 161
  - \_gsasl\_pbkdf2, 162
- N\_
  - error.c, 98
- n\_client\_mechs
  - Gsasl, 35
- n\_server\_mechs
  - Gsasl, 35
- name
  - error.c, 99
  - Gsasl\_mechanism, 36
- nc
  - digest\_md5\_response, 32
- nonascii.c, 162
  - latin1toutf8, 162
  - utf8tolatin1ifpossible, 162
- nonascii.h, 163
  - latin1toutf8, 163
  - utf8tolatin1ifpossible, 163
- nonce
  - digest\_md5\_challenge, 29
  - digest\_md5\_response, 33
  - scram\_client\_final, 47
  - scram\_server\_first, 52
- NONCE\_ENTROPY\_BYTES
  - digest-md5/server.c, 192
- NONCELEN
  - challenge.c, 65
- nrealms
  - digest\_md5\_challenge, 29



- ntlm.c, 163
  - \_Gsasl\_ntlm\_state, 164
  - \_gsasl\_ntlm\_client\_finish, 164
  - \_gsasl\_ntlm\_client\_start, 164
  - \_gsasl\_ntlm\_client\_step, 164
- ntlm/mechinfo.c
  - \_gsasl\_ntlm\_mechanism, 154
- openid20.h, 165
  - \_gsasl\_openid20\_client\_finish, 166
  - \_gsasl\_openid20\_client\_start, 166
  - \_gsasl\_openid20\_client\_step, 166
  - \_gsasl\_openid20\_mechanism, 167
  - \_gsasl\_openid20\_server\_finish, 166
  - \_gsasl\_openid20\_server\_start, 166
  - \_gsasl\_openid20\_server\_step, 166
  - GSASL\_OPENID20\_NAME, 165
- openid20/client.c
  - \_gsasl\_openid20\_client\_finish, 76
  - \_gsasl\_openid20\_client\_start, 76
  - \_gsasl\_openid20\_client\_step, 76
  - ERR\_PREFIX, 76
- openid20/mechinfo.c
  - \_gsasl\_openid20\_mechanism, 154
- openid20/server.c
  - \_gsasl\_openid20\_server\_finish, 200
  - \_gsasl\_openid20\_server\_start, 200
  - \_gsasl\_openid20\_server\_step, 200
- openid20\_client\_state, 44
  - step, 44
- openid20\_outcome\_data
  - Gsasl\_session, 41
- openid20\_redirect\_url
  - Gsasl\_session, 41
- openid20\_server\_state, 45
  - allow\_error\_step, 45
  - step, 45
- parser.c, 167, 171
- parser.h, 172, 173
- PASSCODE
  - securid/client.c, 80
  - securid/server.c, 204
- passcode
  - Gsasl\_session, 41
- password
  - \_Gsasl\_login\_server\_state, 27
  - Gsasl\_session, 42
- PIN
  - securid/client.c, 80
  - securid/server.c, 204
- pin
  - Gsasl\_session, 42
- plain.h, 174
  - \_gsasl\_plain\_client\_step, 175
  - \_gsasl\_plain\_mechanism, 175
  - \_gsasl\_plain\_server\_step, 175
  - GSASL\_PLAIN\_NAME, 175
- plain/client.c
  - \_gsasl\_plain\_client\_step, 77
- plain/mechinfo.c
  - \_gsasl\_plain\_mechanism, 155
- plain/server.c
  - \_gsasl\_plain\_server\_step, 201
- plus
  - scram\_client\_state, 50
  - scram\_server\_state, 55
- printer.c, 176, 177
- printer.h, 178, 179
- proof
  - scram\_client\_final, 47
- property.c, 180
  - gsasl\_property\_fast, 180
  - gsasl\_property\_free, 180
  - gsasl\_property\_get, 181
  - gsasl\_property\_set, 181
  - gsasl\_property\_set\_raw, 182
- qop
  - \_Gsasl\_gssapi\_client\_state, 25
  - digest\_md5\_response, 33
  - Gsasl\_session, 42
- qop.c, 183
  - digest\_md5\_qops2qopstr, 183
  - digest\_md5\_qopstr2qops, 183
- qop.h, 183
  - digest\_md5\_qops2qopstr, 184
  - digest\_md5\_qopstr2qops, 184
- QOP\_AUTH
  - digest-md5/parser.c, 169
  - digestthmac.c, 94
- QOP\_AUTH\_CONF
  - digest-md5/parser.c, 169
  - digestthmac.c, 95
- QOP\_AUTH\_INT
  - digest-md5/parser.c, 169
  - digestthmac.c, 95
- qops
  - digest\_md5\_challenge, 29
  - Gsasl\_session, 42
- rc
  - error.c, 99
- readseqnum
  - \_Gsasl\_digest\_md5\_client\_state, 18
  - \_Gsasl\_digest\_md5\_server\_state, 21
- realm
  - digest\_md5\_response, 33
  - Gsasl\_session, 42
- realms
  - digest\_md5\_challenge, 30
- register.c, 184
  - gsasl\_register, 184
- response
  - \_Gsasl\_digest\_md5\_client\_state, 18
  - \_Gsasl\_digest\_md5\_server\_state, 21
  - digest\_md5\_response, 33
- RESPONSE\_AUTHZID

