

gsasl Reference Manual
0.2.15

Generated by Doxygen 1.4.7

Tue Aug 22 12:06:06 2006

Contents

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	gsasl Directory Hierarchy	13
2.1	gsasl Directories	13
3	gsasl Data Structure Index	15
3.1	gsasl Data Structures	15
4	gsasl File Index	17
4.1	gsasl File List	17
5	gsasl Directory Documentation	19
5.1	/home/jas/self/src/gsas/lib/anonymous/ Directory Reference	19
5.2	/home/jas/self/src/gsas/lib/cram-md5/ Directory Reference	20
5.3	/home/jas/self/src/gsas/lib/digest-md5/ Directory Reference	21
5.4	/home/jas/self/src/gsas/lib/external/ Directory Reference	22
5.5	/home/jas/self/src/gsas/lib/gssapi/ Directory Reference	23
5.6	/home/jas/self/src/gsas/lib/kerberos_v5/ Directory Reference	24
5.7	/home/jas/self/src/gsas/lib/ Directory Reference	25
5.8	/home/jas/self/src/gsas/lib/login/ Directory Reference	26
5.9	/home/jas/self/src/gsas/lib/ntlm/ Directory Reference	27
5.10	/home/jas/self/src/gsas/lib/plain/ Directory Reference	28
5.11	/home/jas/self/src/gsas/lib/secuid/ Directory Reference	29
5.12	/home/jas/self/src/gsas/lib/src/ Directory Reference	30
5.13	/home/jas/self/src/gsas/lib/tests/ Directory Reference	31

6	gsasl Data Structure Documentation	33
6.1	_Gsasl_digest_md5_client_state Struct Reference	33
6.2	_Gsasl_digest_md5_server_state Struct Reference	36
6.3	_Gsasl_gssapi_client_state Struct Reference	38
6.4	_Gsasl_gssapi_server_state Struct Reference	39
6.5	_Gsasl_kerberos_v5_client_state Struct Reference	40
6.6	_Gsasl_kerberos_v5_server_state Struct Reference	43
6.7	_Gsasl_login_client_state Struct Reference	47
6.8	_Gsasl_login_server_state Struct Reference	48
6.9	_Gsasl_ntlm_state Struct Reference	49
6.10	digest_md5_challenge Struct Reference	50
6.11	digest_md5_finish Struct Reference	52
6.12	digest_md5_response Struct Reference	53
6.13	Gsasl Struct Reference	56
6.14	Gsasl_mechanism Struct Reference	61
6.15	Gsasl_mechanism_functions Struct Reference	62
6.16	Gsasl_session Struct Reference	64
7	gsasl File Documentation	67
7.1	anonymous.h File Reference	67
7.2	base64.c File Reference	69
7.3	callback.c File Reference	71
7.4	challenge.c File Reference	74
7.5	challenge.h File Reference	76
7.6	client.c File Reference	77
7.7	client.c File Reference	78
7.8	client.c File Reference	79
7.9	client.c File Reference	82
7.10	client.c File Reference	83
7.11	client.c File Reference	85
7.12	client.c File Reference	88
7.13	client.c File Reference	89
7.14	client.c File Reference	90
7.15	cram-md5.h File Reference	92
7.16	crypto.c File Reference	94
7.17	digest-md5.h File Reference	96
7.18	digest.c File Reference	99

7.19	digest.h File Reference	100
7.20	digestmac.c File Reference	101
7.21	digestmac.h File Reference	105
7.22	done.c File Reference	106
7.23	doxygen.c File Reference	107
7.24	error.c File Reference	108
7.25	external.h File Reference	110
7.26	free.c File Reference	111
7.27	free.h File Reference	112
7.28	getsubopt.c File Reference	113
7.29	gsasl-compat.h File Reference	114
7.30	gsasl-mech.h File Reference	149
7.31	gsasl.h File Reference	151
7.32	init.c File Reference	171
7.33	internal.h File Reference	173
7.34	kerberos_v5.c File Reference	174
7.35	kerberos_v5.h File Reference	176
7.36	listmech.c File Reference	179
7.37	login.h File Reference	180
7.38	md5pwd.c File Reference	182
7.39	mechinfo.c File Reference	183
7.40	mechinfo.c File Reference	184
7.41	mechinfo.c File Reference	185
7.42	mechinfo.c File Reference	186
7.43	mechinfo.c File Reference	187
7.44	mechinfo.c File Reference	188
7.45	mechinfo.c File Reference	189
7.46	mechinfo.c File Reference	190
7.47	mechinfo.c File Reference	191
7.48	ntlm.c File Reference	192
7.49	obsolete.c File Reference	194
7.50	parser.c File Reference	223
7.51	parser.h File Reference	226
7.52	plain.h File Reference	227
7.53	printer.c File Reference	229
7.54	printer.h File Reference	230

7.55	property.c File Reference	231
7.56	register.c File Reference	234
7.57	saslprep.c File Reference	235
7.58	securid.h File Reference	236
7.59	server.c File Reference	238
7.60	server.c File Reference	239
7.61	server.c File Reference	240
7.62	server.c File Reference	243
7.63	server.c File Reference	244
7.64	server.c File Reference	246
7.65	server.c File Reference	248
7.66	server.c File Reference	250
7.67	server.c File Reference	251
7.68	session.c File Reference	252
7.69	session.h File Reference	254
7.70	shared.h File Reference	255
7.71	suggest.c File Reference	257
7.72	supportp.c File Reference	258
7.73	test-gc-hmac-md5.c File Reference	259
7.74	test-gc-md5.c File Reference	260
7.75	test-gc.c File Reference	261
7.76	test-parser.c File Reference	262
7.77	tokens.h File Reference	263
7.78	validate.c File Reference	265
7.79	validate.h File Reference	266
7.80	version.c File Reference	267
7.81	x-gssapi.h File Reference	268
7.82	x-ntlm.h File Reference	271
7.83	xcode.c File Reference	273
7.84	xfinish.c File Reference	275
7.85	xstart.c File Reference	276
7.86	xstep.c File Reference	277

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 ('libsasl'), 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/gsas/>

The software archive:

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

Further information and paid contract development:

Simon Josefsson <simon@josefsson.org>

1.2 Logical overview

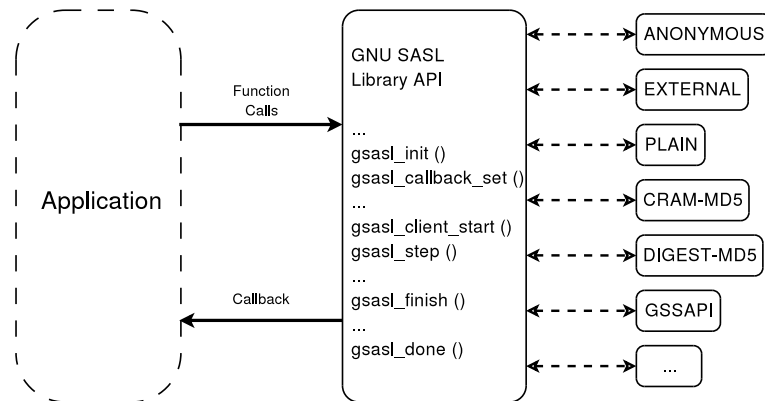


Figure 1.1: Logical overview

1.3 Control flow in application using the library

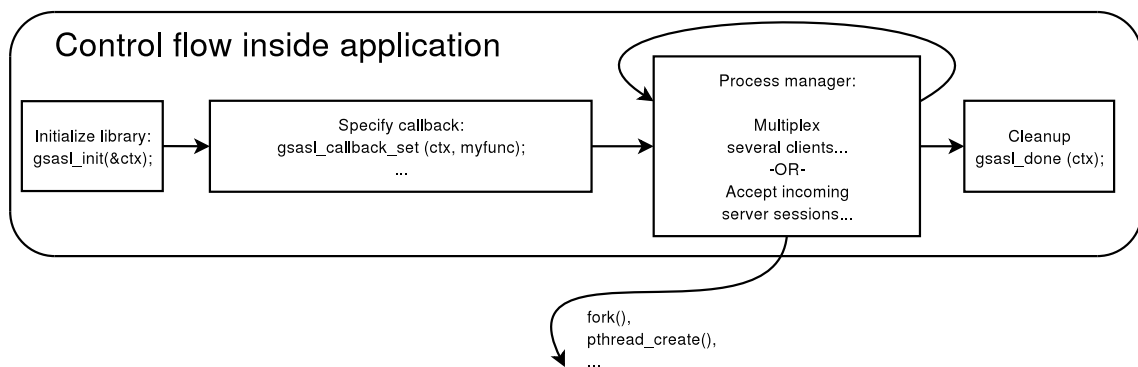


Figure 1.2: Control flow

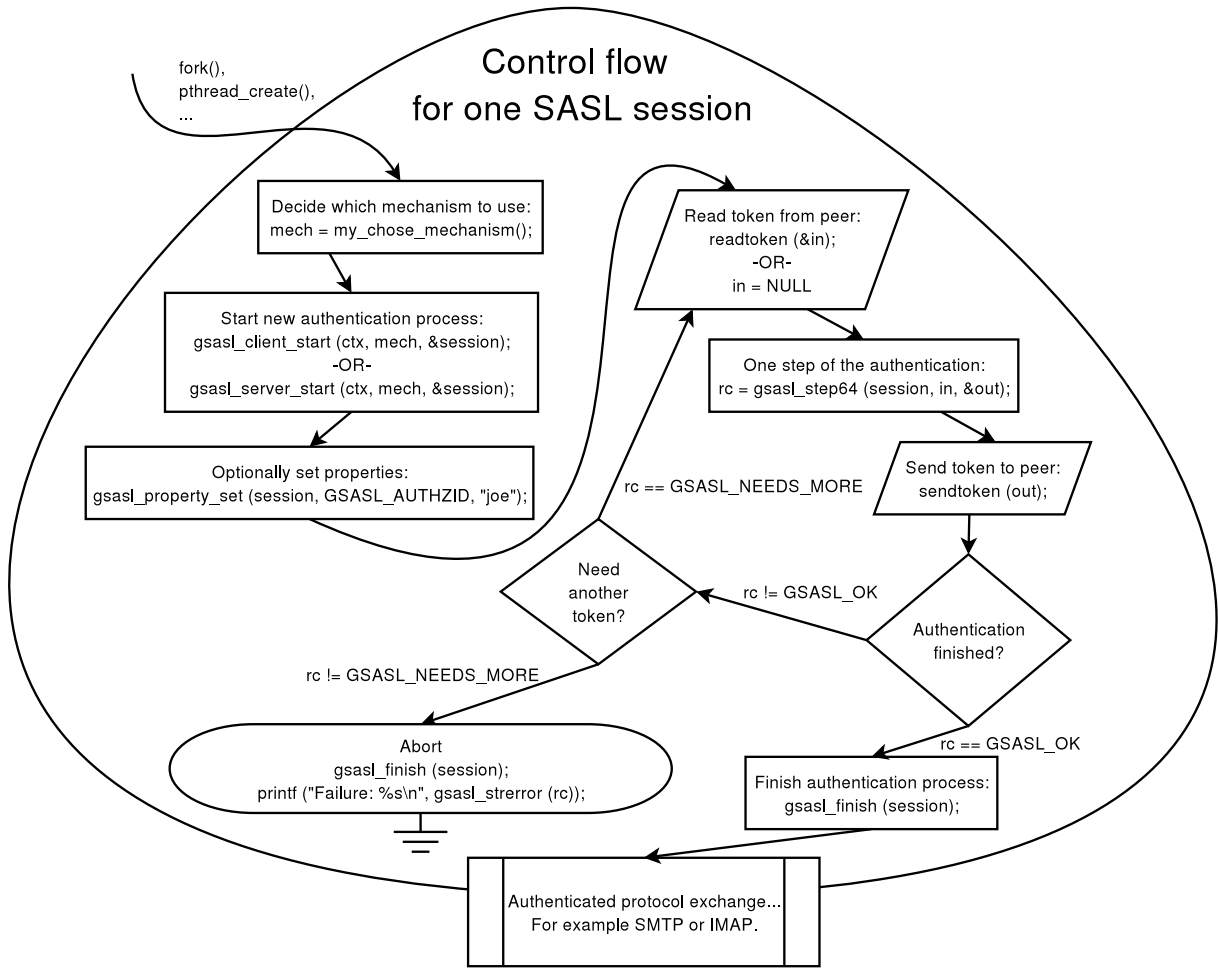


Figure 1.3: Control flow

1.4 Examples

```

/* client.c --- Example SASL client.
 * Copyright (C) 2004, 2005 Simon Josefsson
 *
 * This file is part of GNU SASL.
 *
 * GNU SASL 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 2 of the License, or
 * (at your option) any later version.
 *
 * GNU SASL 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 GNU SASL; if not, write to the Free Software
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
 */

```

```

#include <stdarg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include <gsasl.h>

static void
client_authenticate (Gsasl * ctx, Gsasl_session * session)
{
    char buf[BUFSIZ] = "";
    char *p;
    int rc;

    /* This loop mimic 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 sucessful, print it. */
            printf ("Output:\n%s\n", p);
            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");
            fgets (buf, sizeof (buf) - 1, stdin);
            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. */
gsasl_property_set (session, GSASL_AUTHID, "jas");
gsasl_property_set (session, GSASL_PASSWORD, "secret");

/* Do it. */
client_authenticate (ctx, session);

/* Cleanup. */
gsasl_finish (session);
}

int
main (int argc, char *argv[])
{
    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, 2005 Simon Josefsson
 *
 * This file is part of GNU SASL.
 *
 * GNU SASL 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 2 of the License, or
 * (at your option) any later version.
 *
 * GNU SASL 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 GNU SASL; if not, write to the Free Software
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
 */

#include <stdarg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include <gsasl.h>

static void
client_authenticate (Gsasl * ctx, Gsasl_session * session)
{
    char buf[BUFSIZ] = "";
    char *p;
```

```

int rc;

/* This loop mimic a protocol where the server get to send data first. */
do
{
    printf ("Input base64 encoded data from server:\n");
    fgets (buf, sizeof (buf) - 1, stdin);
    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);
        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. */
    gsasl_property_set (session, GSASL_AUTHID, "jas");
    gsasl_property_set (session, GSASL_PASSWORD, "secret");

    /* Do it. */
    client_authenticate (ctx, session);

    /* Cleanup. */
    gsasl_finish (session);
}

int
main (int argc, char *argv[])
{
    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, 2005 Simon Josefsson
 *
 * This file is part of GNU SASL.
 *
 * GNU SASL 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 2 of the License, or
 * (at your option) any later version.
 *
 * GNU SASL 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 GNU SASL; if not, write to the Free Software
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
 */

#include <stdarg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include <gsasl.h>

static void
client_authenticate (Gsasl * ctx, Gsasl_session * session)
{
    char buf[BUFSIZ] = "";
    char *p;
    int rc;

    /* This loop mimic a protocol where the server get to send data first. */

    do
    {
        printf ("Input base64 encoded data from server:\n");
        fgets (buf, sizeof (buf) - 1, stdin);
        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);
            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;

    printf ("Enter list of mechanism that server support, separate by SPC:\n");
    fgets (meclist, sizeof (meclist) - 1, stdin);

    suggestion = gsasl_client_suggest_mechanism (ctx, meclist);
    if (suggestion)
        printf ("Library suggest use of '%s'.\n", suggestion);

    printf ("Enter mechanism to use:\n");
    fgets (mech, sizeof (mech) - 1, stdin);
    mech[strlen (mech) - 1] = '\0';

    return mech;
}

static void
client (Gsasl * ctx)
{
    Gsasl_session *session;
    const char *mech;
    int rc;

    /* Find out which mechanism to use. */
    mech = client_mechanism (ctx);

    /* 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. */
    gsasl_property_set (session, GSASL_AUTHID, "jas");
    gsasl_property_set (session, GSASL_PASSWORD, "secret");

    /* Do it. */
    client_authenticate (ctx, session);

    /* Cleanup. */
    gsasl_finish (session);
}

```

```

int
main (int argc, char *argv[])
{
    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-callback.c --- Example SASL client, with callback for user info.
 * Copyright (C) 2004, 2005 Simon Josefsson
 *
 * This file is part of GNU SASL.
 *
 * GNU SASL 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 2 of the License, or
 * (at your option) any later version.
 *
 * GNU SASL 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 GNU SASL; if not, write to the Free Software
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
 */

#include <stdarg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include <gsasl.h>

static void
client_authenticate (Gsasl * ctx, Gsasl_session * session)
{
    char buf[BUFSIZ] = "";
    char *p;
    int rc;

    /* This loop mimic a protocol where the server get to send data first. */

    do
    {
        printf ("Input base64 encoded data from server:\n");
        fgets (buf, sizeof (buf) - 1, stdin);
        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);
        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 (ctx, session);

    /* Cleanup. */
    gsasl_finish (session);
}

static int
callback (Gsasl * ctx, Gsasl_session * sctx, Gsasl_property prop)
{
    char buf[BUFSIZ] = "";
    int rc = GSASL_NO_CALLBACK;

    /* Get user info from user. */

    printf ("Callback invoked, for property %d.\n", prop);

    switch (prop)
    {
        case GSASL_PASSCODE:
            printf ("Enter passcode:\n");
            fgets (buf, sizeof (buf) - 1, stdin);
            buf[strlen (buf) - 1] = '\0';

            gsasl_property_set (sctx, GSASL_PASSCODE, buf);
            rc = GSASL_OK;
            break;

        case GSASL_AUTHID:
            printf ("Enter username:\n");

```



```
fgets (buf, sizeof (buf) - 1, stdin);
buf[strlen (buf) - 1] = '\0';

gsasl_property_set (sctx, GSASL_AUTHID, buf);
rc = GSASL_OK;
break;

default:
    printf ("Unknown property! Don't worry.\n");
    break;
}

return rc;
}

int
main (int argc, char *argv[])
{
    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;
    }

    /* Set the callback handler for the library. */
    gsasl_callback_set (ctx, callback);

    /* Do it. */
    client (ctx);

    /* Cleanup. */
    gsasl_done (ctx);

    return 0;
}
```


Chapter 2

gsasl Directory Hierarchy

2.1 gsasl Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

lib	25
anonymous	19
cram-md5	20
digest-md5	21
external	22
gssapi	23
kerberos_v5	24
login	26
ntlm	27
plain	28
securid	29
src	30
tests	31

Chapter 3

gsasl Data Structure Index

3.1 gsasl Data Structures

Here are the data structures with brief descriptions:

_Gsasl_digest_md5_client_state	33
_Gsasl_digest_md5_server_state	36
_Gsasl_gssapi_client_state	38
_Gsasl_gssapi_server_state	39
_Gsasl_kerberos_v5_client_state	40
_Gsasl_kerberos_v5_server_state	43
_Gsasl_login_client_state	47
_Gsasl_login_server_state	48
_Gsasl_ntlm_state	49
digest_md5_challenge	50
digest_md5_finish	52
digest_md5_response	53
Gsasl	56
Gsasl_mechanism	61
Gsasl_mechanism_functions	62
Gsasl_session	64

Chapter 4

gsasl File Index

4.1 gsasl File List

Here is a list of all files with brief descriptions:

anonymous.h	67
base64.c	69
callback.c	71
challenge.c	74
challenge.h	76
anonymous/client.c	77
cram-md5/client.c	78
digest-md5/client.c	79
external/client.c	82
gssapi/client.c	83
kerberos_v5/client.c	85
login/client.c	88
plain/client.c	89
securid/client.c	90
cram-md5.h	92
crypto.c	94
digest-md5.h	96
digest.c	99
digest.h	100
digesthmac.c	101
digesthmac.h	105
done.c	106
doxygen.c	107
error.c	108
external.h	110
free.c	111
free.h	112
getsubopt.c	113
gsasl-compat.h	114
gsasl-mech.h	149
gsasl.h	151
init.c	171
internal.h	173

kerberos_v5.c	174
kerberos_v5.h	176
listmech.c	179
login.h	180
md5pwd.c	182
anonymous/mechinfo.c	183
cram-md5/mechinfo.c	184
digest-md5/mechinfo.c	185
external/mechinfo.c	186
gssapi/mechinfo.c	187
login/mechinfo.c	188
ntlm/mechinfo.c	189
plain/mechinfo.c	190
securid/mechinfo.c	191
ntlm.c	192
obsolete.c	194
parser.c	223
parser.h	226
plain.h	227
printer.c	229
printer.h	230
property.c	231
register.c	234
saslprep.c	235
securid.h	236
anonymous/server.c	238
cram-md5/server.c	239
digest-md5/server.c	240
external/server.c	243
gssapi/server.c	244
kerberos_v5/server.c	246
login/server.c	248
plain/server.c	250
securid/server.c	251
session.c	252
session.h	254
shared.h	255
suggest.c	257
supportp.c	258
test-gc-hmac-md5.c	259
test-gc-md5.c	260
test-gc.c	261
test-parser.c	262
tokens.h	263
validate.c	265
validate.h	266
version.c	267
x-gssapi.h	268
x-ntlm.h	271
xcode.c	273
xfinish.c	275
xstart.c	276
xstep.c	277

Chapter 5

gsasl Directory Documentation

5.1 /home/jas/self/src/gsas/lib/anonymous/ Directory Reference

Files

- file [anonymous.h](#)
- file [anonymous/client.c](#)
- file [anonymous/mechinfo.c](#)
- file [anonymous/server.c](#)

5.2 /home/jas/self/src/gsaslib/cram-md5/ Directory Reference

Files

- file [challenge.c](#)
- file [challenge.h](#)
- file [cram-md5/client.c](#)
- file [cram-md5.h](#)
- file [digest.c](#)
- file [digest.h](#)
- file [cram-md5/mechinfo.c](#)
- file [cram-md5/server.c](#)

5.3 /home/jas/self/src/gsas/lib/digest-md5/ Directory Reference

Files

- file [digest-md5/client.c](#)
- file [digest-md5.h](#)
- file [digesthmac.c](#)
- file [digesthmac.h](#)
- file [free.c](#)
- file [free.h](#)
- file [getsubopt.c](#)
- file [digest-md5/mechinfo.c](#)
- file [parser.c](#)
- file [parser.h](#)
- file [printer.c](#)
- file [printer.h](#)
- file [digest-md5/server.c](#)
- file [session.c](#)
- file [session.h](#)
- file [test-parser.c](#)
- file [tokens.h](#)
- file [validate.c](#)
- file [validate.h](#)

5.4 /home/jas/self/src/gsas/lib/external/ Directory Reference

Files

- file [external/client.c](#)
- file [external.h](#)
- file [external/mechinfo.c](#)
- file [external/server.c](#)

5.5 /home/jas/self/src/gsas/lib/gssapi/ Directory Reference

Files

- file [gssapi/client.c](#)
- file [gssapi/mechinfo.c](#)
- file [gssapi/server.c](#)
- file [x-gssapi.h](#)

5.6 /home/jas/self/src/gsaslib/kerberos_v5/ Directory Reference

Files

- file [kerberos_v5/client.c](#)
- file [kerberos_v5.c](#)
- file [kerberos_v5.h](#)
- file [kerberos_v5/server.c](#)
- file [shared.h](#)

5.7 /home/jas/self/src/gsas/lib/ Directory Reference

Directories

- directory [anonymous](#)
- directory [cram-md5](#)
- directory [digest-md5](#)
- directory [external](#)
- directory [gssapi](#)
- directory [kerberos_v5](#)
- directory [login](#)
- directory [ntlm](#)
- directory [plain](#)
- directory [securid](#)
- directory [src](#)
- directory [tests](#)

5.8 /home/jas/self/src/gsas/lib/login/ Directory Reference

Files

- file [login/client.c](#)
- file [login.h](#)
- file [login/mechinfo.c](#)
- file [login/server.c](#)

5.9 /home/jas/self/src/gsas/lib/ntlm/ Directory Reference

Files

- file [ntlm/mechinfo.c](#)
- file [ntlm.c](#)
- file [x-ntlm.h](#)

5.10 /home/jas/self/src/gsaslib/plain/ Directory Reference

Files

- file [plain/client.c](#)
- file [plain/mechinfo.c](#)
- file [plain.h](#)
- file [plain/server.c](#)

5.11 /home/jas/self/src/gsas/lib/securid/ Directory Reference

Files

- file [securid/client.c](#)
- file [securid/mechinfo.c](#)
- file [securid.h](#)
- file [securid/server.c](#)

5.12 /home/jas/self/src/gsaslib/src/ Directory Reference

Files

- file [base64.c](#)
- file [callback.c](#)
- file [crypto.c](#)
- file [done.c](#)
- file [doxygen.c](#)
- file [error.c](#)
- file [gsasl-compat.h](#)
- file [gsasl-mech.h](#)
- file [gsasl.h](#)
- file [init.c](#)
- file [internal.h](#)
- file [listmech.c](#)
- file [md5pwd.c](#)
- file [obsolete.c](#)
- file [property.c](#)
- file [register.c](#)
- file [saslprep.c](#)
- file [suggest.c](#)
- file [supportp.c](#)
- file [version.c](#)
- file [xcode.c](#)
- file [xfinish.c](#)
- file [xstart.c](#)
- file [xstep.c](#)

5.13 /home/jas/self/src/gsas/lib/tests/ Directory Reference

Files

- file [test-gc-hmac-md5.c](#)
- file [test-gc-md5.c](#)
- file [test-gc.c](#)

Chapter 6

gsasl Data Structure Documentation

6.1 `_Gsasl_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

6.1.1 Detailed Description

Definition at line 46 of file `digest-md5/client.c`.

6.1.2 Field Documentation

6.1.2.1 `digest_md5_challenge` `_Gsasl_digest_md5_client_state::challenge`

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

Referenced by `_gsasl_digest_md5_client_finish()`, and `_gsasl_digest_md5_client_step()`.

6.1.2.2 `digest_md5_finish` `_Gsasl_digest_md5_client_state::finish`

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

Referenced by `_gsasl_digest_md5_client_finish()`, and `_gsasl_digest_md5_client_step()`.

6.1.2.3 char [_Gsasl_digest_md5_client_state::kcc](#)[DIGEST_MD5_LENGTH]

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

Referenced by `_gsasl_digest_md5_client_step()`.

6.1.2.4 char [_Gsasl_digest_md5_client_state::kcs](#)[DIGEST_MD5_LENGTH]

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

Referenced by `_gsasl_digest_md5_client_step()`.

6.1.2.5 char [_Gsasl_digest_md5_client_state::kic](#)[DIGEST_MD5_LENGTH]

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

Referenced by `_gsasl_digest_md5_client_encode()`, and `_gsasl_digest_md5_client_step()`.

6.1.2.6 char [_Gsasl_digest_md5_client_state::kis](#)[DIGEST_MD5_LENGTH]

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

Referenced by `_gsasl_digest_md5_client_decode()`, and `_gsasl_digest_md5_client_step()`.

6.1.2.7 unsigned long [_Gsasl_digest_md5_client_state::readseqnum](#)

Definition at line 49 of file digest-md5/client.c.

Referenced by `_gsasl_digest_md5_client_decode()`.

6.1.2.8 [digest_md5_response](#) [_Gsasl_digest_md5_client_state::response](#)

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

Referenced by `_gsasl_digest_md5_client_decode()`, `_gsasl_digest_md5_client_encode()`, `_gsasl_digest_md5_client_finish()`, and `_gsasl_digest_md5_client_step()`.

6.1.2.9 char [_Gsasl_digest_md5_client_state::secret](#)[DIGEST_MD5_LENGTH]

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

Referenced by `_gsasl_digest_md5_client_step()`.

6.1.2.10 unsigned long [_Gsasl_digest_md5_client_state::sendseqnum](#)

Definition at line 49 of file digest-md5/client.c.

Referenced by `_gsasl_digest_md5_client_encode()`.

6.1.2.11 int [_Gsasl_digest_md5_client_state::step](#)

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

Referenced by `_gsasl_digest_md5_client_step()`.

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

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

6.2 `_Gsasl_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

6.2.1 Detailed Description

Definition at line 47 of file `digest-md5/server.c`.

6.2.2 Field Documentation

6.2.2.1 `digest_md5_challenge` `_Gsasl_digest_md5_server_state::challenge`

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

Referenced by `_gsasl_digest_md5_server_finish()`, and `_gsasl_digest_md5_server_step()`.

6.2.2.2 `digest_md5_finish` `_Gsasl_digest_md5_server_state::finish`

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

Referenced by `_gsasl_digest_md5_server_finish()`, and `_gsasl_digest_md5_server_step()`.

6.2.2.3 `char` `_Gsasl_digest_md5_server_state::kcc`[DIGEST_MD5_LENGTH]

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

6.2.2.4 `char` `_Gsasl_digest_md5_server_state::kcs`[DIGEST_MD5_LENGTH]

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

6.2.2.5 `char` `_Gsasl_digest_md5_server_state::kic`[DIGEST_MD5_LENGTH]

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

Referenced by `_gsasl_digest_md5_server_decode()`.

6.2.2.6 `char _Gsassl_digest_md5_server_state::kis[DIGEST_MD5_LENGTH]`

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

Referenced by `_gsasl_digest_md5_server_encode()`.

6.2.2.7 `unsigned long _Gsassl_digest_md5_server_state::readseqnum`

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

Referenced by `_gsasl_digest_md5_server_decode()`.

6.2.2.8 `digest_md5_response _Gsassl_digest_md5_server_state::response`

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

Referenced by `_gsasl_digest_md5_server_decode()`, `_gsasl_digest_md5_server_encode()`, `_gsasl_digest_md5_server_finish()`, and `_gsasl_digest_md5_server_step()`.

6.2.2.9 `char _Gsassl_digest_md5_server_state::secret[DIGEST_MD5_LENGTH]`

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

Referenced by `_gsasl_digest_md5_server_step()`.

6.2.2.10 `unsigned long _Gsassl_digest_md5_server_state::sendseqnum`

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

Referenced by `_gsasl_digest_md5_server_encode()`.

6.2.2.11 `int _Gsassl_digest_md5_server_state::step`

Definition at line 49 of file `digest-md5/server.c`.

Referenced by `_gsasl_digest_md5_server_step()`.

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

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

6.3 `_Gsasl_gssapi_client_state` Struct Reference

Data Fields

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

6.3.1 Detailed Description

Definition at line 49 of file `gssapi/client.c`.

6.3.2 Field Documentation

6.3.2.1 `gss_ctx_id_t` [_Gsasl_gssapi_client_state::context](#)

Definition at line 53 of file `gssapi/client.c`.

Referenced by `_gsasl_gssapi_client_decode()`, `_gsasl_gssapi_client_encode()`, `_gsasl_gssapi_client_finish()`, and `_gsasl_gssapi_client_step()`.

6.3.2.2 `gss_qop_t` [_Gsasl_gssapi_client_state::qop](#)

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

Referenced by `_gsasl_gssapi_client_decode()`, `_gsasl_gssapi_client_encode()`, and `_gsasl_gssapi_client_step()`.

6.3.2.3 `gss_name_t` [_Gsasl_gssapi_client_state::service](#)

Definition at line 52 of file `gssapi/client.c`.

Referenced by `_gsasl_gssapi_client_finish()`, and `_gsasl_gssapi_client_step()`.

6.3.2.4 `int` [_Gsasl_gssapi_client_state::step](#)

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

Referenced by `_gsasl_gssapi_client_decode()`, `_gsasl_gssapi_client_encode()`, and `_gsasl_gssapi_client_step()`.

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

- [gssapi/client.c](#)

6.4 `_Gssapi_server_state` Struct Reference

Data Fields

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

6.4.1 Detailed Description

Definition at line 49 of file `gssapi/server.c`.

6.4.2 Field Documentation

6.4.2.1 `gss_name_t` [_Gssapi_server_state::client](#)

Definition at line 52 of file `gssapi/server.c`.

Referenced by `_gsasl_gssapi_server_finish()`, and `_gsasl_gssapi_server_step()`.

6.4.2.2 `gss_ctx_id_t` [_Gssapi_server_state::context](#)

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

Referenced by `_gsasl_gssapi_server_finish()`, and `_gsasl_gssapi_server_step()`.

6.4.2.3 `gss_cred_id_t` [_Gssapi_server_state::cred](#)

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

Referenced by `_gsasl_gssapi_server_finish()`, and `_gsasl_gssapi_server_step()`.

6.4.2.4 `int` [_Gssapi_server_state::step](#)

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

Referenced by `_gsasl_gssapi_server_step()`.

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

- [gssapi/server.c](#)

6.5 `_Gsasl_kerberos_v5_client_state` Struct Reference

Data Fields

- int `step`
- char `serverhello` [BITMAP_LEN+MAXBUF_LEN+RANDOM_LEN]
- int `serverqops`
- int `clientqop`
- int `servermutual`
- uint32_t `servermaxbuf`
- uint32_t `clientmaxbuf`
- Shishi * `sh`
- Shishi_tkt * `tk`
- Shishi_as * `as`
- Shishi_ap * `ap`
- Shishi_key * `sessionkey`
- Shishi_safe * `safe`

6.5.1 Detailed Description

Definition at line 30 of file `kerberos_v5/client.c`.

6.5.2 Field Documentation

6.5.2.1 `Shishi_ap* _Gsasl_kerberos_v5_client_state::ap`

Definition at line 42 of file `kerberos_v5/client.c`.

Referenced by `_gsasl_kerberos_v5_client_step()`.

6.5.2.2 `Shishi_as* _Gsasl_kerberos_v5_client_state::as`

Definition at line 41 of file `kerberos_v5/client.c`.

Referenced by `_gsasl_kerberos_v5_client_step()`.

6.5.2.3 `uint32_t _Gsasl_kerberos_v5_client_state::clientmaxbuf`

Definition at line 38 of file `kerberos_v5/client.c`.

Referenced by `_gsasl_kerberos_v5_client_step()`.

6.5.2.4 `int _Gsasl_kerberos_v5_client_state::clientqop`

Definition at line 35 of file `kerberos_v5/client.c`.

Referenced by `_gsasl_kerberos_v5_client_decode()`, `_gsasl_kerberos_v5_client_encode()`, and `_gsasl_kerberos_v5_client_step()`.

6.5.2.5 `Shishi_safe* _Gsassl_kerberos_v5_client_state::safe`

Definition at line 44 of file `kerberos_v5/client.c`.

Referenced by `_gsasl_kerberos_v5_client_encode()`.

6.5.2.6 `char _Gsassl_kerberos_v5_client_state::serverhello[BITMAP_LEN+MAXBUF_LEN+RANDOM_LEN]`

Definition at line 33 of file `kerberos_v5/client.c`.

Referenced by `_gsasl_kerberos_v5_client_step()`.

6.5.2.7 `uint32_t _Gsassl_kerberos_v5_client_state::servermaxbuf`

Definition at line 37 of file `kerberos_v5/client.c`.

Referenced by `_gsasl_kerberos_v5_client_step()`.

6.5.2.8 `int _Gsassl_kerberos_v5_client_state::servermutual`

Definition at line 36 of file `kerberos_v5/client.c`.

Referenced by `_gsasl_kerberos_v5_client_step()`.

6.5.2.9 `int _Gsassl_kerberos_v5_client_state::serverqops`

Definition at line 34 of file `kerberos_v5/client.c`.

Referenced by `_gsasl_kerberos_v5_client_step()`.

6.5.2.10 `Shishi_key* _Gsassl_kerberos_v5_client_state::sessionkey`

Definition at line 43 of file `kerberos_v5/client.c`.

Referenced by `_gsasl_kerberos_v5_client_decode()`, `_gsasl_kerberos_v5_client_encode()`, and `_gsasl_kerberos_v5_client_step()`.

6.5.2.11 `Shishi* _Gsassl_kerberos_v5_client_state::sh`

Definition at line 39 of file `kerberos_v5/client.c`.

Referenced by `_gsasl_kerberos_v5_client_encode()`, `_gsasl_kerberos_v5_client_finish()`, and `_gsasl_kerberos_v5_client_step()`.

6.5.2.12 `int _Gsassl_kerberos_v5_client_state::step`

Definition at line 32 of file `kerberos_v5/client.c`.

Referenced by `_gsasl_kerberos_v5_client_step()`.

6.5.2.13 Shishi_tkt* [_Gsasl_kerberos_v5_client_state::tkt](#)

Definition at line 40 of file kerberos_v5/client.c.

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

- [kerberos_v5/client.c](#)

6.6 _Gssapi_kerberos_v5_server_state Struct Reference

Data Fields

- int [firststep](#)
- Shishi * [sh](#)
- char [serverhello](#) [BITMAP_LEN+MAXBUF_LEN+RANDOM_LEN]
- char * [random](#)
- int [serverqops](#)
- uint32_t [servermaxbuf](#)
- int [clientqop](#)
- int [clientmutual](#)
- uint32_t [clientmaxbuf](#)
- char * [username](#)
- char * [userrealm](#)
- char * [serverrealm](#)
- char * [serverservice](#)
- char * [serverhostname](#)
- char * [password](#)
- Shishi_key * [userkey](#)
- Shishi_key * [sessionkey](#)
- Shishi_key * [sessiontktkey](#)
- Shishi_ap * [ap](#)
- Shishi_as * [as](#)
- Shishi_safe * [safe](#)

6.6.1 Detailed Description

Definition at line 30 of file `kerberos_v5/server.c`.

6.6.2 Field Documentation

6.6.2.1 Shishi_ap* [_Gssapi_kerberos_v5_server_state::ap](#)

Definition at line 50 of file `kerberos_v5/server.c`.

Referenced by `_gssapi_kerberos_v5_server_step()`.

6.6.2.2 Shishi_as* [_Gssapi_kerberos_v5_server_state::as](#)

Definition at line 51 of file `kerberos_v5/server.c`.

Referenced by `_gssapi_kerberos_v5_server_step()`.

6.6.2.3 uint32_t [_Gssapi_kerberos_v5_server_state::clientmaxbuf](#)

Definition at line 40 of file `kerberos_v5/server.c`.

Referenced by `_gssapi_kerberos_v5_server_step()`.

6.6.2.4 int [_Gsasl_kerberos_v5_server_state::clientmutual](#)

Definition at line 39 of file kerberos_v5/server.c.

Referenced by [_gsasl_kerberos_v5_server_step\(\)](#).

6.6.2.5 int [_Gsasl_kerberos_v5_server_state::clientqop](#)

Definition at line 38 of file kerberos_v5/server.c.

Referenced by [_gsasl_kerberos_v5_server_decode\(\)](#), [_gsasl_kerberos_v5_server_encode\(\)](#), and [_gsasl_kerberos_v5_server_step\(\)](#).

6.6.2.6 int [_Gsasl_kerberos_v5_server_state::firststep](#)

Definition at line 32 of file kerberos_v5/server.c.

Referenced by [_gsasl_kerberos_v5_server_step\(\)](#).

6.6.2.7 char* [_Gsasl_kerberos_v5_server_state::password](#)

Definition at line 46 of file kerberos_v5/server.c.

Referenced by [_gsasl_kerberos_v5_server_finish\(\)](#), and [_gsasl_kerberos_v5_server_step\(\)](#).

6.6.2.8 char* [_Gsasl_kerberos_v5_server_state::random](#)

Definition at line 35 of file kerberos_v5/server.c.

Referenced by [_gsasl_kerberos_v5_server_finish\(\)](#), and [_gsasl_kerberos_v5_server_step\(\)](#).

6.6.2.9 Shishi_safe* [_Gsasl_kerberos_v5_server_state::safe](#)

Definition at line 52 of file kerberos_v5/server.c.

Referenced by [_gsasl_kerberos_v5_server_decode\(\)](#), and [_gsasl_kerberos_v5_server_encode\(\)](#).

6.6.2.10 char [_Gsasl_kerberos_v5_server_state::serverhello](#)[BITMAP_LEN+MAXBUF_LEN+RANDOM_LEN]

Definition at line 34 of file kerberos_v5/server.c.