- digest-md5/parser.c, 169
- RESPONSE\_CHARSET
  - digest-md5/parser.c, 169
- RESPONSE\_CIPHER
  - digest-md5/parser.c, 169
- RESPONSE\_CNONCE
  - digest-md5/parser.c, 169
- RESPONSE\_DIGEST\_URI
  - digest-md5/parser.c, 169
- RESPONSE\_MAXBUF
  - digest-md5/parser.c, 169
- RESPONSE\_NC
  - digest-md5/parser.c, 169
- RESPONSE\_NONCE
  - digest-md5/parser.c, 169
- RESPONSE\_QOP
  - digest-md5/parser.c, 169
- RESPONSE\_REALM
  - digest-md5/parser.c, 169
- RESPONSE\_RESPONSE
  - digest-md5/parser.c, 169
- RESPONSE\_USERNAME
  - digest-md5/parser.c, 169
- RESPONSEAUTH\_RSPAUTH
  - digest-md5/parser.c, 170
- rspauth
  - digest\_md5\_finish, 31
- salt
  - scram\_server\_first, 52
- saml20.h, 185
  - \_gsasl\_saml20\_client\_finish, 185
  - \_gsasl\_saml20\_client\_start, 186
  - \_gsasl\_saml20\_client\_step, 186
  - \_gsasl\_saml20\_mechanism, 187
  - \_gsasl\_saml20\_server\_finish, 186
  - \_gsasl\_saml20\_server\_start, 186
  - \_gsasl\_saml20\_server\_step, 186
  - GSASL\_SAML20\_NAME, 185
- saml20/client.c
  - \_gsasl\_saml20\_client\_finish, 78
  - \_gsasl\_saml20\_client\_start, 78
  - \_gsasl\_saml20\_client\_step, 78
- saml20/mechinfo.c
  - \_gsasl\_saml20\_mechanism, 156
- saml20/server.c
  - \_gsasl\_saml20\_server\_finish, 202
  - \_gsasl\_saml20\_server\_start, 202
  - \_gsasl\_saml20\_server\_step, 202
- saml20\_client\_state, 45
  - step, 46
- saml20\_idp\_identifier
  - Gsasl\_session, 42
- saml20\_redirect\_url
  - Gsasl\_session, 43
- saml20\_server\_state, 46
  - step, 46
- SASL\_INTEGRITY\_PREFIX\_LENGTH
  - session.c, 207
- saslprep.c, 187
  - gsasl\_saslprep, 187
- scram.h, 188
- scram/client.c
  - \_gsasl\_scram\_client\_finish, 79
  - \_gsasl\_scram\_client\_step, 79
  - CNONCE\_ENTROPY\_BYTES, 79
- scram/parser.c
  - scram\_parse\_client\_final, 171
  - scram\_parse\_client\_first, 171
  - scram\_parse\_server\_final, 171
  - scram\_parse\_server\_first, 171
- scram/parser.h
  - scram\_parse\_client\_final, 173
  - scram\_parse\_client\_first, 173
  - scram\_parse\_server\_final, 174
  - scram\_parse\_server\_first, 174
- scram/printer.c
  - scram\_print\_client\_final, 177
  - scram\_print\_client\_first, 177
  - scram\_print\_server\_final, 177
  - scram\_print\_server\_first, 177
- scram/printer.h
  - scram\_print\_client\_final, 179
  - scram\_print\_client\_first, 179
  - scram\_print\_server\_final, 179
  - scram\_print\_server\_first, 179
- scram/server.c
  - \_gsasl\_scram\_server\_finish, 203
  - \_gsasl\_scram\_server\_step, 203
  - DEFAULT\_SALT\_BYTES, 203
  - SNONCE\_ENTROPY\_BYTES, 203
- scram/tokens.h
  - scram\_free\_client\_final, 216
  - scram\_free\_client\_first, 216
  - scram\_free\_server\_final, 216
  - scram\_free\_server\_first, 216
- scram/validate.c
  - scram\_valid\_client\_final, 219
  - scram\_valid\_client\_first, 219
  - scram\_valid\_server\_final, 220
  - scram\_valid\_server\_first, 220
- scram/validate.h
  - scram\_valid\_client\_final, 221
  - scram\_valid\_client\_first, 221
  - scram\_valid\_server\_final, 222
  - scram\_valid\_server\_first, 222
- scram\_client\_final, 46
  - cbind, 47
  - nonce, 47
  - proof, 47
- scram\_client\_first, 47
  - authzid, 48
  - cbflag, 48
  - cbname, 48
  - client\_nonce, 48
  - username, 48
- scram\_client\_state, 49