Referenced by [_gsasl_kerberos_v5_server_step\(\)](#).

6.6.2.11 char* [_Gsasl_kerberos_v5_server_state::serverhostname](#)

Definition at line 45 of file kerberos_v5/server.c.

Referenced by [_gsasl_kerberos_v5_server_step\(\)](#).

6.6.2.12 `uint32_t _Gssasl_kerberos_v5_server_state::servermaxbuf`

Definition at line 37 of file `kerberos_v5/server.c`.

Referenced by `_gssasl_kerberos_v5_server_step()`.

6.6.2.13 `int _Gssasl_kerberos_v5_server_state::serverqops`

Definition at line 36 of file `kerberos_v5/server.c`.

Referenced by `_gssasl_kerberos_v5_server_step()`.

6.6.2.14 `char* _Gssasl_kerberos_v5_server_state::serverrealm`

Definition at line 43 of file `kerberos_v5/server.c`.

Referenced by `_gssasl_kerberos_v5_server_step()`.

6.6.2.15 `char* _Gssasl_kerberos_v5_server_state::serverservice`

Definition at line 44 of file `kerberos_v5/server.c`.

Referenced by `_gssasl_kerberos_v5_server_step()`.

6.6.2.16 `Shishi_key* _Gssasl_kerberos_v5_server_state::sessionkey`

Definition at line 48 of file `kerberos_v5/server.c`.

Referenced by `_gssasl_kerberos_v5_server_decode()`, `_gssasl_kerberos_v5_server_encode()`, and `_gssasl_kerberos_v5_server_step()`.

6.6.2.17 `Shishi_key* _Gssasl_kerberos_v5_server_state::sessiontktkey`

Definition at line 49 of file `kerberos_v5/server.c`.

Referenced by `_gssasl_kerberos_v5_server_step()`.

6.6.2.18 `Shishi* _Gssasl_kerberos_v5_server_state::sh`

Definition at line 33 of file `kerberos_v5/server.c`.

Referenced by `_gssasl_kerberos_v5_server_decode()`, `_gssasl_kerberos_v5_server_encode()`, `_gssasl_kerberos_v5_server_finish()`, and `_gssasl_kerberos_v5_server_step()`.

6.6.2.19 `Shishi_key* _Gssasl_kerberos_v5_server_state::userkey`

Definition at line 47 of file `kerberos_v5/server.c`.

Referenced by `_gssasl_kerberos_v5_server_step()`.

6.6.2.20 char* [_Gsasl_kerberos_v5_server_state::username](#)

Definition at line 41 of file [kerberos_v5/server.c](#).

Referenced by [_gsasl_kerberos_v5_server_finish\(\)](#), and [_gsasl_kerberos_v5_server_step\(\)](#).

6.6.2.21 char* [_Gsasl_kerberos_v5_server_state::userrealm](#)

Definition at line 42 of file [kerberos_v5/server.c](#).

Referenced by [_gsasl_kerberos_v5_server_step\(\)](#).

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

- [kerberos_v5/server.c](#)

6.7 _Gsasl_login_client_state Struct Reference

Data Fields

- [int step](#)

6.7.1 Detailed Description

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

6.7.2 Field Documentation

6.7.2.1 [int _Gsasl_login_client_state::step](#)

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

Referenced by [_gsasl_login_client_step\(\)](#), [_gsasl_securid_client_finish\(\)](#), [_gsasl_securid_client_start\(\)](#), and [_gsasl_securid_client_step\(\)](#).

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

- [login/client.c](#)

6.8 `_Gsasl_login_server_state` Struct Reference

Data Fields

- `int step`
- `char * username`
- `char * password`

6.8.1 Detailed Description

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

6.8.2 Field Documentation

6.8.2.1 `char* _Gsasl_login_server_state::password`

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

Referenced by `_gsasl_login_server_finish()`, and `_gsasl_login_server_step()`.

6.8.2.2 `int _Gsasl_login_server_state::step`

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

Referenced by `_gsasl_login_server_step()`.

6.8.2.3 `char* _Gsasl_login_server_state::username`

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

Referenced by `_gsasl_login_server_finish()`, and `_gsasl_login_server_step()`.

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

- `login/server.c`

6.9 _Gssasl_ntlm_state Struct Reference

Data Fields

- int [step](#)

6.9.1 Detailed Description

Definition at line 38 of file ntlm.c.

6.9.2 Field Documentation

6.9.2.1 int [_Gssasl_ntlm_state::step](#)

Definition at line 40 of file ntlm.c.

Referenced by [_gssasl_ntlm_client_step\(\)](#).

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

- [ntlm.c](#)

6.10 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`

6.10.1 Detailed Description

Definition at line 82 of file tokens.h.

6.10.2 Field Documentation

6.10.2.1 `int digest_md5_challenge::ciphers`

Definition at line 91 of file tokens.h.

Referenced by `digest_md5_validate()`, `digest_md5_validate_challenge()`, and `main()`.

6.10.2.2 `char* digest_md5_challenge::nonce`

Definition at line 86 of file tokens.h.

Referenced by `_gsasl_digest_md5_client_step()`, `digest_md5_free_challenge()`, `digest_md5_validate()`, `digest_md5_validate_challenge()`, and `main()`.

6.10.2.3 `size_t digest_md5_challenge::nrealms`

Definition at line 84 of file tokens.h.

Referenced by `_gsasl_digest_md5_client_step()`, `_gsasl_digest_md5_server_step()`, `digest_md5_free_challenge()`, and `main()`.

6.10.2.4 `int digest_md5_challenge::qops`

Definition at line 87 of file tokens.h.

Referenced by `digest_md5_validate()`, `digest_md5_validate_challenge()`, and `main()`.

6.10.2.5 char [digest_md5_challenge::realms](#)**

Definition at line 85 of file tokens.h.

Referenced by [_gsasl_digest_md5_client_step\(\)](#), [_gsasl_digest_md5_server_step\(\)](#), [digest_md5_free_challenge\(\)](#), and [main\(\)](#).

6.10.2.6 unsigned long [digest_md5_challenge::servermaxbuf](#)

Definition at line 89 of file tokens.h.

6.10.2.7 int [digest_md5_challenge::stale](#)

Definition at line 88 of file tokens.h.

6.10.2.8 int [digest_md5_challenge::utf8](#)

Definition at line 90 of file tokens.h.

Referenced by [digest_md5_validate\(\)](#).

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

- [tokens.h](#)

6.11 digest_md5_finish Struct Reference

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

Data Fields

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

6.11.1 Detailed Description

Definition at line 146 of file tokens.h.

6.11.2 Field Documentation

6.11.2.1 char [digest_md5_finish::rspauth](#)[DIGEST_MD5_RESPONSE_LENGTH+1]

Definition at line 148 of file tokens.h.

Referenced by `_gsasl_digest_md5_client_step()`, `_gsasl_digest_md5_server_step()`, `digest_md5_validate_finish()`, and `main()`.

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

- [tokens.h](#)

6.12 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]

6.12.1 Detailed Description

Definition at line 126 of file tokens.h.

6.12.2 Field Documentation

6.12.2.1 char* [digest_md5_response::authzid](#)

Definition at line 138 of file tokens.h.

Referenced by [_gsasl_digest_md5_client_step\(\)](#), [_gsasl_digest_md5_server_step\(\)](#), and [digest_md5_free_response\(\)](#).

6.12.2.2 [digest_md5_cipher](#) [digest_md5_response::cipher](#)

Definition at line 137 of file tokens.h.

Referenced by [_gsasl_digest_md5_client_step\(\)](#), [_gsasl_digest_md5_server_step\(\)](#), [digest_md5_validate\(\)](#), and [digest_md5_validate_response\(\)](#).

6.12.2.3 unsigned long [digest_md5_response::clientmaxbuf](#)

Definition at line 135 of file tokens.h.

6.12.2.4 char* [digest_md5_response::cnonce](#)

Definition at line 131 of file tokens.h.

Referenced by [_gsasl_digest_md5_client_step\(\)](#), [_gsasl_digest_md5_server_step\(\)](#), [digest_md5_free_response\(\)](#), [digest_md5_validate_response\(\)](#), and [main\(\)](#).

6.12.2.5 char* `digest_md5_response::digesturi`

Definition at line 134 of file tokens.h.

Referenced by `_gsasl_digest_md5_client_step()`, `_gsasl_digest_md5_server_step()`, `digest_md5_free_response()`, `digest_md5_validate_response()`, and `main()`.

6.12.2.6 unsigned long `digest_md5_response::nc`

Definition at line 132 of file tokens.h.

Referenced by `_gsasl_digest_md5_client_step()`, `_gsasl_digest_md5_server_step()`, `digest_md5_validate()`, `digest_md5_validate_response()`, and `main()`.

6.12.2.7 char* `digest_md5_response::nonce`

Definition at line 130 of file tokens.h.

Referenced by `_gsasl_digest_md5_client_step()`, `_gsasl_digest_md5_server_step()`, `digest_md5_free_response()`, `digest_md5_validate()`, `digest_md5_validate_response()`, and `main()`.

6.12.2.8 `digest_md5_qop` `digest_md5_response::qop`

Definition at line 133 of file tokens.h.

Referenced by `_gsasl_digest_md5_client_decode()`, `_gsasl_digest_md5_client_encode()`, `_gsasl_digest_md5_client_step()`, `_gsasl_digest_md5_server_decode()`, `_gsasl_digest_md5_server_encode()`, `_gsasl_digest_md5_server_step()`, `digest_md5_validate()`, and `digest_md5_validate_response()`.

6.12.2.9 char* `digest_md5_response::realm`

Definition at line 129 of file tokens.h.

Referenced by `_gsasl_digest_md5_client_step()`, `_gsasl_digest_md5_server_step()`, and `digest_md5_free_response()`.

6.12.2.10 char `digest_md5_response::response`[DIGEST_MD5_RESPONSE_LENGTH+1]

Definition at line 139 of file tokens.h.

Referenced by `_gsasl_digest_md5_client_step()`, `_gsasl_digest_md5_server_step()`, `digest_md5_validate_response()`, and `main()`.

6.12.2.11 char* `digest_md5_response::username`

Definition at line 128 of file tokens.h.

Referenced by `_gsasl_digest_md5_client_step()`, `_gsasl_digest_md5_server_step()`, `digest_md5_free_response()`, `digest_md5_validate_response()`, and `main()`.

6.12.2.12 int [digest_md5_response::utf8](#)

Definition at line 136 of file [tokens.h](#).

Referenced by [_gsasl_digest_md5_client_step\(\)](#), and [digest_md5_validate\(\)](#).

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

- [tokens.h](#)

6.13 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](#)
- [Gsasl_client_callback_authorization_id cbc_authorization_id](#)
- [Gsasl_client_callback_authentication_id cbc_authentication_id](#)
- [Gsasl_client_callback_password cbc_password](#)
- [Gsasl_client_callback_passcode cbc_passcode](#)
- [Gsasl_client_callback_pin cbc_pin](#)
- [Gsasl_client_callback_anonymous cbc_anonymous](#)
- [Gsasl_client_callback_qop cbc_qop](#)
- [Gsasl_client_callback_maxbuf cbc_maxbuf](#)
- [Gsasl_client_callback_service cbc_service](#)
- [Gsasl_client_callback_realm cbc_realm](#)
- [Gsasl_server_callback_validate cbs_validate](#)
- [Gsasl_server_callback_securid cbs_securid](#)
- [Gsasl_server_callback_retrieve cbs_retrieve](#)
- [Gsasl_server_callback_cram_md5 cbs_cram_md5](#)
- [Gsasl_server_callback_digest_md5 cbs_digest_md5](#)
- [Gsasl_server_callback_external cbs_external](#)
- [Gsasl_server_callback_anonymous cbs_anonymous](#)
- [Gsasl_server_callback_realm cbs_realm](#)
- [Gsasl_server_callback_qop cbs_qop](#)
- [Gsasl_server_callback_maxbuf cbs_maxbuf](#)
- [Gsasl_server_callback_cipher cbs_cipher](#)
- [Gsasl_server_callback_service cbs_service](#)
- [Gsasl_server_callback_gssapi cbs_gssapi](#)

6.13.1 Detailed Description

Definition at line 40 of file internal.h.

6.13.2 Field Documentation

6.13.2.1 void* [Gsasl::application_hook](#)

Definition at line 48 of file internal.h.

Referenced by [gsasl_application_data_get\(\)](#), [gsasl_application_data_set\(\)](#), [gsasl_callback_hook_get\(\)](#), and [gsasl_callback_hook_set\(\)](#).

6.13.2.2 [Gsasl_callback_function](#) `Gsasl::cb`

Definition at line 47 of file internal.h.

Referenced by `gsasl_callback()`, and `gsasl_callback_set()`.

6.13.2.3 [Gsasl_client_callback_anonymous](#) `Gsasl::cbc_anonymous`

Definition at line 56 of file internal.h.

Referenced by `gsasl_client_callback_anonymous_get()`, and `gsasl_client_callback_anonymous_set()`.

6.13.2.4 [Gsasl_client_callback_authentication_id](#) `Gsasl::cbc_authentication_id`

Definition at line 52 of file internal.h.

Referenced by `gsasl_client_callback_authentication_id_get()`, and `gsasl_client_callback_authentication_id_set()`.

6.13.2.5 [Gsasl_client_callback_authorization_id](#) `Gsasl::cbc_authorization_id`

Definition at line 51 of file internal.h.

Referenced by `gsasl_client_callback_authorization_id_get()`, and `gsasl_client_callback_authorization_id_set()`.

6.13.2.6 [Gsasl_client_callback_maxbuf](#) `Gsasl::cbc_maxbuf`

Definition at line 58 of file internal.h.

Referenced by `gsasl_client_callback_maxbuf_get()`, and `gsasl_client_callback_maxbuf_set()`.

6.13.2.7 [Gsasl_client_callback_passcode](#) `Gsasl::cbc_passcode`

Definition at line 54 of file internal.h.

Referenced by `gsasl_client_callback_passcode_get()`, and `gsasl_client_callback_passcode_set()`.

6.13.2.8 [Gsasl_client_callback_password](#) `Gsasl::cbc_password`

Definition at line 53 of file internal.h.

Referenced by `gsasl_client_callback_password_get()`, and `gsasl_client_callback_password_set()`.

6.13.2.9 [Gsasl_client_callback_pin](#) `Gsasl::cbc_pin`

Definition at line 55 of file internal.h.

Referenced by `gsasl_client_callback_pin_get()`, and `gsasl_client_callback_pin_set()`.

6.13.2.10 Gsasl_client_callback_qop Gsasl::cbc_qop

Definition at line 57 of file internal.h.

Referenced by `gsasl_client_callback_qop_get()`, and `gsasl_client_callback_qop_set()`.

6.13.2.11 Gsasl_client_callback_realm Gsasl::cbc_realm

Definition at line 60 of file internal.h.

Referenced by `gsasl_client_callback_realm_get()`, and `gsasl_client_callback_realm_set()`.

6.13.2.12 Gsasl_client_callback_service Gsasl::cbc_service

Definition at line 59 of file internal.h.

Referenced by `gsasl_client_callback_service_get()`, and `gsasl_client_callback_service_set()`.

6.13.2.13 Gsasl_server_callback_anonymous Gsasl::cbs_anonymous

Definition at line 67 of file internal.h.

Referenced by `gsasl_server_callback_anonymous_get()`, and `gsasl_server_callback_anonymous_set()`.

6.13.2.14 Gsasl_server_callback_cipher Gsasl::cbs_cipher

Definition at line 71 of file internal.h.

Referenced by `gsasl_server_callback_cipher_get()`, and `gsasl_server_callback_cipher_set()`.

6.13.2.15 Gsasl_server_callback_cram_md5 Gsasl::cbs_cram_md5

Definition at line 64 of file internal.h.

Referenced by `gsasl_server_callback_cram_md5_get()`, and `gsasl_server_callback_cram_md5_set()`.

6.13.2.16 Gsasl_server_callback_digest_md5 Gsasl::cbs_digest_md5

Definition at line 65 of file internal.h.

Referenced by `gsasl_server_callback_digest_md5_get()`, and `gsasl_server_callback_digest_md5_set()`.

6.13.2.17 Gsasl_server_callback_external Gsasl::cbs_external

Definition at line 66 of file internal.h.

Referenced by `gsasl_server_callback_external_get()`, and `gsasl_server_callback_external_set()`.

6.13.2.18 Gsasl_server_callback_gssapi Gsasl::cbs_gssapi

Definition at line 73 of file internal.h.

Referenced by `gsasl_server_callback_gssapi_get()`, and `gsasl_server_callback_gssapi_set()`.

6.13.2.19 Gsasl_server_callback_maxbuf Gsasl::cbs_maxbuf

Definition at line 70 of file internal.h.

Referenced by `gsasl_server_callback_maxbuf_get()`, and `gsasl_server_callback_maxbuf_set()`.

6.13.2.20 Gsasl_server_callback_qop Gsasl::cbs_qop

Definition at line 69 of file internal.h.

Referenced by `gsasl_server_callback_qop_get()`, and `gsasl_server_callback_qop_set()`.

6.13.2.21 Gsasl_server_callback_realm Gsasl::cbs_realm

Definition at line 68 of file internal.h.

Referenced by `gsasl_server_callback_realm_get()`, and `gsasl_server_callback_realm_set()`.

6.13.2.22 Gsasl_server_callback_retrieve Gsasl::cbs_retrieve

Definition at line 63 of file internal.h.

Referenced by `gsasl_server_callback_retrieve_get()`, and `gsasl_server_callback_retrieve_set()`.

6.13.2.23 Gsasl_server_callback_securid Gsasl::cbs_securid

Definition at line 62 of file internal.h.

Referenced by `gsasl_server_callback_securid_get()`, and `gsasl_server_callback_securid_set()`.

6.13.2.24 Gsasl_server_callback_service Gsasl::cbs_service

Definition at line 72 of file internal.h.

Referenced by `gsasl_server_callback_service_get()`, and `gsasl_server_callback_service_set()`.

6.13.2.25 Gsasl_server_callback_validate Gsasl::cbs_validate

Definition at line 61 of file internal.h.

Referenced by `gsasl_server_callback_validate_get()`, and `gsasl_server_callback_validate_set()`.

6.13.2.26 Gsasl_mechanism* Gsasl::client_mechs

Definition at line 43 of file internal.h.

Referenced by `gsasl_client_mechlist()`, `gsasl_client_start()`, `gsasl_client_suggest_mechanism()`, `gsasl_client_support_p()`, `gsasl_done()`, and `gsasl_register()`.

6.13.2.27 size_t Gsasl::n_client_mechs

Definition at line 42 of file internal.h.

Referenced by `gsasl_client_mechlist()`, `gsasl_client_start()`, `gsasl_client_suggest_mechanism()`, `gsasl_client_support_p()`, `gsasl_done()`, and `gsasl_register()`.

6.13.2.28 `size_t Gsasl::n_server_mechs`

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

Referenced by `gsasl_register()`, `gsasl_server_mechlist()`, `gsasl_server_start()`, and `gsasl_server_support_p()`.

6.13.2.29 `Gsasl_mechanism* Gsasl::server_mechs`

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

Referenced by `gsasl_register()`, `gsasl_server_mechlist()`, `gsasl_server_start()`, and `gsasl_server_support_p()`.

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

- [internal.h](#)

6.14 Gsasl_mechanism Struct Reference

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

Data Fields

- const char * [name](#)
- [Gsasl_mechanism_functions](#) client
- [Gsasl_mechanism_functions](#) server

6.14.1 Detailed Description

Definition at line 52 of file `gsasl-mech.h`.

6.14.2 Field Documentation

6.14.2.1 struct [Gsasl_mechanism_functions](#) [Gsasl_mechanism::client](#)

Definition at line 56 of file `gsasl-mech.h`.

Referenced by `gsasl_decode()`, `gsasl_done()`, `gsasl_encode()`, `gsasl_finish()`, `gsasl_register()`, and `gsasl_step()`.