- authmessage, 49
- cf, 49
- cfmb, 49
- cl, 50
- hash, 50
- plus, 50
- serversignature, 50
- sf, 50
- sl, 50
- step, 51
- scram\_free\_client\_final
  - scram/tokens.h, 216
  - tokens.c, 212
- scram\_free\_client\_first
  - scram/tokens.h, 216
  - tokens.c, 212
- scram\_free\_server\_final
  - scram/tokens.h, 216
  - tokens.c, 212
- scram\_free\_server\_first
  - scram/tokens.h, 216
  - tokens.c, 213
- scram\_iter
  - Gsasl\_session, 43
- scram\_parse\_client\_final
  - scram/parser.c, 171
  - scram/parser.h, 173
- scram\_parse\_client\_first
  - scram/parser.c, 171
  - scram/parser.h, 173
- scram\_parse\_server\_final
  - scram/parser.c, 171
  - scram/parser.h, 174
- scram\_parse\_server\_first
  - scram/parser.c, 171
  - scram/parser.h, 174
- scram\_print\_client\_final
  - scram/printer.c, 177
  - scram/printer.h, 179
- scram\_print\_client\_first
  - scram/printer.c, 177
  - scram/printer.h, 179
- scram\_print\_server\_final
  - scram/printer.c, 177
  - scram/printer.h, 179
- scram\_print\_server\_first
  - scram/printer.c, 177
  - scram/printer.h, 179
- scram\_salt
  - Gsasl\_session, 43
- scram\_salted\_password
  - Gsasl\_session, 43
- scram\_server\_final, 51
  - verifier, 51
- scram\_server\_first, 52
  - iter, 52
  - nonce, 52
  - salt, 52
  - scram\_server\_state, 53
  - authmessage, 53
  - cb, 53
  - cbind, 53
  - cblen, 54
  - cf, 54
  - cfmb\_str, 54
  - cl, 54
  - clientproof, 54
  - gs2header, 54
  - hash, 55
  - plus, 55
  - serverkey, 55
  - sf, 55
  - sf\_str, 55
  - sl, 55
  - snonce, 56
  - step, 56
  - storedkey, 56
- scram\_serverkey
  - Gsasl\_session, 43
- scram\_storedkey
  - Gsasl\_session, 43
- scram\_valid\_client\_final
  - scram/validate.c, 219
  - scram/validate.h, 221
- scram\_valid\_client\_first
  - scram/validate.c, 219
  - scram/validate.h, 221
- scram\_valid\_server\_final
  - scram/validate.c, 220
  - scram/validate.h, 222
- scram\_valid\_server\_first
  - scram/validate.c, 220
  - scram/validate.h, 222
- secret
  - \_Gsasl\_digest\_md5\_client\_state, 19
  - \_Gsasl\_digest\_md5\_server\_state, 21
- securid.h, 188
  - \_gsasl\_securid\_client\_finish, 188
  - \_gsasl\_securid\_client\_start, 189
  - \_gsasl\_securid\_client\_step, 189
  - \_gsasl\_securid\_mechanism, 189
  - \_gsasl\_securid\_server\_step, 189
  - GSASL\_SECURID\_NAME, 188
- securid/client.c
  - \_gsasl\_securid\_client\_finish, 80
  - \_gsasl\_securid\_client\_start, 81
  - \_gsasl\_securid\_client\_step, 81
  - PASSCODE, 80
  - PIN, 80
- securid/mechinfo.c
  - \_gsasl\_securid\_mechanism, 156
- securid/server.c
  - \_gsasl\_securid\_server\_step, 205
  - PASSCODE, 204
  - PIN, 204
- sendseqnum