6.14.2.2 const char* [Gsasl_mechanism::name](#)

Definition at line 54 of file `gsasl-mech.h`.

Referenced by `gsasl_client_suggest_mechanism()`.

6.14.2.3 struct [Gsasl_mechanism_functions](#) [Gsasl_mechanism::server](#)

Definition at line 57 of file `gsasl-mech.h`.

Referenced by `gsasl_decode()`, `gsasl_encode()`, `gsasl_finish()`, `gsasl_register()`, and `gsasl_step()`.

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

- [gsasl-mech.h](#)

6.15 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

6.15.1 Detailed Description

Definition at line 39 of file gsasl-mech.h.

6.15.2 Field Documentation

6.15.2.1 [Gsasl_code_function Gsasl_mechanism_functions::decode](#)

Definition at line 47 of file gsasl-mech.h.

Referenced by [gsasl_decode\(\)](#).

6.15.2.2 [Gsasl_done_function Gsasl_mechanism_functions::done](#)

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

Referenced by [gsasl_done\(\)](#).

6.15.2.3 [Gsasl_code_function Gsasl_mechanism_functions::encode](#)

Definition at line 46 of file gsasl-mech.h.

Referenced by [gsasl_encode\(\)](#).

6.15.2.4 [Gsasl_finish_function Gsasl_mechanism_functions::finish](#)

Definition at line 45 of file gsasl-mech.h.

Referenced by [gsasl_finish\(\)](#).

6.15.2.5 [Gsasl_init_function Gsasl_mechanism_functions::init](#)

Definition at line 41 of file gsasl-mech.h.

Referenced by [gsasl_register\(\)](#).

6.15.2.6 [Gsasl_start_function Gsasl_mechanism_functions::start](#)

Definition at line 43 of file gsasl-mech.h.

6.15.2.7 [Gsasl_step_function Gsasl_mechanism_functions::step](#)

Definition at line 44 of file gsasl-mech.h.

Referenced by [gsasl_step\(\)](#).

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

- [gsasl-mech.h](#)

6.16 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](#)
- [void * application_data](#)

6.16.1 Detailed Description

Definition at line 78 of file internal.h.

6.16.2 Field Documentation

6.16.2.1 [char* Gsasl_session::anonymous_token](#)

Definition at line 86 of file internal.h.

Referenced by [gsasl_callback\(\)](#), and [gsasl_finish\(\)](#).

6.16.2.2 [void* Gsasl_session::application_data](#)

Definition at line 99 of file internal.h.

Referenced by [gsasl_appinfo_get\(\)](#), and [gsasl_appinfo_set\(\)](#).

6.16.2.3 [void* Gsasl_session::application_hook](#)

Definition at line 84 of file internal.h.

Referenced by [gsasl_session_hook_get\(\)](#), and [gsasl_session_hook_set\(\)](#).

6.16.2.4 char* Gsasl_session::authid

Definition at line 87 of file internal.h.

Referenced by gsasl_callback(), and gsasl_finish().

6.16.2.5 char* Gsasl_session::authzid

Definition at line 88 of file internal.h.

Referenced by gsasl_callback(), and gsasl_finish().

6.16.2.6 int Gsasl_session::clientp

Definition at line 81 of file internal.h.

Referenced by gsasl_decode(), gsasl_encode(), gsasl_finish(), and gsasl_step().

6.16.2.7 Gsasl* Gsasl_session::ctx

Definition at line 80 of file internal.h.

Referenced by gsasl_callback(), gsasl_client_ctx_get(), gsasl_ctx_get(), gsasl_init(), gsasl_property_get(), and gsasl_server_ctx_get().

6.16.2.8 char* Gsasl_session::gssapi_display_name

Definition at line 95 of file internal.h.

Referenced by gsasl_callback(), and gsasl_finish().

6.16.2.9 char* Gsasl_session::hostname

Definition at line 94 of file internal.h.

Referenced by gsasl_callback(), and gsasl_finish().

6.16.2.10 Gsasl_mechanism* Gsasl_session::mech

Definition at line 82 of file internal.h.

Referenced by gsasl_decode(), gsasl_encode(), gsasl_finish(), and gsasl_step().

6.16.2.11 void* Gsasl_session::mech_data

Definition at line 83 of file internal.h.

Referenced by gsasl_finish(), and gsasl_step().

6.16.2.12 char* Gsasl_session::passcode

Definition at line 90 of file internal.h.

Referenced by `gsasl_callback()`, and `gsasl_finish()`.

6.16.2.13 `char* Gsasl_session::password`

Definition at line 89 of file `internal.h`.

Referenced by `gsasl_callback()`, and `gsasl_finish()`.

6.16.2.14 `char* Gsasl_session::pin`

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

Referenced by `gsasl_callback()`, and `gsasl_finish()`.

6.16.2.15 `char* Gsasl_session::realm`

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

Referenced by `gsasl_finish()`.

6.16.2.16 `char* Gsasl_session::service`

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

Referenced by `gsasl_finish()`.

6.16.2.17 `char* Gsasl_session::suggestedpin`

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

Referenced by `gsasl_finish()`, and `gsasl_property_get()`.

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

- [internal.h](#)

Chapter 7

gsasl File Documentation

7.1 anonymous.h File Reference

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

Defines

- `#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`

7.1.1 Define Documentation

7.1.1.1 `#define GSASL_ANONYMOUS_NAME "ANONYMOUS"`

Definition at line 28 of file anonymous.h.

7.1.2 Function Documentation

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

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

References `GSASL_ANONYMOUS_TOKEN`, `GSASL_MALLOC_ERROR`, `GSASL_NO_-ANONYMOUS_TOKEN`, `GSASL_OK`, and `gsasl_property_get()`.

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

Definition at line 31 of file anonymous/server.c.

References `GSASL_ANONYMOUS_TOKEN`, `gsasl_callback()`, `GSASL_MECHANISM_PARSE_ERROR`, `GSASL_NEEDS_MORE`, `gsasl_property_set_raw()`, and `GSASL_VALIDATE_ANONYMOUS`.

7.1.3 Variable Documentation

7.1.3.1 [Gsasl_mechanism](#) `gsasl_anonymous_mechanism`

Definition at line 30 of file anonymous/mechinfo.c.

7.2 base64.c File Reference

```
#include "internal.h"
#include "base64.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)

7.2.1 Function Documentation

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

`gsasl_base64_from`:

Parameters:

- in* input byte array
- inlen* size of input byte array
- out* pointer to newly allocated output byte array
- outlen* 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 74 of file base64.c.

References GSASL_BASE64_ERROR, GSASL_MALLOC_ERROR, and GSASL_OK.

Referenced by `gsasl_step64()`.

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

`gsasl_base64_to`:

Parameters:

- in* input byte array
- inlen* size of input byte array
- out* pointer to newly allocated output byte array
- outlen* pointer to size of newly allocated output byte array

Encode data as base64. The 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 44 of file base64.c.

References GSASL_MALLOC_ERROR, and GSASL_OK.

Referenced by `_gsasl_digest_md5_client_start()`, `_gsasl_digest_md5_server_start()`, and `gsasl_step64()`.

7.3 callback.c File Reference

```
#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)

7.3.1 Function Documentation

7.3.1.1 int [gsasl_callback](#) ([Gsasl](#) * ctx, [Gsasl_session](#) * sctx, [Gsasl_property](#) prop)

[gsasl_callback](#):

Parameters:

ctx handle received from [gsasl_init\(\)](#), may be NULL to derive it from .

sctx session handle.

prop enumerated value of [Gsasl_property](#) type.

Invoke the application callback. The 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.

Note that if no callback has been set by the application, but the obsolete callback interface has been used, this function will translate the old callback interface into the new. This interface should be sufficient to invoke all callbacks, both new and old.

Return value: Returns whatever the application callback return, or [GSASL_NO_CALLBACK](#) if no application was known.

Since: 0.2.0

Definition at line 75 of file [callback.c](#).

References [Gsasl_session::anonymous_token](#), [Gsasl_session::authid](#), [Gsasl_session::authzid](#), [Gsasl::cb](#), [Gsasl_session::ctx](#), [GSASL_NO_CALLBACK](#), [GSASL_OK](#), [GSASL_PASSWORD](#), [gsasl_property_set\(\)](#), [gsasl_property_set_raw\(\)](#), [gsasl_server_callback_anonymous_get\(\)](#), [gsasl_server_callback_external_get\(\)](#), [gsasl_server_callback_gssapi_get\(\)](#), [gsasl_server_callback_retrieve_get\(\)](#), [gsasl_server_callback_securid_get\(\)](#), [gsasl_server_callback_validate_get\(\)](#), [GSASL_SUGGESTED_PIN](#), [GSASL_VALIDATE_ANONYMOUS](#), [GSASL_VALIDATE_EXTERNAL](#), [GSASL_VALIDATE_GSSAPI](#), [GSASL_VALIDATE_SECURID](#), [GSASL_VALIDATE_SIMPLE](#), [Gsasl_session::gssapi_display_name](#), [Gsasl_session::hostname](#), [Gsasl_session::passcode](#), [Gsasl_session::password](#), and [Gsasl_session::pin](#).

Referenced by [_gsasl_anonymous_server_step\(\)](#), [_gsasl_digest_md5_client_step\(\)](#), [_gsasl_external_server_step\(\)](#), [_gsasl_gssapi_server_step\(\)](#), [_gsasl_login_server_step\(\)](#), [_gsasl_plain_server_step\(\)](#), [_gsasl_securid_server_step\(\)](#), and [gsasl_property_get\(\)](#).

7.3.1.2 void* gsasl_callback_hook_get (Gsasl * ctx)

gsasl_callback_hook_get:

Parameters:

ctx 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 206 of file callback.c.

References Gsasl::application_hook.

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

gsasl_callback_hook_set:

Parameters:

ctx libgsasl handle.

hook 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 186 of file callback.c.

References Gsasl::application_hook.

7.3.1.4 void gsasl_callback_set (Gsasl * ctx, Gsasl_callback_function cb)

gsasl_callback_set:

Parameters:

ctx handle received from [gsasl_init\(\)](#).

cb 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 44 of file callback.c.

References Gsasl::cb.

7.3.1.5 void* gsasl_session_hook_get (Gsasl_session * sctx)

gsasl_session_hook_get:

Parameters:

sctx 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 246 of file callback.c.

References Gsasl_session::application_hook.

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

gsasl_session_hook_set:

Parameters:

sctx libgsasl session handle.

hook 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 226 of file callback.c.

References Gsasl_session::application_hook.

7.4 challenge.c File Reference

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

Defines

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

Functions

- void `cram_md5_challenge` (char challenge[CRAM_MD5_CHALLENGE_LEN])

7.4.1 Define Documentation

7.4.1.1 #define DIGIT(c)

Value:

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

Definition at line 60 of file challenge.c.

Referenced by `cram_md5_challenge()`.

7.4.1.2 #define NONCELEN 10

Definition at line 55 of file challenge.c.

Referenced by `cram_md5_challenge()`.

7.4.1.3 #define TEMPLATE "<XXXXXXXXXXXXXXXXXXXXXXXXX.0@localhost>"

Definition at line 56 of file challenge.c.

Referenced by `cram_md5_challenge()`.

7.4.2 Function Documentation

7.4.2.1 void cram_md5_challenge (char *challenge*[CRAM_MD5_CHALLENGE_LEN])

Definition at line 65 of file challenge.c.

References DIGIT, NONCELEN, and TEMPLATE.

Referenced by _gsasl_cram_md5_server_start().

7.5 challenge.h File Reference

Defines

- #define [CRAM_MD5_CHALLENGE_LEN](#) 35

Functions

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

7.5.1 Define Documentation

7.5.1.1 #define CRAM_MD5_CHALLENGE_LEN 35

Definition at line 26 of file challenge.h.

Referenced by [_gsasl_cram_md5_server_start\(\)](#).

7.5.2 Function Documentation

7.5.2.1 void [cram_md5_challenge](#) (char *challenge*[CRAM_MD5_CHALLENGE_LEN])

Definition at line 65 of file challenge.c.

References [DIGIT](#), [NONCELEN](#), and [TEMPLATE](#).

Referenced by [_gsasl_cram_md5_server_start\(\)](#).

7.6 client.c File Reference

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

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)`

7.6.1 Function Documentation

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

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

References GSASL_ANONYMOUS_TOKEN, GSASL_MALLOC_ERROR, GSASL_NO_ANONYMOUS_TOKEN, GSASL_OK, and `gsasl_property_get()`.

7.7 client.c File Reference

```
#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, const char *input, size_t input_len, char **output, size_t *output_len)`

7.7.1 Function Documentation

7.7.1.1 `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)`

Definition at line 40 of file `cram-md5/client.c`.

References `cram_md5_digest()`, `CRAM_MD5_DIGEST_LEN`, `GSASL_ALLOW_UNASSIGNED`, `GSASL_AUTHID`, `GSASL_MALLOC_ERROR`, `GSASL_NEEDS_MORE`, `GSASL_NO_AUTHID`, `GSASL_NO_PASSWORD`, `GSASL_OK`, `GSASL_PASSWORD`, `gsasl_property_get()`, and `gsasl_saslprep()`.

7.8 client.c File Reference

```
#include "digest-md5.h"
#include <stdlib.h>
#include <string.h>
#include "tokens.h"
#include "parser.h"
#include "printer.h"
#include "free.h"
#include "session.h"
#include "digesthmac.h"
```

Data Structures

- [struct _Gssl_digest_md5_client_state](#)

Defines

- [#define CNONCE_ENTROPY_BYTES 16](#)

Typedefs

- [typedef _Gssl_digest_md5_client_state _Gssl_digest_md5_client_state](#)

Functions

- [int _gssl_digest_md5_client_start \(Gssl_session *sctx, void **mech_data\)](#)
- [int _gssl_digest_md5_client_step \(Gssl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len\)](#)
- [void _gssl_digest_md5_client_finish \(Gssl_session *sctx, void *mech_data\)](#)
- [int _gssl_digest_md5_client_encode \(Gssl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len\)](#)
- [int _gssl_digest_md5_client_decode \(Gssl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len\)](#)

7.8.1 Define Documentation

7.8.1.1 [#define CNONCE_ENTROPY_BYTES 16](#)

Definition at line 44 of file digest-md5/client.c.

Referenced by [_gssl_digest_md5_client_start\(\)](#).

7.8.2 Typedef Documentation

7.8.2.1 typedef struct `_Gsasl_digest_md5_client_state` `_Gsasl_digest_md5_client_state`

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

7.8.3 Function Documentation

7.8.3.1 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`)

Definition at line 297 of file digest-md5/client.c.

References `digest_md5_decode()`, `GSASL_INTEGRITY_ERROR`, `GSASL_NEEDS_MORE`, `GSASL_OK`, `_Gsasl_digest_md5_client_state::kis`, `digest_md5_response::qop`, `_Gsasl_digest_md5_client_state::readseqnum`, and `_Gsasl_digest_md5_client_state::response`.

7.8.3.2 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`)

Definition at line 273 of file digest-md5/client.c.

References `digest_md5_encode()`, `GSASL_INTEGRITY_ERROR`, `GSASL_NEEDS_MORE`, `GSASL_OK`, `_Gsasl_digest_md5_client_state::kic`, `digest_md5_response::qop`, `_Gsasl_digest_md5_client_state::response`, and `_Gsasl_digest_md5_client_state::sendseqnum`.

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

Definition at line 258 of file digest-md5/client.c.

References `_Gsasl_digest_md5_client_state::challenge`, `digest_md5_free_challenge()`, `digest_md5_free_finish()`, `digest_md5_free_response()`, `_Gsasl_digest_md5_client_state::finish`, and `_Gsasl_digest_md5_client_state::response`.

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

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

References `CNONCE_ENTROPY_BYTES`, `gsasl_base64_to()`, `GSASL_MALLOC_ERROR`, `gsasl_nonce()`, and `GSASL_OK`.

7.8.3.5 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 93 of file digest-md5/client.c.

References `digest_md5_response::authzid`, `_Gsasl_digest_md5_client_state::challenge`, `digest_md5_response::cipher`, `digest_md5_response::cnonce`, `digest_md5_hmac()`, `DIGEST_MD5_LENGTH`, `digest_md5_parse_challenge()`, `digest_md5_parse_finish()`, `digest_md5_print_response()`, `DIGEST_MD5_RESPONSE_LENGTH`, `digest_md5_response::digesturi`, `_Gsasl_digest_md5_client_state::finish`, `GSASL_AUTHENTICATION_ERROR`, `GSASL_AUTHID`, `GSASL_AUTHZID`, `gsasl_callback()`, `GSASL_CRYPT_ERROR`, `GSASL_HOSTNAME`, `GSASL_MALLOC_ERROR`, `gsasl_md5()`,

GSASL_MECHANISM_CALLED_TOO_MANY_TIMES, GSASL_MECHANISM_PARSE_ERROR, GSASL_NEEDS_MORE, GSASL_NO_AUTHID, GSASL_NO_HOSTNAME, GSASL_NO_PASSWORD, GSASL_NO_SERVICE, GSASL_OK, GSASL_PASSWORD, gsasl_property_fast(), gsasl_property_get(), gsasl_property_set(), GSASL_REALM, GSASL_SERVICE, _Gsasl_digest_md5_client_state::kcc, _Gsasl_digest_md5_client_state::kcs, _Gsasl_digest_md5_client_state::kic, _Gsasl_digest_md5_client_state::kis, digest_md5_response::nc, digest_md5_response::nonce, digest_md5_challenge::nonce, digest_md5_challenge::nrealms, digest_md5_response::qop, digest_md5_response::realm, digest_md5_challenge::realms, _Gsasl_digest_md5_client_state::response, digest_md5_response::response, digest_md5_finish::rspauth, _Gsasl_digest_md5_client_state::secret, _Gsasl_digest_md5_client_state::step, digest_md5_response::username, and digest_md5_response::utf8.

7.9 client.c File Reference

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

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)`

7.9.1 Function Documentation

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

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

References GSASL_AUTHZID, GSASL_MALLOC_ERROR, GSASL_OK, and gsasl_property_get().

7.10 client.c File Reference

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

Data Structures

- [struct _Gsasl_gssapi_client_state](#)

Typedefs

- [typedef _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)

7.10.1 Typedef Documentation

7.10.1.1 typedef struct _Gsasl_gssapi_client_state _Gsasl_gssapi_client_state

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

7.10.2 Function Documentation

7.10.2.1 [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 330 of file gssapi/client.c.

References [_Gsasl_gssapi_client_state::context](#), [GSASL_GSSAPI_RELEASE_BUFFER_ERROR](#), [GSASL_GSSAPI_UNWRAP_ERROR](#), [GSASL_MALLOC_ERROR](#), [GSASL_OK](#), [GSASL_QOP_AUTH_CONF](#), [GSASL_QOP_AUTH_INT](#), [_Gsasl_gssapi_client_state::qop](#), and [_Gsasl_gssapi_client_state::step](#).

7.10.2.2 [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 275 of file gssapi/client.c.

References `_Gsasl_gssapi_client_state::context`, `GSASL_GSSAPI_RELEASE_BUFFER_ERROR`, `GSASL_GSSAPI_WRAP_ERROR`, `GSASL_MALLOC_ERROR`, `GSASL_OK`, `GSASL_QOP_AUTH_CONF`, `GSASL_QOP_AUTH_INT`, `_Gsasl_gssapi_client_state::qop`, and `_Gsasl_gssapi_client_state::step`.

7.10.2.3 `void _gsasl_gssapi_client_finish (Gsasl_session * sctx, void * mech_data)`

Definition at line 257 of file `gssapi/client.c`.

References `_Gsasl_gssapi_client_state::context`, and `_Gsasl_gssapi_client_state::service`.

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

Definition at line 59 of file `gssapi/client.c`.

References `GSASL_MALLOC_ERROR`, `GSASL_OK`, and `GSASL_QOP_AUTH`.

7.10.2.5 `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 78 of file `gssapi/client.c`.

References `_Gsasl_gssapi_client_state::context`, `GSASL_AUTHID`, `GSASL_GSSAPI_IMPORT_NAME_ERROR`, `GSASL_GSSAPI_INIT_SEC_CONTEXT_ERROR`, `GSASL_GSSAPI_RELEASE_BUFFER_ERROR`, `GSASL_GSSAPI_UNSUPPORTED_PROTECTION_ERROR`, `GSASL_GSSAPI_UNWRAP_ERROR`, `GSASL_GSSAPI_WRAP_ERROR`, `GSASL_HOSTNAME`, `GSASL_MALLOC_ERROR`, `GSASL_MECHANISM_CALLED_TOO_MANY_TIMES`, `GSASL_MECHANISM_PARSE_ERROR`, `GSASL_NEEDS_MORE`, `GSASL_NO_AUTHID`, `GSASL_NO_HOSTNAME`, `GSASL_NO_SERVICE`, `GSASL_OK`, `gsasl_property_get()`, `GSASL_SERVICE`, `_Gsasl_gssapi_client_state::qop`, `_Gsasl_gssapi_client_state::service`, and `_Gsasl_gssapi_client_state::step`.

7.11 client.c File Reference

```
#include "kerberos_v5.h"
#include "shared.h"
```

Data Structures

- struct [_Gsasl_kerberos_v5_client_state](#)

Defines

- #define [STEP_FIRST](#) 0
- #define [STEP_NONINFRA_SEND_ASREQ](#) 1
- #define [STEP_NONINFRA_WAIT_ASREP](#) 2
- #define [STEP_NONINFRA_SEND_APREQ](#) 3
- #define [STEP_NONINFRA_WAIT_APREP](#) 4
- #define [STEP_SUCCESS](#) 5

Functions

- int [_gsasl_kerberos_v5_client_init](#) (Gsasl_ctx *ctx)
- int [_gsasl_kerberos_v5_client_start](#) (Gsasl_session *sctx, void **mech_data)
- int [_gsasl_kerberos_v5_client_step](#) (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char *output, size_t *output_len)
- int [_gsasl_kerberos_v5_client_encode](#) (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)
- int [_gsasl_kerberos_v5_client_decode](#) (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char *output, size_t *output_len)
- int [_gsasl_kerberos_v5_client_finish](#) (Gsasl_session *sctx, void *mech_data)

7.11.1 Define Documentation

7.11.1.1 #define STEP_FIRST 0

Definition at line 81 of file kerberos_v5/client.c.

Referenced by [_gsasl_kerberos_v5_client_step\(\)](#).

7.11.1.2 #define STEP_NONINFRA_SEND_APREQ 3

Definition at line 84 of file kerberos_v5/client.c.

Referenced by [_gsasl_kerberos_v5_client_step\(\)](#).

7.11.1.3 #define STEP_NONINFRA_SEND_ASREQ 1

Definition at line 82 of file kerberos_v5/client.c.

Referenced by [_gsasl_kerberos_v5_client_step\(\)](#).

7.11.1.4 #define STEP_NONINFRA_WAIT_APREP 4

Definition at line 85 of file kerberos_v5/client.c.

Referenced by `_gsasl_kerberos_v5_client_step()`.

7.11.1.5 #define STEP_NONINFRA_WAIT_ASREP 2

Definition at line 83 of file kerberos_v5/client.c.

Referenced by `_gsasl_kerberos_v5_client_step()`.

7.11.1.6 #define STEP_SUCCESS 5

Definition at line 86 of file kerberos_v5/client.c.

Referenced by `_gsasl_kerberos_v5_client_step()`.

7.11.2 Function Documentation**7.11.2.1 `int _gsasl_kerberos_v5_client_decode (Gsasl_session * sctx, void * mech_data, const char * input, size_t input_len, char * output, size_t * output_len)`**

Definition at line 414 of file kerberos_v5/client.c.

References `_Gsasl_kerberos_v5_client_state::clientqop`, `GSASL_INTEGRITY_ERROR`, `GSASL_MALLOC_ERROR`, `GSASL_OK`, `GSASL_QOP_AUTH_CONF`, `GSASL_QOP_AUTH_INT`, and `_Gsasl_kerberos_v5_client_state::sessionkey`.

7.11.2.2 `int _gsasl_kerberos_v5_client_encode (Gsasl_session * sctx, void * mech_data, const char * input, size_t input_len, char ** output, size_t * output_len)`

Definition at line 367 of file kerberos_v5/client.c.

References `_Gsasl_kerberos_v5_client_state::clientqop`, `GSASL_INTEGRITY_ERROR`, `GSASL_KERBEROS_V5_INTERNAL_ERROR`, `GSASL_MALLOC_ERROR`, `GSASL_OK`, `GSASL_QOP_AUTH_CONF`, `GSASL_QOP_AUTH_INT`, `_Gsasl_kerberos_v5_client_state::safe`, `_Gsasl_kerberos_v5_client_state::sessionkey`, and `_Gsasl_kerberos_v5_client_state::sh`.

7.11.2.3 `int _gsasl_kerberos_v5_client_finish (Gsasl_session * sctx, void * mech_data)`

Definition at line 444 of file kerberos_v5/client.c.

References `GSASL_OK`, and `_Gsasl_kerberos_v5_client_state::sh`.

7.11.2.4 `int _gsasl_kerberos_v5_client_init (Gsasl_ctx * ctx)`

Definition at line 48 of file kerberos_v5/client.c.

References `GSASL_OK`, and `GSASL_UNKNOWN_MECHANISM`.

7.11.2.5 int _gsasl_kerberos_v5_client_start (Gsasl_session * sctx, void ** mech_data)

Definition at line 57 of file kerberos_v5/client.c.

References GSASL_KERBEROS_V5_INIT_ERROR, GSASL_MALLOC_ERROR, GSASL_OK, and GSASL_QOP_AUTH_INT.

7.11.2.6 int _gsasl_kerberos_v5_client_step (Gsasl_session * sctx, void * mech_data, const char * input, size_t input_len, char * output, size_t * output_len)

Definition at line 89 of file kerberos_v5/client.c.

References _Gsasl_kerberos_v5_client_state::ap, _Gsasl_kerberos_v5_client_state::as, BITMAP_LEN, CLIENT_HELLO_LEN, _Gsasl_kerberos_v5_client_state::clientmaxbuf, _Gsasl_kerberos_v5_client_state::clientqop, GSASL_AUTHENTICATION_ERROR, GSASL_CANNOT_GET_CTX, gsasl_client_callback_authentication_id_get(), gsasl_client_callback_authorization_id_get(), gsasl_client_callback_maxbuf_get(), gsasl_client_callback_password_get(), gsasl_client_callback_qop_get(), gsasl_client_callback_realm_get(), gsasl_client_callback_service_get(), gsasl_client_ctx_get(), GSASL_KERBEROS_V5_INTERNAL_ERROR, GSASL_MECHANISM_CALLED_TOO_MANY_TIMES, GSASL_MECHANISM_PARSE_ERROR, GSASL_NEED_CLIENT_PASSWORD_CALLBACK, GSASL_NEED_CLIENT_SERVICE_CALLBACK, GSASL_NEEDS_MORE, GSASL_OK, GSASL_QOP_AUTH, GSASL_QOP_AUTH_CONF, GSASL_QOP_AUTH_INT, GSASL_TOO_SMALL_BUFFER, MAXBUF_DEFAULT, MAXBUF_LEN, MUTUAL, SERVER_HELLO_LEN, _Gsasl_kerberos_v5_client_state::serverhello, _Gsasl_kerberos_v5_client_state::servermaxbuf, _Gsasl_kerberos_v5_client_state::servermutual, _Gsasl_kerberos_v5_client_state::serverqops, _Gsasl_kerberos_v5_client_state::sessionkey, _Gsasl_kerberos_v5_client_state::sh, _Gsasl_kerberos_v5_client_state::step, STEP_FIRST, STEP_NONINFRA_SEND_APREQ, STEP_NONINFRA_SEND_ASREQ, STEP_NONINFRA_WAIT_APREP, STEP_NONINFRA_WAIT_ASREP, and STEP_SUCCESS.

7.12 client.c File Reference

```
#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, 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)

7.12.1 Function Documentation

7.12.1.1 void [_gsasl_login_client_finish](#) ([Gsasl_session](#) * sctx, void * mech_data)

Definition at line 102 of file login/client.c.

7.12.1.2 int [_gsasl_login_client_start](#) ([Gsasl_session](#) * sctx, void ** mech_data)

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

References [GSASL_MALLOC_ERROR](#), and [GSASL_OK](#).

7.12.1.3 int [_gsasl_login_client_step](#) ([Gsasl_session](#) * sctx, void * mech_data, const char * input, size_t input_len, char ** output, size_t * output_len)

Definition at line 58 of file login/client.c.

References [GSASL_AUTHID](#), [GSASL_MECHANISM_CALLED_TOO_MANY_TIMES](#), [GSASL_NEEDS_MORE](#), [GSASL_NO_AUTHID](#), [GSASL_NO_PASSWORD](#), [GSASL_OK](#), [GSASL_PASSWORD](#), [gsasl_property_get\(\)](#), and [_Gsasl_login_client_state::step](#).

7.13 client.c File Reference

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

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)`

7.13.1 Function Documentation

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

Definition at line 37 of file plain/client.c.

References GSASL_AUTHID, GSASL_AUTHZID, GSASL_MALLOC_ERROR, GSASL_NO_AUTHID, GSASL_NO_PASSWORD, GSASL_OK, GSASL_PASSWORD, and `gsasl_property_get()`.

7.14 client.c File Reference

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

Defines

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

Functions

- int `_gsasl_securid_client_start` (`Gsasl_session` *sctx, 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, void *mech_data)

7.14.1 Define Documentation

7.14.1.1 #define `PASSCODE` "passcode"

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

Referenced by `_gsasl_securid_client_step()`, and `_gsasl_securid_server_step()`.

7.14.1.2 #define `PIN` "pin"

Definition at line 37 of file securid/client.c.

Referenced by `_gsasl_securid_client_step()`, and `_gsasl_securid_server_step()`.

7.14.2 Function Documentation

7.14.2.1 void `_gsasl_securid_client_finish` (`Gsasl_session` * sctx, void * mech_data)

Definition at line 163 of file securid/client.c.

References `_Gsasl_login_client_state::step`.

7.14.2.2 int `_gsasl_securid_client_start` (`Gsasl_session` * sctx, void ** mech_data)

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

References `GSASL_MALLOC_ERROR`, `GSASL_OK`, and `_Gsasl_login_client_state::step`.

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

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

References GSASL_AUTHID, GSASL_AUTHZID, GSASL_MALLOC_ERROR, GSASL_MECHANISM_CALLED_TOO_MANY_TIMES, GSASL_NO_AUTHID, GSASL_NO_PASSCODE, GSASL_NO_PIN, GSASL_OK, GSASL_PASSCODE, GSASL_PIN, `gsasl_property_get()`, `gsasl_property_set_raw()`, GSASL_SUGGESTED_PIN, PASSCODE, PIN, and `_Gsasl_login_client_state::step`.

7.15 cram-md5.h File Reference

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

Defines

- `#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`

7.15.1 Define Documentation

7.15.1.1 `#define GSASL_CRAM_MD5_NAME "CRAM-MD5"`

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

7.15.2 Function Documentation

7.15.2.1 `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)`

Definition at line 40 of file `cram-md5/client.c`.

References `cram_md5_digest()`, `CRAM_MD5_DIGEST_LEN`, `GSASL_ALLOW_UNASSIGNED`, `GSASL_AUTHID`, `GSASL_MALLOC_ERROR`, `GSASL_NEEDS_MORE`, `GSASL_NO_AUTHID`, `GSASL_NO_PASSWORD`, `GSASL_OK`, `GSASL_PASSWORD`, `gsasl_property_get()`, and `gsasl_saslprep()`.

7.15.2.2 `void _gsasl_cram_md5_server_finish (Gsasl_session * sctx, void * mech_data)`

Definition at line 124 of file `cram-md5/server.c`.

7.15.2.3 `int _gsasl_cram_md5_server_start (Gsasl_session * sctx, void ** mech_data)`

Definition at line 45 of file `cram-md5/server.c`.

References `cram_md5_challenge()`, `CRAM_MD5_CHALLENGE_LEN`, `GSASL_MALLOC_ERROR`, and `GSASL_OK`.

7.15.2.4 `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 61 of file `cram-md5/server.c`.

References `cram_md5_digest()`, `CRAM_MD5_DIGEST_LEN`, `GSASL_AUTHENTICATION_ERROR`, `GSASL_AUTHID`, `GSASL_MALLOC_ERROR`, `GSASL_MECHANISM_PARSE_ERROR`, `GSASL_NEEDS_MORE`, `GSASL_NO_PASSWORD`, `GSASL_OK`, `GSASL_PASSWORD`, `gsasl_property_get()`, `gsasl_property_set()`, `gsasl_saslprep()`, and `MD5LEN`.

7.15.3 Variable Documentation

7.15.3.1 `Gsasl_mechanism gsasl_cram_md5_mechanism`

Definition at line 30 of file `cram-md5/mechinfo.c`.

7.16 crypto.c File Reference

```
#include "internal.h"
#include "gc.h"
```

Functions

- int `gsasl_nonce` (char *data, size_t datalen)
- int `gsasl_random` (char *data, size_t datalen)
- int `gsasl_md5` (const char *in, size_t inlen, char *out[16])
- int `gsasl_hmac_md5` (const char *key, size_t keylen, const char *in, size_t inlen, char *outhash[16])

7.16.1 Function Documentation

7.16.1.1 int `gsasl_hmac_md5` (const char * *key*, size_t *keylen*, const char * *in*, size_t *inlen*, char * *outhash*[16])

`gsasl_hmac_md5`:

Parameters:

- key* input character array with key to use.
- keylen* length of input character array with key to use.
- in* input character array of data to hash.
- inlen* length of input character array of data to hash.
- outhash* newly allocated character array with keyed hash of data.

Compute keyed checksum of data using HMAC-MD5. The buffer must be deallocated by the caller.

Return value: Returns GSASL_OK iff successful.

Definition at line 92 of file `crypto.c`.

References GSASL_MALLOC_ERROR.

7.16.1.2 int `gsasl_md5` (const char * *in*, size_t *inlen*, char * *out*[16])

`gsasl_md5`:

Parameters:

- in* input character array of data to hash.
- inlen* length of input character array of data to hash.
- out* newly allocated character array with hash of data.

Compute hash of data using MD5. The buffer must be deallocated by the caller.

Return value: Returns GSASL_OK iff successful.

Definition at line 70 of file `crypto.c`.

References GSASL_MALLOC_ERROR.

Referenced by `_gsasl_digest_md5_client_step()`, and `_gsasl_digest_md5_server_step()`.

7.16.1.3 int gssl_nonce (char * data, size_t datalen)

gssl_nonce:

Parameters:

data output array to be filled with unpredictable random data.

datalen size of output array.

Store unpredictable data of given size in the provided buffer.

Return value: Returns GSASL_OK iff successful.

Definition at line 37 of file crypto.c.

Referenced by `_gssl_digest_md5_client_start()`, `_gssl_digest_md5_server_start()`, and `gssl_randomize()`.

7.16.1.4 int gssl_random (char * data, size_t datalen)

gssl_random:

Parameters:

data output array to be filled with strong random data.

datalen 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 53 of file crypto.c.

Referenced by `gssl_randomize()`.

7.17 digest-md5.h File Reference

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

Defines

- `#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`

7.17.1 Define Documentation

7.17.1.1 `#define GSASL_DIGEST_MD5_NAME "DIGEST-MD5"`

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

7.17.2 Function Documentation

7.17.2.1 `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)`

Definition at line 297 of file digest-md5/client.c.

References `digest_md5_decode()`, `GSASL_INTEGRITY_ERROR`, `GSASL_NEEDS_MORE`, `GSASL_OK`, `_Gsasl_digest_md5_client_state::kis`, `digest_md5_response::qop`, `_Gsasl_digest_md5_client_state::readseqnum`, and `_Gsasl_digest_md5_client_state::response`.

7.17.2.2 `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)`

Definition at line 273 of file digest-md5/client.c.

References `digest_md5_encode()`, `GSASL_INTEGRITY_ERROR`, `GSASL_NEEDS_MORE`, `GSASL_OK`, `_Gsasl_digest_md5_client_state::kic`, `digest_md5_response::qop`, `_Gsasl_digest_md5_client_state::response`, and `_Gsasl_digest_md5_client_state::sendseqnum`.

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

Definition at line 258 of file digest-md5/client.c.

References `_Gsasl_digest_md5_client_state::challenge`, `digest_md5_free_challenge()`, `digest_md5_free_finish()`, `digest_md5_free_response()`, `_Gsasl_digest_md5_client_state::finish`, and `_Gsasl_digest_md5_client_state::response`.

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

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

References `CNONCE_ENTROPY_BYTES`, `gsasl_base64_to()`, `GSASL_MALLOC_ERROR`, `gsasl_nonce()`, and `GSASL_OK`.

7.17.2.5 `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 93 of file digest-md5/client.c.

References `digest_md5_response::authzid`, `_Gsasl_digest_md5_client_state::challenge`, `digest_md5_response::cipher`, `digest_md5_response::cnonce`, `digest_md5_hmac()`, `DIGEST_MD5_LENGTH`, `digest_md5_parse_challenge()`, `digest_md5_parse_finish()`, `digest_md5_print_response()`, `DIGEST_MD5_RESPONSE_LENGTH`, `digest_md5_response::digesturi`, `_Gsasl_digest_md5_client_state::finish`, `GSASL_AUTHENTICATION_ERROR`, `GSASL_AUTHID`, `GSASL_AUTHZID`, `gsasl_callback()`, `GSASL_CRYPTO_ERROR`, `GSASL_HOSTNAME`, `GSASL_MALLOC_ERROR`, `gsasl_md5()`, `GSASL_MECHANISM_CALLED_TOO_MANY_TIMES`, `GSASL_MECHANISM_PARSE_ERROR`, `GSASL_NEEDS_MORE`, `GSASL_NO_AUTHID`, `GSASL_NO_HOSTNAME`, `GSASL_NO_PASSWORD`, `GSASL_NO_SERVICE`, `GSASL_OK`, `GSASL_PASSWORD`, `gsasl_property_fast()`, `gsasl_property_get()`, `gsasl_property_set()`, `GSASL_REALM`, `GSASL_SERVICE`, `_Gsasl_digest_md5_client_state::kcc`, `_Gsasl_digest_md5_client_state::kcs`, `_Gsasl_digest_md5_client_state::kic`, `_Gsasl_digest_md5_client_state::kis`, `digest_md5_response::nc`, `digest_md5_challenge::nonce`, `digest_md5_response::nonce`, `digest_md5_challenge::nrealms`, `digest_md5_response::qop`, `digest_md5_response::realm`, `digest_md5_challenge::realms`, `digest_md5_response::response`, `_Gsasl_digest_md5_client_state::response`, `digest_md5_finish::rspauth`, `_Gsasl_digest_md5_client_state::secret`, `_Gsasl_digest_md5_client_state::step`, `digest_md5_response::username`, and `digest_md5_response::utf8`.

7.17.2.6 `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)`

Definition at line 272 of file digest-md5/server.c.

References `digest_md5_decode()`, `GSASL_INTEGRITY_ERROR`, `GSASL_NEEDS_MORE`, `GSASL_OK`, `_Gsasl_digest_md5_server_state::kic`, `digest_md5_response::qop`, `_Gsasl_digest_md5_server_`

state::readseqnum, and _Gsasl_digest_md5_server_state::response.

7.17.2.7 `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)`

Definition at line 248 of file digest-md5/server.c.

References `digest_md5_encode()`, `GSASL_INTEGRITY_ERROR`, `GSASL_NEEDS_MORE`, `GSASL_OK`, `_Gsasl_digest_md5_server_state::kis`, `digest_md5_response::qop`, `_Gsasl_digest_md5_server_state::response`, and `_Gsasl_digest_md5_server_state::sendseqnum`.