- [\\_Gsasl\\_digest\\_md5\\_client\\_state](#), 19
  - [\\_Gsasl\\_digest\\_md5\\_server\\_state](#), 21
- server
  - [Gsasl\\_mechanism](#), 36
- [server.c](#), 190, 192, 194–196, 198, 199, 201, 202, 204
- SERVER\_KEY
  - [crypto.c](#), 84
- server\_mechs
  - [Gsasl](#), 35
- serverkey
  - [scram\\_server\\_state](#), 55
- servermaxbuf
  - [digest\\_md5\\_challenge](#), 30
- serversignature
  - [scram\\_client\\_state](#), 50
- service
  - [\\_Gsasl\\_gssapi\\_client\\_state](#), 25
  - [\\_gsasl\\_gs2\\_client\\_state](#), 22
  - [Gsasl\\_session](#), 44
- [session.c](#), 205
  - [C2I](#), 205
  - [digest\\_md5\\_decode](#), 207
  - [digest\\_md5\\_encode](#), 207
  - [MAC\\_DATA\\_LEN](#), 206
  - [MAC\\_HMAC\\_LEN](#), 206
  - [MAC\\_MSG\\_TYPE](#), 206
  - [MAC\\_MSG\\_TYPE\\_LEN](#), 206
  - [MAC\\_SEQNUM\\_LEN](#), 206
  - [MD5LEN](#), 207
  - [SASL\\_INTEGRITY\\_PREFIX\\_LENGTH](#), 207
- [session.h](#), 208
  - [digest\\_md5\\_decode](#), 208
  - [digest\\_md5\\_encode](#), 208
- set\_saltedpassword
  - [tools.c](#), 217
  - [tools.h](#), 217
- sf
  - [scram\\_client\\_state](#), 50
  - [scram\\_server\\_state](#), 55
- sf\_str
  - [scram\\_server\\_state](#), 55
- sl
  - [scram\\_client\\_state](#), 50
  - [scram\\_server\\_state](#), 55
- snonce
  - [scram\\_server\\_state](#), 56
- SNONCE\_ENTROPY\_BYTES
  - [scram/server.c](#), 203
- [src/free.c](#)
  - [gsasl\\_free](#), 102
- stale
  - [digest\\_md5\\_challenge](#), 30
- start
  - [Gsasl\\_mechanism\\_functions](#), 38
- step
  - [\\_Gsasl\\_digest\\_md5\\_client\\_state](#), 19
  - [\\_Gsasl\\_digest\\_md5\\_server\\_state](#), 21
  - [\\_Gsasl\\_gs2\\_server\\_state](#), 24
  - [\\_Gsasl\\_gssapi\\_client\\_state](#), 25
  - [\\_Gsasl\\_gssapi\\_server\\_state](#), 26
  - [\\_Gsasl\\_login\\_client\\_state](#), 27
  - [\\_Gsasl\\_login\\_server\\_state](#), 28
  - [\\_Gsasl\\_ntlm\\_state](#), 28
  - [\\_gsasl\\_gs2\\_client\\_state](#), 23
  - [Gsasl\\_mechanism\\_functions](#), 38
  - [openid20\\_client\\_state](#), 44
  - [openid20\\_server\\_state](#), 45
  - [saml20\\_client\\_state](#), 46
  - [saml20\\_server\\_state](#), 46
  - [scram\\_client\\_state](#), 51
  - [scram\\_server\\_state](#), 56
- storedkey
  - [scram\\_server\\_state](#), 56
- [suggest.c](#), 208
  - [\\_GSASL\\_VALID\\_MECHANISM\\_CHARACTERS](#), 210
  - [gsasl\\_client\\_suggest\\_mechanism](#), 209
  - [gsasl\\_mechanism\\_name\\_p](#), 209
- suggestedpin
  - [Gsasl\\_session](#), 44
- [supportp.c](#), 210
  - [gsasl\\_client\\_support\\_p](#), 210
  - [gsasl\\_server\\_support\\_p](#), 211
- TEMPLATE
  - [challenge.c](#), 65
- [test-parser.c](#), 211
  - [main](#), 212
- token
  - [\\_gsasl\\_gs2\\_client\\_state](#), 23
- [tokens.c](#), 212
  - [scram\\_free\\_client\\_final](#), 212
  - [scram\\_free\\_client\\_first](#), 212
  - [scram\\_free\\_server\\_final](#), 212
  - [scram\\_free\\_server\\_first](#), 213
- [tokens.h](#), 213, 216
- [tools.c](#), 217
  - [set\\_saltedpassword](#), 217
- [tools.h](#), 217
  - [set\\_saltedpassword](#), 217
- username
  - [\\_Gsasl\\_login\\_server\\_state](#), 28
  - [digest\\_md5\\_response](#), 33
  - [scram\\_client\\_first](#), 48
- utf8
  - [digest\\_md5\\_challenge](#), 30
  - [digest\\_md5\\_response](#), 33
- utf8tolatin1ifpossible
  - [nonascii.c](#), 162
  - [nonascii.h](#), 163
- [validate.c](#), 218, 219
- [validate.h](#), 220, 221
- verifier
  - [scram\\_server\\_final](#), 51
- [version.c](#), 222