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

Definition at line 233 of file digest-md5/server.c.

References `_Gsasl_digest_md5_server_state::challenge`, `digest_md5_free_challenge()`, `digest_md5_free_finish()`, `digest_md5_free_response()`, `_Gsasl_digest_md5_server_state::finish`, and `_Gsasl_digest_md5_server_state::response`.

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

Definition at line 63 of file digest-md5/server.c.

References `DIGEST_MD5_QOP_AUTH`, `DIGEST_MD5_QOP_AUTH_INT`, `gsasl_base64_to()`, `GSASL_MALLOC_ERROR`, `gsasl_nonce()`, `GSASL_OK`, and `NONCE_ENTROPY_BYTES`.

7.17.2.10 `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 97 of file digest-md5/server.c.

References `digest_md5_response::authzid`, `_Gsasl_digest_md5_server_state::challenge`, `digest_md5_response::cipher`, `digest_md5_response::cnonce`, `digest_md5_hmac()`, `DIGEST_MD5_LENGTH`, `digest_md5_parse_response()`, `digest_md5_print_challenge()`, `digest_md5_print_finish()`, `DIGEST_MD5_RESPONSE_LENGTH`, `digest_md5_validate()`, `digest_md5_response::digesturi`, `_Gsasl_digest_md5_server_state::finish`, `GSASL_AUTHENTICATION_ERROR`, `GSASL_AUTHID`, `GSASL_AUTHZID`, `GSASL_MALLOC_ERROR`, `gsasl_md5()`, `GSASL_MECHANISM_CALLED_TOO_MANY_TIMES`, `GSASL_MECHANISM_PARSE_ERROR`, `GSASL_NEEDS_MORE`, `GSASL_NO_PASSWORD`, `GSASL_OK`, `GSASL_PASSWORD`, `gsasl_property_get()`, `gsasl_property_set()`, `GSASL_REALM`, `digest_md5_response::nc`, `digest_md5_response::nonce`, `digest_md5_challenge::nrealms`, `digest_md5_response::qop`, `digest_md5_response::realm`, `digest_md5_challenge::realms`, `_Gsasl_digest_md5_server_state::response`, `digest_md5_response::response`, `digest_md5_finish::rspauth`, `_Gsasl_digest_md5_server_state::secret`, `_Gsasl_digest_md5_server_state::step`, and `digest_md5_response::username`.

7.17.3 Variable Documentation

7.17.3.1 [Gsasl_mechanism_gsasl_digest_md5_mechanism](#)

Definition at line 30 of file digest-md5/mechinfo.c.

7.18 digest.c File Reference

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

Defines

- #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])

7.18.1 Define Documentation

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

Definition at line 55 of file digest.c.

Referenced by `cram_md5_digest()`, and `digest_md5_hmac()`.

7.18.2 Function Documentation

7.18.2.1 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 58 of file digest.c.

References `HEXCHAR`.

Referenced by `_gsasl_cram_md5_client_step()`, and `_gsasl_cram_md5_server_step()`.

7.19 digest.h File Reference

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

Defines

- #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])

7.19.1 Define Documentation

7.19.1.1 #define CRAM_MD5_DIGEST_LEN 32

Definition at line 29 of file digest.h.

Referenced by [_gsasl_cram_md5_client_step\(\)](#), and [_gsasl_cram_md5_server_step\(\)](#).

7.19.2 Function Documentation

7.19.2.1 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 58 of file digest.c.

References [HEXCHAR](#).

Referenced by [_gsasl_cram_md5_client_step\(\)](#), and [_gsasl_cram_md5_server_step\(\)](#).

7.20 digestmac.c File Reference

```
#include "digestmac.h"
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <gc.h>
```

Defines

- #define [HEXCHAR\(c\)](#) ((c & 0x0F) > 9 ? 'a' + (c & 0x0F) - 10 : '0' + (c & 0x0F))
- #define [QOP_AUTH](#) "auth"
- #define [QOP_AUTH_INT](#) "auth-int"
- #define [QOP_AUTH_CONF](#) "auth-conf"
- #define [A2_PRE](#) "AUTHENTICATE:"
- #define [A2_POST](#) ":00000000000000000000000000000000"
- #define [COLON](#) ":"
- #define [MD5LEN](#) 16
- #define [RESPONSE_LENGTH](#) 32
- #define [RSPAUTH_LENGTH](#) RESPONSE_LENGTH
- #define [DERIVE_CLIENT_INTEGRITY_KEY_STRING](#) "Digest session key to client-to-server signing key magic constant"
- #define [DERIVE_CLIENT_INTEGRITY_KEY_STRING_LEN](#) 65
- #define [DERIVE_SERVER_INTEGRITY_KEY_STRING](#) "Digest session key to server-to-client signing key magic constant"
- #define [DERIVE_SERVER_INTEGRITY_KEY_STRING_LEN](#) 65
- #define [DERIVE_CLIENT_CONFIDENTIALITY_KEY_STRING](#) "Digest H(A1) to client-to-server sealing key magic constant"
- #define [DERIVE_CLIENT_CONFIDENTIALITY_KEY_STRING_LEN](#) 59
- #define [DERIVE_SERVER_CONFIDENTIALITY_KEY_STRING](#) "Digest H(A1) to server-to-client sealing key magic constant"
- #define [DERIVE_SERVER_CONFIDENTIALITY_KEY_STRING_LEN](#) 59

Functions

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

7.20.1 Define Documentation

7.20.1.1 #define [A2_POST](#) ":00000000000000000000000000000000"

Definition at line 49 of file digestmac.c.

Referenced by [digest_md5_hmac\(\)](#).

7.20.1.2 #define A2_PRE "AUTHENTICATE:"

Definition at line 48 of file digestmac.c.

Referenced by digest_md5_hmac().

7.20.1.3 #define COLON ":"

Definition at line 50 of file digestmac.c.

Referenced by digest_md5_hmac().

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

Definition at line 60 of file digestmac.c.

Referenced by digest_md5_hmac().

7.20.1.5 #define DERIVE_CLIENT_CONFIDENTIALITY_KEY_STRING_LEN 59

Definition at line 62 of file digestmac.c.

Referenced by digest_md5_hmac().

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

Definition at line 54 of file digestmac.c.

Referenced by digest_md5_hmac().

7.20.1.7 #define DERIVE_CLIENT_INTEGRITY_KEY_STRING_LEN 65

Definition at line 56 of file digestmac.c.

Referenced by digest_md5_hmac().

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

Definition at line 63 of file digestmac.c.

Referenced by digest_md5_hmac().

7.20.1.9 #define DERIVE_SERVER_CONFIDENTIALITY_KEY_STRING_LEN 59

Definition at line 65 of file digestmac.c.

Referenced by digest_md5_hmac().

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

Definition at line 57 of file digestmac.c.

Referenced by digest_md5_hmac().

7.20.1.11 #define DERIVE_SERVER_INTEGRITY_KEY_STRING_LEN 65

Definition at line 59 of file digestmac.c.

Referenced by digest_md5_hmac().

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

Definition at line 42 of file digestmac.c.

7.20.1.13 #define MD5LEN 16

Definition at line 51 of file digestmac.c.

7.20.1.14 #define QOP_AUTH "auth"

Definition at line 44 of file digestmac.c.

Referenced by digest_md5_hmac().

7.20.1.15 #define QOP_AUTH_CONF "auth-conf"

Definition at line 46 of file digestmac.c.

Referenced by digest_md5_hmac().

7.20.1.16 #define QOP_AUTH_INT "auth-int"

Definition at line 45 of file digestmac.c.

Referenced by digest_md5_hmac().

7.20.1.17 #define RESPONSE_LENGTH 32

Definition at line 52 of file digestmac.c.

7.20.1.18 #define RSPAUTH_LENGTH RESPONSE_LENGTH

Definition at line 53 of file digestmac.c.

7.20.2 Function Documentation

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

Definition at line 81 of file digesthmac.c.

References A2_POST, A2_PRE, COLON, DERIVE_CLIENT_CONFIDENTIALITY_KEY_STRING, DERIVE_CLIENT_CONFIDENTIALITY_KEY_STRING_LEN, DERIVE_CLIENT_INTEGRITY_KEY_STRING, DERIVE_CLIENT_INTEGRITY_KEY_STRING_LEN, DERIVE_SERVER_CONFIDENTIALITY_KEY_STRING, DERIVE_SERVER_CONFIDENTIALITY_KEY_STRING_LEN, DERIVE_SERVER_INTEGRITY_KEY_STRING, DERIVE_SERVER_INTEGRITY_KEY_STRING_LEN, DIGEST_MD5_CIPHER_RC4_40, DIGEST_MD5_CIPHER_RC4_56, DIGEST_MD5_QOP_AUTH, DIGEST_MD5_QOP_AUTH_CONF, DIGEST_MD5_QOP_AUTH_INT, HEX_CHAR, QOP_AUTH, QOP_AUTH_CONF, and QOP_AUTH_INT.

Referenced by `_gsasl_digest_md5_client_step()`, `_gsasl_digest_md5_server_step()`, and `main()`.

7.21 digestmac.h File Reference

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

Functions

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

7.21.1 Function Documentation

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

7.22 done.c File Reference

```
#include "internal.h"
```

Functions

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

7.22.1 Function Documentation

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

[gsasl_done](#):

Parameters:

- ctx* libgsasl handle.

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

Definition at line 33 of file [done.c](#).

References [Gsasl_mechanism::client](#), [Gsasl::client_mechs](#), [Gsasl_mechanism_functions::done](#), and [Gsasl::n_client_mechs](#).

Referenced by [gsasl_init](#)().

7.23 doxygen.c File Reference

7.24 error.c File Reference

```
#include "internal.h"
#include "gettext.h"
```

Defines

- #define `_(String) dgettext (PACKAGE, String)`
- #define `gettext_noop(String) String`
- #define `N_(String) gettext_noop (String)`

Functions

- `const char * gsasl_strerror (int err)`

7.24.1 Define Documentation

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

Definition at line 27 of file error.c.

Referenced by `gsasl_strerror()`.

7.24.1.2 #define `gettext_noop(String) String`

Definition at line 28 of file error.c.

7.24.1.3 #define `N_(String) gettext_noop (String)`

Definition at line 29 of file error.c.

7.24.2 Function Documentation

7.24.2.1 `const char* gsasl_strerror (int err)`

`gsasl_strerror`:

Parameters:

err libgsasl error code

Convert return code to human readable string.

Return value: Returns a pointer to a statically allocated string containing a description of the error with the error value . This string can be used to output a diagnostic message to the user.

Definition at line 42 of file error.c.

References `_`, `GSASL_AUTHENTICATION_ERROR`, `GSASL_BASE64_ERROR`, `GSASL_CANNOT_GET_CTX`, `GSASL_CRYPTO_ERROR`, `GSASL_FCLOSE_ERROR`, `GSASL_FOPEN_ERROR`,

GSASL_GSSAPI_ACCEPT_SEC_CONTEXT_ERROR, GSASL_GSSAPI_ACQUIRE_CRED_ERROR, GSASL_GSSAPI_DISPLAY_NAME_ERROR, GSASL_GSSAPI_IMPORT_NAME_ERROR, GSASL_GSSAPI_INIT_SEC_CONTEXT_ERROR, GSASL_GSSAPI_RELEASE_BUFFER_ERROR, GSASL_GSSAPI_UNSUPPORTED_PROTECTION_ERROR, GSASL_GSSAPI_UNWRAP_ERROR, GSASL_GSSAPI_WRAP_ERROR, GSASL_INTEGRITY_ERROR, GSASL_INVALID_HANDLE, GSASL_MALLOC_ERROR, GSASL_MECHANISM_CALLED_TOO_MANY_TIMES, GSASL_MECHANISM_PARSE_ERROR, GSASL_NEED_CLIENT_ANONYMOUS_CALLBACK, GSASL_NEED_CLIENT_AUTHENTICATION_ID_CALLBACK, GSASL_NEED_CLIENT_AUTHORIZATION_ID_CALLBACK, GSASL_NEED_CLIENT_PASSCODE_CALLBACK, GSASL_NEED_CLIENT_PASSWORD_CALLBACK, GSASL_NEED_CLIENT_PIN_CALLBACK, GSASL_NEED_CLIENT_SERVICE_CALLBACK, GSASL_NEED_SERVER_ANONYMOUS_CALLBACK, GSASL_NEED_SERVER_CRAM_MD5_CALLBACK, GSASL_NEED_SERVER_DIGEST_MD5_CALLBACK, GSASL_NEED_SERVER_EXTERNAL_CALLBACK, GSASL_NEED_SERVER_GSSAPI_CALLBACK, GSASL_NEED_SERVER_REALM_CALLBACK, GSASL_NEED_SERVER_RETRIEVE_CALLBACK, GSASL_NEED_SERVER_SECURID_CALLBACK, GSASL_NEED_SERVER_SERVICE_CALLBACK, GSASL_NEED_SERVER_VALIDATE_CALLBACK, GSASL_NEEDS_MORE, GSASL_NO_ANONYMOUS_TOKEN, GSASL_NO_AUTHID, GSASL_NO_AUTHZID, GSASL_NO_CALLBACK, GSASL_NO_CLIENT_CODE, GSASL_NO_HOSTNAME, GSASL_NO_MORE_REALMS, GSASL_NO_PASSCODE, GSASL_NO_PASSWORD, GSASL_NO_PIN, GSASL_NO_SERVER_CODE, GSASL_NO_SERVICE, GSASL_OK, GSASL_SASLPREP_ERROR, GSASL_TOO_SMALL_BUFFER, GSASL_UNICODE_NORMALIZATION_ERROR, and GSASL_UNKNOWN_MECHANISM.

7.25 external.h File Reference

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

Defines

- `#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`

7.25.1 Define Documentation

7.25.1.1 `#define GSASL_EXTERNAL_NAME "EXTERNAL"`

Definition at line 28 of file external.h.

7.25.2 Function Documentation

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

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

References `GSASL_AUTHZID`, `GSASL_MALLOC_ERROR`, `GSASL_OK`, and `gsasl_property_get()`.

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

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

References `GSASL_AUTHZID`, `gsasl_callback()`, `GSASL_MECHANISM_PARSE_ERROR`, `GSASL_NEEDS_MORE`, `gsasl_property_set()`, `gsasl_property_set_raw()`, and `GSASL_VALIDATE_EXTERNAL`.

7.25.3 Variable Documentation

7.25.3.1 `Gsasl_mechanism gsasl_external_mechanism`

Definition at line 30 of file external/mechinfo.c.

7.26 free.c File Reference

```
#include "free.h"  
#include <stddef.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)

7.26.1 Function Documentation

7.26.1.1 void [digest_md5_free_challenge](#) ([digest_md5_challenge](#) *c)

Definition at line 36 of file free.c.

References [digest_md5_challenge::nonce](#), [digest_md5_challenge::nrealms](#), and [digest_md5_challenge::realms](#).

Referenced by [_gsasl_digest_md5_client_finish\(\)](#), and [_gsasl_digest_md5_server_finish\(\)](#).

7.26.1.2 void [digest_md5_free_finish](#) ([digest_md5_finish](#) *f)

Definition at line 70 of file free.c.

Referenced by [_gsasl_digest_md5_client_finish\(\)](#), and [_gsasl_digest_md5_server_finish\(\)](#).

7.26.1.3 void [digest_md5_free_response](#) ([digest_md5_response](#) *r)

Definition at line 51 of file free.c.

References [digest_md5_response::authzid](#), [digest_md5_response::cnonce](#), [digest_md5_response::digesturi](#), [digest_md5_response::nonce](#), [digest_md5_response::realm](#), and [digest_md5_response::username](#).

Referenced by [_gsasl_digest_md5_client_finish\(\)](#), and [_gsasl_digest_md5_server_finish\(\)](#).

7.27 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)

7.27.1 Function Documentation

7.27.1.1 void [digest_md5_free_challenge](#) ([digest_md5_challenge](#) * c)

Definition at line 36 of file free.c.

References [digest_md5_challenge::nonce](#), [digest_md5_challenge::nrealms](#), and [digest_md5_challenge::realms](#).

Referenced by [_gsasl_digest_md5_client_finish\(\)](#), and [_gsasl_digest_md5_server_finish\(\)](#).

7.27.1.2 void [digest_md5_free_finish](#) ([digest_md5_finish](#) * f)

Definition at line 70 of file free.c.

Referenced by [_gsasl_digest_md5_client_finish\(\)](#), and [_gsasl_digest_md5_server_finish\(\)](#).

7.27.1.3 void [digest_md5_free_response](#) ([digest_md5_response](#) * r)

Definition at line 51 of file free.c.

References [digest_md5_response::authzid](#), [digest_md5_response::cnonce](#), [digest_md5_response::digesturi](#), [digest_md5_response::nonce](#), [digest_md5_response::realm](#), and [digest_md5_response::username](#).

Referenced by [_gsasl_digest_md5_client_finish\(\)](#), and [_gsasl_digest_md5_server_finish\(\)](#).

7.28 getsubopt.c File Reference

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

Functions

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

7.28.1 Function Documentation

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

Definition at line 46 of file getsubopt.c.

7.29 gsasl-compat.h File Reference

Defines

- #define `__attribute__`(Spec)

Typedefs

- typedef `Gsasl` `Gsasl_ctx` `__attribute__` ((deprecated))
- typedef `Gsasl_session` `Gsasl_session_ctx` `__attribute__` ((deprecated))
- typedef int(*) `Gsasl_client_callback_anonymous` (`Gsasl_session` *sctx, char *out, size_t *outlen)
- typedef int(*) `Gsasl_client_callback_authentication_id` (`Gsasl_session` *sctx, char *out, size_t *outlen)
- typedef int(*) `Gsasl_client_callback_authorization_id` (`Gsasl_session` *sctx, char *out, size_t *outlen)
- typedef int(*) `Gsasl_client_callback_password` (`Gsasl_session` *sctx, char *out, size_t *outlen)
- typedef int(*) `Gsasl_client_callback_passcode` (`Gsasl_session` *sctx, char *out, size_t *outlen)
- typedef int(*) `Gsasl_client_callback_pin` (`Gsasl_session` *sctx, char *suggestion, char *out, size_t *outlen)
- typedef int(*) `Gsasl_client_callback_service` (`Gsasl_session` *sctx, char *service, size_t *servicelen, char *hostname, size_t *hostnamelen, char *servicename, size_t *servicenamelen)
- typedef `Gsasl_qop`(*) `Gsasl_client_callback_qop` (`Gsasl_session` *sctx, `Gsasl_qop` serverqops)
- typedef size_t(*) `Gsasl_client_callback_maxbuf` (`Gsasl_session` *sctx, size_t servermaxbuf)
- typedef int(*) `Gsasl_client_callback_realm` (`Gsasl_session` *sctx, char *out, size_t *outlen)
- typedef int(*) `Gsasl_server_callback_retrieve` (`Gsasl_session` *sctx, const char *authentication_id, const char *authorization_id, const char *realm, char *key, size_t *keylen)
- typedef int(*) `Gsasl_server_callback_validate` (`Gsasl_session` *sctx, const char *authorization_id, const char *authentication_id, const char *password)
- typedef int(*) `Gsasl_server_callback_gssapi` (`Gsasl_session` *sctx, const char *clientname, const char *authentication_id)
- typedef int(*) `Gsasl_server_callback_securid` (`Gsasl_session` *sctx, const char *authentication_id, const char *authorization_id, const char *passcode, char *pin, char *suggestpin, size_t *suggestpinlen)
- typedef int(*) `Gsasl_server_callback_cram_md5` (`Gsasl_session` *sctx, char *username, char *challenge, char *response)
- typedef int(*) `Gsasl_server_callback_digest_md5` (`Gsasl_session` *sctx, char *username, char *realm, char *secrethash)
- typedef int(*) `Gsasl_server_callback_service` (`Gsasl_session` *sctx, char *service, size_t *servicelen, char *hostname, size_t *hostnamelen)
- typedef int(*) `Gsasl_server_callback_external` (`Gsasl_session` *sctx)
- typedef int(*) `Gsasl_server_callback_anonymous` (`Gsasl_session` *sctx, const char *token)
- typedef int(*) `Gsasl_server_callback_realm` (`Gsasl_session` *sctx, char *out, size_t *outlen, size_t nth)
- typedef `Gsasl_qop`(*) `Gsasl_server_callback_qop` (`Gsasl_session` *sctx)
- typedef size_t(*) `Gsasl_server_callback_maxbuf` (`Gsasl_session` *sctx)
- typedef `Gsasl_cipher`(*) `Gsasl_server_callback_cipher` (`Gsasl_session` *sctx)

Enumerations

- enum {
 - GSASL_TOO_SMALL_BUFFER = 4, GSASL_FOPEN_ERROR = 5, GSASL_FCLOSE_ERROR = 6, GSASL_GCRYPT_ERROR = GSASL_CRYPT_ERROR,
 - GSASL_CANNOT_GET_CTX = 32, GSASL_NEED_CLIENT_ANONYMOUS_CALLBACK = 11, GSASL_NEED_CLIENT_PASSWORD_CALLBACK = 12, GSASL_NEED_CLIENT_PASSCODE_CALLBACK = 13,
 - GSASL_NEED_CLIENT_PIN_CALLBACK = 14, GSASL_NEED_CLIENT_AUTHORIZATION_ID_CALLBACK = 15, GSASL_NEED_CLIENT_AUTHENTICATION_ID_CALLBACK = 16, GSASL_NEED_CLIENT_SERVICE_CALLBACK = 17,
 - GSASL_NEED_SERVER_VALIDATE_CALLBACK = 18, GSASL_NEED_SERVER_CRAM_MD5_CALLBACK = 19, GSASL_NEED_SERVER_DIGEST_MD5_CALLBACK = 20, GSASL_NEED_SERVER_EXTERNAL_CALLBACK = 21,
 - GSASL_NEED_SERVER_ANONYMOUS_CALLBACK = 22, GSASL_NEED_SERVER_REALM_CALLBACK = 23, GSASL_NEED_SERVER_SECURID_CALLBACK = 24, GSASL_NEED_SERVER_SERVICE_CALLBACK = 25,
 - GSASL_NEED_SERVER_GSSAPI_CALLBACK = 26, GSASL_NEED_SERVER_RETRIEVE_CALLBACK = 27, GSASL_UNICODE_NORMALIZATION_ERROR = 28, GSASL_NO_MORE_REALMS = 34,
 - GSASL_INVALID_HANDLE = 50 }

Functions

- int gssapi_client_listmech (Gssapi *ctx, char *out, size_t *outlen) __attribute__((deprecated))
- int gssapi_server_listmech (Gssapi *ctx, char *out, size_t *outlen) __attribute__((deprecated))
- int gssapi_client_step (Gssapi_session *sctx, const char *input, size_t input_len, char *output, size_t *output_len) __attribute__((deprecated))
- int gssapi_client_step_base64 (Gssapi_session *sctx, const char *b64input, char *b64output, size_t b64output_len) __attribute__((deprecated))
- int gssapi_server_step (Gssapi_session *sctx, const char *input, size_t input_len, char *output, size_t *output_len) __attribute__((deprecated))
- int gssapi_server_step_base64 (Gssapi_session *sctx, const char *b64input, char *b64output, size_t b64output_len) __attribute__((deprecated))
- void gssapi_client_finish (Gssapi_session *sctx) __attribute__((deprecated))
- void gssapi_server_finish (Gssapi_session *sctx) __attribute__((deprecated))
- Gssapi * gssapi_client_ctx_get (Gssapi_session *sctx) __attribute__((deprecated))
- Gssapi * gssapi_server_ctx_get (Gssapi_session *sctx) __attribute__((deprecated))
- void gssapi_client_application_data_set (Gssapi_session *sctx, void *application_data) __attribute__((deprecated))
- void * gssapi_client_application_data_get (Gssapi_session *sctx) __attribute__((deprecated))
- void gssapi_server_application_data_set (Gssapi_session *sctx, void *application_data) __attribute__((deprecated))
- void * gssapi_server_application_data_get (Gssapi_session *sctx) __attribute__((deprecated))
- int gssapi_randomize (int strong, char *data, size_t datalen) __attribute__((deprecated))
- Gssapi * gssapi_ctx_get (Gssapi_session *sctx) __attribute__((deprecated))
- int gssapi_encode_inline (Gssapi_session *sctx, const char *input, size_t input_len, char *output, size_t *output_len) __attribute__((deprecated))
- int gssapi_decode_inline (Gssapi_session *sctx, const char *input, size_t input_len, char *output, size_t *output_len) __attribute__((deprecated))

- void `gsasl_application_data_set` (Gsasl *ctx, void *appdata) `__attribute__((deprecated))`
- void * `gsasl_application_data_get` (Gsasl *ctx) `__attribute__((deprecated))`
- void `gsasl_appinfo_set` (Gsasl_session *sctx, void *appdata) `__attribute__((deprecated))`
- void * `gsasl_appinfo_get` (Gsasl_session *sctx) `__attribute__((deprecated))`
- const char * `gsasl_server_suggest_mechanism` (Gsasl *ctx, const char *mechlist) `__attribute__((deprecated))`
- int `gsasl_base64_encode` (char const *src, size_t srclength, char *target, size_t targsize) `__attribute__((deprecated))`
- int `gsasl_base64_decode` (char const *src, char *target, size_t targsize) `__attribute__((deprecated))`
- char * `gsasl_stringprep_nfkc` (const char *in, ssize_t len) `__attribute__((deprecated))`
- char * `gsasl_stringprep_saslprep` (const char *in, int *stringprep_rc) `__attribute__((deprecated))`
- char * `gsasl_stringprep_trace` (const char *in, int *stringprep_rc) `__attribute__((deprecated))`
- int `gsasl_md5pwd_get_password` (const char *filename, const char *username, char *key, size_t *keylen) `__attribute__((deprecated))`
- void `gsasl_client_callback_authorization_id_set` (Gsasl *ctx, Gsasl_client_callback_authorization_id cb) `__attribute__((deprecated))`
- Gsasl_client_callback_authorization_id `gsasl_client_callback_authorization_id_get` (Gsasl *ctx) `__attribute__((deprecated))`
- void `gsasl_client_callback_authentication_id_set` (Gsasl *ctx, Gsasl_client_callback_authentication_id cb) `__attribute__((deprecated))`
- Gsasl_client_callback_authentication_id `gsasl_client_callback_authentication_id_get` (Gsasl *ctx) `__attribute__((deprecated))`
- void `gsasl_client_callback_anonymous_set` (Gsasl *ctx, Gsasl_client_callback_anonymous cb) `__attribute__((deprecated))`
- Gsasl_client_callback_anonymous `gsasl_client_callback_anonymous_get` (Gsasl *ctx) `__attribute__((deprecated))`
- void `gsasl_client_callback_password_set` (Gsasl *ctx, Gsasl_client_callback_password cb) `__attribute__((deprecated))`
- Gsasl_client_callback_password `gsasl_client_callback_password_get` (Gsasl *ctx) `__attribute__((deprecated))`
- void `gsasl_client_callback_passcode_set` (Gsasl *ctx, Gsasl_client_callback_passcode cb) `__attribute__((deprecated))`
- Gsasl_client_callback_passcode `gsasl_client_callback_passcode_get` (Gsasl *ctx) `__attribute__((deprecated))`
- void `gsasl_client_callback_pin_set` (Gsasl *ctx, Gsasl_client_callback_pin cb) `__attribute__((deprecated))`
- Gsasl_client_callback_pin `gsasl_client_callback_pin_get` (Gsasl *ctx) `__attribute__((deprecated))`
- void `gsasl_client_callback_service_set` (Gsasl *ctx, Gsasl_client_callback_service cb) `__attribute__((deprecated))`
- Gsasl_client_callback_service `gsasl_client_callback_service_get` (Gsasl *ctx) `__attribute__((deprecated))`
- void `gsasl_client_callback_qop_set` (Gsasl *ctx, Gsasl_client_callback_qop cb) `__attribute__((deprecated))`
- Gsasl_client_callback_qop `gsasl_client_callback_qop_get` (Gsasl *ctx) `__attribute__((deprecated))`
- void `gsasl_client_callback_maxbuf_set` (Gsasl *ctx, Gsasl_client_callback_maxbuf cb) `__attribute__((deprecated))`
- Gsasl_client_callback_maxbuf `gsasl_client_callback_maxbuf_get` (Gsasl *ctx) `__attribute__((deprecated))`
- void `gsasl_client_callback_realm_set` (Gsasl *ctx, Gsasl_client_callback_realm cb) `__attribute__((deprecated))`
- Gsasl_client_callback_realm `gsasl_client_callback_realm_get` (Gsasl *ctx) `__attribute__((deprecated))`

- void `gssapi_server_callback_validate_set` (`Gssapi *ctx`, `Gssapi_server_callback_validate cb`) `__attribute__((deprecated))`
- `Gssapi_server_callback_validate` `gssapi_server_callback_validate_get` (`Gssapi *ctx`) `__attribute__((deprecated))`
- void `gssapi_server_callback_retrieve_set` (`Gssapi *ctx`, `Gssapi_server_callback_retrieve cb`) `__attribute__((deprecated))`
- `Gssapi_server_callback_retrieve` `gssapi_server_callback_retrieve_get` (`Gssapi *ctx`) `__attribute__((deprecated))`
- void `gssapi_server_callback_cram_md5_set` (`Gssapi *ctx`, `Gssapi_server_callback_cram_md5 cb`) `__attribute__((deprecated))`
- `Gssapi_server_callback_cram_md5` `gssapi_server_callback_cram_md5_get` (`Gssapi *ctx`) `__attribute__((deprecated))`
- void `gssapi_server_callback_digest_md5_set` (`Gssapi *ctx`, `Gssapi_server_callback_digest_md5 cb`) `__attribute__((deprecated))`
- `Gssapi_server_callback_digest_md5` `gssapi_server_callback_digest_md5_get` (`Gssapi *ctx`) `__attribute__((deprecated))`
- void `gssapi_server_callback_external_set` (`Gssapi *ctx`, `Gssapi_server_callback_external cb`) `__attribute__((deprecated))`
- `Gssapi_server_callback_external` `gssapi_server_callback_external_get` (`Gssapi *ctx`) `__attribute__((deprecated))`
- void `gssapi_server_callback_anonymous_set` (`Gssapi *ctx`, `Gssapi_server_callback_anonymous cb`) `__attribute__((deprecated))`
- `Gssapi_server_callback_anonymous` `gssapi_server_callback_anonymous_get` (`Gssapi *ctx`) `__attribute__((deprecated))`
- void `gssapi_server_callback_realm_set` (`Gssapi *ctx`, `Gssapi_server_callback_realm cb`) `__attribute__((deprecated))`
- `Gssapi_server_callback_realm` `gssapi_server_callback_realm_get` (`Gssapi *ctx`) `__attribute__((deprecated))`
- void `gssapi_server_callback_qop_set` (`Gssapi *ctx`, `Gssapi_server_callback_qop cb`) `__attribute__((deprecated))`
- `Gssapi_server_callback_qop` `gssapi_server_callback_qop_get` (`Gssapi *ctx`) `__attribute__((deprecated))`
- void `gssapi_server_callback_maxbuf_set` (`Gssapi *ctx`, `Gssapi_server_callback_maxbuf cb`) `__attribute__((deprecated))`
- `Gssapi_server_callback_maxbuf` `gssapi_server_callback_maxbuf_get` (`Gssapi *ctx`) `__attribute__((deprecated))`
- void `gssapi_server_callback_cipher_set` (`Gssapi *ctx`, `Gssapi_server_callback_cipher cb`) `__attribute__((deprecated))`
- `Gssapi_server_callback_cipher` `gssapi_server_callback_cipher_get` (`Gssapi *ctx`) `__attribute__((deprecated))`
- void `gssapi_server_callback_securid_set` (`Gssapi *ctx`, `Gssapi_server_callback_securid cb`) `__attribute__((deprecated))`
- `Gssapi_server_callback_securid` `gssapi_server_callback_securid_get` (`Gssapi *ctx`) `__attribute__((deprecated))`
- void `gssapi_server_callback_gssapi_set` (`Gssapi *ctx`, `Gssapi_server_callback_gssapi cb`) `__attribute__((deprecated))`
- `Gssapi_server_callback_gssapi` `gssapi_server_callback_gssapi_get` (`Gssapi *ctx`) `__attribute__((deprecated))`
- void `gssapi_server_callback_service_set` (`Gssapi *ctx`, `Gssapi_server_callback_service cb`) `__attribute__((deprecated))`
- `Gssapi_server_callback_service` `gssapi_server_callback_service_get` (`Gssapi *ctx`) `__attribute__((deprecated))`

Variables

- enum { ... } [deprecated](#)

7.29.1 Define Documentation

7.29.1.1 #define [__attribute__](#)(Spec)

Definition at line 29 of file gsasl-compat.h.

7.29.2 Typedef Documentation

7.29.2.1 typedef [Gsasl_session](#) Gsasl_session_ctx [__attribute__](#)(([deprecated](#)))

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

7.29.2.2 typedef [Gsasl](#) Gsasl_ctx [__attribute__](#)(([deprecated](#)))

Definition at line 63 of file gsasl-compat.h.

7.29.2.3 typedef int(*) [Gsasl_client_callback_anonymous](#)([Gsasl_session](#) *sctx, char *out, size_t *outlen)

Definition at line 148 of file gsasl-compat.h.

7.29.2.4 typedef int(*) [Gsasl_client_callback_authentication_id](#)([Gsasl_session](#) *sctx, char *out, size_t *outlen)

Definition at line 150 of file gsasl-compat.h.

7.29.2.5 typedef int(*) [Gsasl_client_callback_authorization_id](#)([Gsasl_session](#) *sctx, char *out, size_t *outlen)

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

7.29.2.6 typedef size_t(*) [Gsasl_client_callback_maxbuf](#)([Gsasl_session](#) *sctx, size_t servermaxbuf)

Definition at line 172 of file gsasl-compat.h.

7.29.2.7 typedef int(*) [Gsasl_client_callback_passcode](#)([Gsasl_session](#) *sctx, char *out, size_t *outlen)

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

7.29.2.8 `typedef int(*) Gssapi_client_callback_password(Gssapi_session *sctx, char *out, size_t *outlen)`

Definition at line 156 of file gssapi-compat.h.

7.29.2.9 `typedef int(*) Gssapi_client_callback_pin(Gssapi_session *sctx, char *suggestion, char *out, size_t *outlen)`

Definition at line 160 of file gssapi-compat.h.

7.29.2.10 `typedef Gssapi_qop(*) Gssapi_client_callback_qop(Gssapi_session *sctx, Gssapi_qop serverqops)`

Definition at line 170 of file gssapi-compat.h.

7.29.2.11 `typedef int(*) Gssapi_client_callback_realm(Gssapi_session *sctx, char *out, size_t *outlen)`

Definition at line 174 of file gssapi-compat.h.

7.29.2.12 `typedef int(*) Gssapi_client_callback_service(Gssapi_session *sctx, char *service, size_t *servicelen, char *hostname, size_t *hostnamelen, char *servicename, size_t *servicenamelen)`

Definition at line 163 of file gssapi-compat.h.

7.29.2.13 `typedef int(*) Gssapi_server_callback_anonymous(Gssapi_session *sctx, const char *token)`

Definition at line 207 of file gssapi-compat.h.

7.29.2.14 `typedef Gssapi_cipher(*) Gssapi_server_callback_cipher(Gssapi_session *sctx)`

Definition at line 213 of file gssapi-compat.h.

7.29.2.15 `typedef int(*) Gssapi_server_callback_cram_md5(Gssapi_session *sctx, char *username, char *challenge, char *response)`

Definition at line 194 of file gssapi-compat.h.

7.29.2.16 `typedef int(*) Gssapi_server_callback_digest_md5(Gssapi_session *sctx, char *username, char *realm, char *secret)`

Definition at line 198 of file gssapi-compat.h.

7.29.2.17 `typedef int(*) Gssapi_server_callback_external(Gssapi_session *sctx)`

Definition at line 206 of file gssapi-compat.h.

7.29.2.18 `typedef int(*) Gsasl_server_callback_gssapi(Gsasl_session *sctx, const char *clientname, const char *authentication_id)`

Definition at line 185 of file gsasl-compat.h.

7.29.2.19 `typedef size_t(*) Gsasl_server_callback_maxbuf(Gsasl_session *sctx)`

Definition at line 212 of file gsasl-compat.h.

7.29.2.20 `typedef Gsasl_qop(*) Gsasl_server_callback_qop(Gsasl_session *sctx)`

Definition at line 211 of file gsasl-compat.h.

7.29.2.21 `typedef int(*) Gsasl_server_callback_realm(Gsasl_session *sctx, char *out, size_t *outlen, size_t nth)`

Definition at line 209 of file gsasl-compat.h.

7.29.2.22 `typedef int(*) Gsasl_server_callback_retrieve(Gsasl_session *sctx, const char *authentication_id, const char *authorization_id, const char *realm, char *key, size_t *keylen)`

Definition at line 176 of file gsasl-compat.h.

7.29.2.23 `typedef int(*) Gsasl_server_callback_secured(Gsasl_session *sctx, const char *authentication_id, const char *authorization_id, const char *passcode, char *pin, char *suggestpin, size_t *suggestpinlen)`

Definition at line 188 of file gsasl-compat.h.

7.29.2.24 `typedef int(*) Gsasl_server_callback_service(Gsasl_session *sctx, char *service, size_t *servicelen, char *hostname, size_t *hostnamelen)`

Definition at line 201 of file gsasl-compat.h.

7.29.2.25 `typedef int(*) Gsasl_server_callback_validate(Gsasl_session *sctx, const char *authorization_id, const char *authentication_id, const char *password)`

Definition at line 181 of file gsasl-compat.h.

7.29.3 Enumeration Type Documentation

7.29.3.1 anonymous enum

Enumerator:

GSASL_TOO_SMALL_BUFFER

GSASL_FOPEN_ERROR

GSASL_FCLOSE_ERROR
GSASL_GCRYPT_ERROR
GSASL_CANNOT_GET_CTX
GSASL_NEED_CLIENT_ANONYMOUS_CALLBACK
GSASL_NEED_CLIENT_PASSWORD_CALLBACK
GSASL_NEED_CLIENT_PASSCODE_CALLBACK
GSASL_NEED_CLIENT_PIN_CALLBACK
GSASL_NEED_CLIENT_AUTHORIZATION_ID_CALLBACK
GSASL_NEED_CLIENT_AUTHENTICATION_ID_CALLBACK
GSASL_NEED_CLIENT_SERVICE_CALLBACK
GSASL_NEED_SERVER_VALIDATE_CALLBACK
GSASL_NEED_SERVER_CRAM_MD5_CALLBACK
GSASL_NEED_SERVER_DIGEST_MD5_CALLBACK
GSASL_NEED_SERVER_EXTERNAL_CALLBACK
GSASL_NEED_SERVER_ANONYMOUS_CALLBACK
GSASL_NEED_SERVER_REALM_CALLBACK
GSASL_NEED_SERVER_SECURID_CALLBACK
GSASL_NEED_SERVER_SERVICE_CALLBACK
GSASL_NEED_SERVER_GSSAPI_CALLBACK
GSASL_NEED_SERVER_RETRIEVE_CALLBACK
GSASL_UNICODE_NORMALIZATION_ERROR
GSASL_NO_MORE_REALMS
GSASL_INVALID_HANDLE

Definition at line 34 of file gssapi.h.

7.29.4 Function Documentation

7.29.4.1 void* gssapi_appinfo_get (Gssapi_session * sctx)

gssapi_appinfo_get:

Parameters:

sctx libgssapi session handle.

Retrieve application specific data from libgssapi session handle. The application data is set using [gssapi_appinfo_set\(\)](#). It is normally used by the application to maintain state between the main program and the callback.

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

Deprecated: Use [gssapi_callback_hook_get\(\)](#) instead.

Definition at line 594 of file obsolete.c.

References Gssapi_session::application_data.

Referenced by gssapi_client_application_data_get(), and gssapi_server_application_data_get().

7.29.4.2 void gsasl_appinfo_set ([Gsasl_session](#) * *sctx*, void * *appdata*)

gsasl_appinfo_set:

Parameters:

sctx libgsasl session handle.