gsasl\_check\_version, [222](#)

x-gssapi.h, [223](#)

- [\\_gsasl\\_gssapi\\_client\\_decode](#), [224](#)
- [\\_gsasl\\_gssapi\\_client\\_encode](#), [224](#)
- [\\_gsasl\\_gssapi\\_client\\_finish](#), [224](#)
- [\\_gsasl\\_gssapi\\_client\\_start](#), [224](#)
- [\\_gsasl\\_gssapi\\_client\\_step](#), [225](#)
- [\\_gsasl\\_gssapi\\_mechanism](#), [226](#)
- [\\_gsasl\\_gssapi\\_server\\_finish](#), [225](#)
- [\\_gsasl\\_gssapi\\_server\\_start](#), [225](#)
- [\\_gsasl\\_gssapi\\_server\\_step](#), [225](#)
- GSASL\_GSSAPI\_NAME, [223](#)

x-ntlm.h, [226](#)

- [\\_gsasl\\_ntlm\\_client\\_finish](#), [226](#)
- [\\_gsasl\\_ntlm\\_client\\_start](#), [227](#)
- [\\_gsasl\\_ntlm\\_client\\_step](#), [227](#)
- [\\_gsasl\\_ntlm\\_mechanism](#), [227](#)
- GSASL\_NTLM\_NAME, [226](#)

xcode.c, [227](#)

- [gsasl\\_decode](#), [228](#)
- [gsasl\\_encode](#), [228](#)

xfinish.c, [229](#)

- [gsasl\\_finish](#), [229](#)

xstart.c, [230](#)

- [gsasl\\_client\\_start](#), [230](#)
- [gsasl\\_server\\_start](#), [230](#)

xstep.c, [231](#)

- [gsasl\\_step](#), [231](#)
- [gsasl\\_step64](#), [232](#)