appdata 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_appinfo_get\(\)](#). It is normally used by the application to maintain state between the main program and the callback.

Deprecated: Use [gsasl_callback_hook_set\(\)](#) instead.

Definition at line 575 of file obsolete.c.

References [Gsasl_session::application_data](#).

Referenced by [gsasl_client_application_data_set\(\)](#), and [gsasl_server_application_data_set\(\)](#).

7.29.4.3 void* gsasl_application_data_get ([Gsasl](#) * *ctx*)

gsasl_application_data_get:

Parameters:

ctx libgsasl handle.

Retrieve application specific data from libgsasl handle. The application data is set using [gsasl_application_data_set\(\)](#). It is normally used by the application to maintain state between the main program and the callback.

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

Deprecated: Use [gsasl_callback_hook_get\(\)](#) instead.

Definition at line 556 of file obsolete.c.

References [Gsasl::application_hook](#).

7.29.4.4 void gsasl_application_data_set ([Gsasl](#) * *ctx*, void * *appdata*)

gsasl_application_data_set:

Parameters:

ctx libgsasl handle.

appdata 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_application_data_get\(\)](#). It is normally used by the application to maintain state between the main program and the callback.

Deprecated: Use [gsasl_callback_hook_set\(\)](#) instead.

Definition at line 537 of file obsolete.c.

References [Gsasl::application_hook](#).

7.29.4.5 int gssl_base64_decode (char const * src, char * target, size_t targsize)

gssl_base64_decode:

Parameters:

src input byte array
target output byte array
targsize size of output byte array

Decode Base64 data. Skips all whitespace anywhere. Converts characters, four at a time, starting at (or after) *src* from Base64 numbers into three 8 bit bytes in the target area.

Return value: Returns the number of data bytes stored at the target, or -1 on error.

Deprecated: Use [gssl_base64_from\(\)](#) instead.

Definition at line 1886 of file obsolete.c.

7.29.4.6 int gssl_base64_encode (char const * src, size_t srclength, char * target, size_t targsize)

gssl_base64_encode:

Parameters:

src input byte array
srclength size of input byte array
target output byte array
targsize size of output byte array

Encode data as base64. Converts characters, three at a time, starting at *src* into four base64 characters in the target area until the entire input buffer is encoded.

Return value: Returns the number of data bytes stored at the target, or -1 on error.

Deprecated: Use [gssl_base64_to\(\)](#) instead.

Definition at line 1807 of file obsolete.c.

References Assert.

7.29.4.7 void* gssl_client_application_data_get (Gssl_session * sctx)

gssl_client_application_data_get:

Parameters:

sctx libgssl client handle.

Retrieve application specific data from libgssl client handle. The application data is set using [gssl_client_application_data_set\(\)](#). It is normally used by the application to maintain state between the main program and the callback.

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

Deprecated: Use [gssl_callback_hook_get\(\)](#) or [gssl_session_hook_get\(\)](#) instead.

Definition at line 348 of file obsolete.c.

References [gssl_appinfo_get\(\)](#).

7.29.4.8 void `gsasl_client_application_data_set` (`Gsasl_session` * *sctx*, void * *application_data*)

`gsasl_client_application_data_set`:

Parameters:

sctx libgsasl client handle.

application_data opaque pointer to application specific data.

Store application specific data in the libgsasl client handle. The application data can be later (for instance, inside a callback) be retrieved by calling `gsasl_client_application_data_get()`. It is normally used by the application to maintain state between the main program and the callback.

Deprecated: Use `gsasl_callback_hook_set()` or `gsasl_session_hook_set()` instead.

Definition at line 327 of file `obsolete.c`.

References `gsasl_appinfo_set()`.

7.29.4.9 `Gsasl_client_callback_anonymous` `gsasl_client_callback_anonymous_get` (`Gsasl` * *ctx*)

`gsasl_client_callback_anonymous_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling `gsasl_client_callback_anonymous_set()`.

Deprecated: This function is part of the old callback interface. The new interface uses `gsasl_callback_set()` to set the application callback, and uses `gsasl_callback()` or `gsasl_property_get()` to invoke the callback for certain properties.

Definition at line 895 of file `obsolete.c`.

References `Gsasl::cbc_anonymous`.

Referenced by `gsasl_property_get()`.

7.29.4.10 void `gsasl_client_callback_anonymous_set` (`Gsasl` * *ctx*, `Gsasl_client_callback_anonymous_cb`)

`gsasl_client_callback_anonymous_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the client to set the anonymous token, which usually is the users email address. The function can be later retrieved using `gsasl_client_callback_anonymous_get()`.

Deprecated: This function is part of the old callback interface. The new interface uses `gsasl_callback_set()` to set the application callback, and uses `gsasl_callback()` or `gsasl_property_get()` to invoke the callback for certain properties.

Definition at line 876 of file `obsolete.c`.

References `Gsasl::cbc_anonymous`.

7.29.4.11 [Gssapi_client_callback_authentication_id](#) `gssapi_client_callback_authentication_id_get` ([Gssapi](#) * *ctx*)

`gssapi_client_callback_authentication_id_get`:

Parameters:

ctx libgssapi handle.

Return value: Returns the callback earlier set by calling [gssapi_client_callback_authentication_id_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gssapi_callback_set\(\)](#) to set the application callback, and uses [gssapi_callback\(\)](#) or [gssapi_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 652 of file `obsolete.c`.

References `Gssapi::cbc_authentication_id`.

Referenced by `_gssapi_kerberos_v5_client_step()`, and `gssapi_property_get()`.

7.29.4.12 `void gssapi_client_callback_authentication_id_set` ([Gssapi](#) * *ctx*, [Gssapi_client_callback_authentication_id](#) *cb*)

`gssapi_client_callback_authentication_id_set`:

Parameters:

ctx libgssapi handle.

cb callback function

Specify the callback function to use in the client to set the authentication identity. The function can be later retrieved using [gssapi_client_callback_authentication_id_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gssapi_callback_set\(\)](#) to set the application callback, and uses [gssapi_callback\(\)](#) or [gssapi_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 632 of file `obsolete.c`.

References `Gssapi::cbc_authentication_id`.

7.29.4.13 [Gssapi_client_callback_authorization_id](#) `gssapi_client_callback_authorization_id_get` ([Gssapi](#) * *ctx*)

`gssapi_client_callback_authorization_id_get`:

Parameters:

ctx libgssapi handle.

Return value: Returns the callback earlier set by calling [gssapi_client_callback_authorization_id_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gssapi_callback_set\(\)](#) to set the application callback, and uses [gssapi_callback\(\)](#) or [gssapi_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 692 of file `obsolete.c`.

References `Gsasl::cbc_authorization_id`.

Referenced by `_gsasl_kerberos_v5_client_step()`, and `gsasl_property_get()`.

7.29.4.14 `void gsasl_client_callback_authorization_id_set (Gsasl * ctx, Gsasl_client_callback_authorization_id cb)`

`gsasl_client_callback_authorization_id_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the client to set the authorization identity. The function can be later retrieved using `gsasl_client_callback_authorization_id_get()`.

Deprecated: This function is part of the old callback interface. The new interface uses `gsasl_callback_set()` to set the application callback, and uses `gsasl_callback()` or `gsasl_property_get()` to invoke the callback for certain properties.

Definition at line 672 of file `obsolete.c`.

References `Gsasl::cbc_authorization_id`.

7.29.4.15 `Gsasl_client_callback_maxbuf gsasl_client_callback_maxbuf_get (Gsasl * ctx)`

`gsasl_client_callback_maxbuf_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling `gsasl_client_callback_maxbuf_set()`.

Deprecated: This function is part of the old callback interface. The new interface uses `gsasl_callback_set()` to set the application callback, and uses `gsasl_callback()` or `gsasl_property_get()` to invoke the callback for certain properties.

Definition at line 975 of file `obsolete.c`.

References `Gsasl::cbc_maxbuf`.

Referenced by `_gsasl_kerberos_v5_client_step()`.

7.29.4.16 `void gsasl_client_callback_maxbuf_set (Gsasl * ctx, Gsasl_client_callback_maxbuf cb)`

`gsasl_client_callback_maxbuf_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the client to inform the server of the largest buffer the client is able to receive when using the DIGEST-MD5 "auth-int" or "auth-conf" Quality of Protection (qop). If this

directive is missing, the default value 65536 will be assumed. The function can be later retrieved using [gssapi_client_callback_maxbuf_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gssapi_callback_set\(\)](#) to set the application callback, and uses [gssapi_callback\(\)](#) or [gssapi_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 956 of file `obsolete.c`.

References `Gssapi::cbc_maxbuf`.

7.29.4.17 [Gssapi_client_callback_passcode](#) [gssapi_client_callback_passcode_get](#) (`Gssapi * ctx`)

`gssapi_client_callback_passcode_get`:

Parameters:

ctx libgssapi handle.

Return value: Returns the callback earlier set by calling [gssapi_client_callback_passcode_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gssapi_callback_set\(\)](#) to set the application callback, and uses [gssapi_callback\(\)](#) or [gssapi_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 772 of file `obsolete.c`.

References `Gssapi::cbc_passcode`.

Referenced by [gssapi_property_get\(\)](#).

7.29.4.18 `void gssapi_client_callback_passcode_set` (`Gssapi * ctx`, [Gssapi_client_callback_passcode cb](#))

`gssapi_client_callback_passcode_set`:

Parameters:

ctx libgssapi handle.

cb callback function

Specify the callback function to use in the client to set the passcode. The function can be later retrieved using [gssapi_client_callback_passcode_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gssapi_callback_set\(\)](#) to set the application callback, and uses [gssapi_callback\(\)](#) or [gssapi_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 752 of file `obsolete.c`.

References `Gssapi::cbc_passcode`.

7.29.4.19 [Gssapi_client_callback_password](#) [gssapi_client_callback_password_get](#) (`Gssapi * ctx`)

`gssapi_client_callback_password_get`:

Parameters:

ctx libgssapi handle.

Return value: Returns the callback earlier set by calling [gsasl_client_callback_password_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 732 of file `obsolete.c`.

References `Gsasl::cbc_password`.

Referenced by `_gsasl_kerberos_v5_client_step()`, and `gsasl_property_get()`.

7.29.4.20 `void gsasl_client_callback_password_set (Gsasl * ctx, Gsasl_client_callback_password cb)`

`gsasl_client_callback_password_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the client to set the password. The function can be later retrieved using [gsasl_client_callback_password_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 712 of file `obsolete.c`.

References `Gsasl::cbc_password`.

7.29.4.21 `Gsasl_client_callback_pin gsasl_client_callback_pin_get (Gsasl * ctx)`

`gsasl_client_callback_pin_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_client_callback_pin_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 813 of file `obsolete.c`.

References `Gsasl::cbc_pin`.

Referenced by `gsasl_property_get()`.

7.29.4.22 `void gsasl_client_callback_pin_set (Gsasl * ctx, Gsasl_client_callback_pin cb)`

`gsasl_client_callback_pin_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the client to choose a new pin, possibly suggested by the server, for the SECURID mechanism. This is not normally invoked, but only when the server requests it. The function can be later retrieved using [gssapi_client_callback_pin_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gssapi_callback_set\(\)](#) to set the application callback, and uses [gssapi_callback\(\)](#) or [gssapi_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 794 of file `obsolete.c`.

References `Gssapi::cbc_pin`.

7.29.4.23 [Gssapi_client_callback_qop](#) [gssapi_client_callback_qop_get](#) (`Gssapi * ctx`)

`gssapi_client_callback_qop_get`:

Parameters:

ctx libgssapi handle.

Return value: Returns the callback earlier set by calling [gssapi_client_callback_qop_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gssapi_callback_set\(\)](#) to set the application callback, and uses [gssapi_callback\(\)](#) or [gssapi_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 933 of file `obsolete.c`.

References `Gssapi::cbc_qop`.

Referenced by `_gssapi_kerberos_v5_client_step()`.

7.29.4.24 `void gssapi_client_callback_qop_set` (`Gssapi * ctx`, [Gssapi_client_callback_qop cb](#))

`gssapi_client_callback_qop_set`:

Parameters:

ctx libgssapi handle.

cb callback function

Specify the callback function to use in the client to determine the qop to use after looking at what the server offered. The function can be later retrieved using [gssapi_client_callback_qop_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gssapi_callback_set\(\)](#) to set the application callback, and uses [gssapi_callback\(\)](#) or [gssapi_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 915 of file `obsolete.c`.

References `Gssapi::cbc_qop`.

7.29.4.25 [Gssapi_client_callback_realm](#) [gssapi_client_callback_realm_get](#) (`Gssapi * ctx`)

`gssapi_client_callback_realm_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_client_callback_realm_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1014 of file `obsolete.c`.

References `Gsasl::cbc_realm`.

Referenced by [_gsasl_kerberos_v5_client_step\(\)](#), and [gsasl_property_get\(\)](#).

7.29.4.26 void gsasl_client_callback_realm_set (Gsasl * ctx, Gsasl_client_callback_realm cb)

`gsasl_client_callback_realm_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the client to know which realm it belongs to. The realm is used by the server to determine which username and password to use. The function can be later retrieved using [gsasl_client_callback_realm_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 996 of file `obsolete.c`.

References `Gsasl::cbc_realm`.

7.29.4.27 Gsasl_client_callback_service gsasl_client_callback_service_get (Gsasl * ctx)

`gsasl_client_callback_service_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_client_callback_service_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 855 of file `obsolete.c`.

References `Gsasl::cbc_service`.

Referenced by [_gsasl_kerberos_v5_client_step\(\)](#), and [gsasl_property_get\(\)](#).

7.29.4.28 void gsasl_client_callback_service_set (Gsasl * ctx, Gsasl_client_callback_service cb)

`gsasl_client_callback_service_set`:

Parameters:

ctx libgsasl handle.
cb callback function

Specify the callback function to use in the client to set the name of the service. The service buffer should be a registered GSSAPI host-based service name, hostname the name of the server. Servicename is used by DIGEST-MD5 and should be the name of generic server in case of a replicated service. The function can be later retrieved using [gsasl_client_callback_service_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 836 of file obsolete.c.

References Gsasl::cbc_service.

7.29.4.29 Gsasl* gsasl_client_ctx_get (Gsasl_session * sctx)

gsasl_client_ctx_get:

Parameters:

sctx libgsasl client handle

Return value: Returns the libgsasl handle given a libgsasl client handle.

Deprecated: This function is not useful with the new 0.2.0 API.

Definition at line 307 of file obsolete.c.

References Gsasl_session::ctx.

Referenced by [_gsasl_kerberos_v5_client_step\(\)](#).

7.29.4.30 void gsasl_client_finish (Gsasl_session * sctx)

gsasl_client_finish:

Parameters:

sctx libgsasl client handle.

Destroy a libgsasl client handle. The handle must not be used with other libgsasl functions after this call.

Deprecated: Use [gsasl_finish\(\)](#) instead.

Definition at line 278 of file obsolete.c.

References [gsasl_finish\(\)](#).

7.29.4.31 int gsasl_client_listmech (Gsasl * ctx, char * out, size_t * outlen)

gsasl_client_listmech:

Parameters:

ctx libgsasl handle.

out output character array.

outlen input maximum size of output character array, on output contains actual length of output array.

Write SASL names, separated by space, of mechanisms supported by the libgsasl client to the output array. To find out how large the output array must be, call this function with `out=NULL`.

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

Deprecated: Use `gsasl_client_mechlist()` instead.

Definition at line 41 of file `obsolete.c`.

References `gsasl_client_mechlist()`, `GSASL_OK`, and `GSASL_TOO_SMALL_BUFFER`.

7.29.4.32 `int gsasl_client_step (Gsasl_session * sctx, const char * input, size_t input_len, char * output, size_t * output_len)`

`gsasl_client_step`:

Parameters:

sctx libgsasl client handle.

input input byte array.

input_len size of input byte array.

output output byte array.

output_len size of output byte array.

Perform one step of SASL authentication in client. This reads data from server (specified with `input` and `input_len`), processes it (potentially invoking callbacks to the application), and writes data to server (into variables `output` and `output_len`).

The contents of the output buffer is unspecified if this functions returns anything other than `GSASL_NEEDS_MORE`.

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

Deprecated: Use `gsasl_step()` instead.

Definition at line 161 of file `obsolete.c`.

7.29.4.33 `int gsasl_client_step_base64 (Gsasl_session * sctx, const char * b64input, char * b64output, size_t b64output_len)`

`gsasl_client_step_base64`:

Parameters:

sctx libgsasl client handle.

b64input input base64 encoded byte array.

b64output output base64 encoded byte array.

b64output_len size of output base64 encoded byte array.

This is a simple wrapper around `gsasl_client_step()` that base64 decodes the input and base64 encodes the output.

Return value: See [gssapi_client_step\(\)](#).

Deprecated: Use [gssapi_step64\(\)](#) instead.

Definition at line 239 of file `obsolete.c`.

7.29.4.34 **Gssapi*** `gssapi_ctx_get (Gssapi_session * sctx)`

`gssapi_ctx_get`:

Parameters:

sctx libgssapi session handle

Return value: Returns the libgssapi handle given a libgssapi session handle.

Deprecated: This function is not useful with the new 0.2.0 API.

Definition at line 438 of file `obsolete.c`.

References `Gssapi_session::ctx`.

7.29.4.35 **int** `gssapi_decode_inline (Gssapi_session * sctx, const char * input, size_t input_len, char * output, size_t * output_len)`

`gssapi_decode_inline`:

Parameters:

sctx libgssapi session handle.

input input byte array.

input_len size of input byte array.

output output byte array.

output_len size of output byte array.

Decode data according to negotiated SASL mechanism. This might mean that data is integrity or privacy protected.

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

Deprecated: Use [gssapi_decode\(\)](#) instead.

Since: 0.2.0

Definition at line 502 of file `obsolete.c`.

References `gssapi_decode()`, `GSASL_OK`, and `GSASL_TOO_SMALL_BUFFER`.

7.29.4.36 **int** `gssapi_encode_inline (Gssapi_session * sctx, const char * input, size_t input_len, char * output, size_t * output_len)`

`gssapi_encode_inline`:

Parameters:

sctx libgssapi session handle.

input input byte array.

input_len size of input byte array.

output output byte array.

output_len size of output byte array.

Encode data according to negotiated SASL mechanism. This might mean that data is integrity or privacy protected.

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

Deprecated: Use [gsasl_encode\(\)](#) instead.

Since: 0.2.0

Definition at line 462 of file obsolete.c.

References [gsasl_encode\(\)](#), [GSASL_OK](#), and [GSASL_TOO_SMALL_BUFFER](#).

7.29.4.37 `int gsasl_md5pwd_get_password (const char * filename, const char * username, char * key, size_t * keylen)`

`gsasl_md5pwd_get_password`:

Parameters:

filename filename of file containing passwords.

username username string.

key output character array.

keylen input maximum size of output character array, on output contains actual length of output array.

Retrieve password for user from specified file. To find out how large the output array must be, call this function with `out=NULL`.

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.

Deprecated: Use [gsasl_simple_getpass\(\)](#) instead.

Definition at line 1685 of file obsolete.c.

References [GSASL_AUTHENTICATION_ERROR](#), [GSASL_FCLOSE_ERROR](#), [GSASL_FOPEN_ERROR](#), [GSASL_OK](#), and [GSASL_TOO_SMALL_BUFFER](#).

7.29.4.38 `int gsasl_randomize (int strong, char * data, size_t datalen)`

`gsasl_randomize`:

Parameters:

strong 0 iff operation should not block, non-0 for very strong randomness.

data output array to be filled with random data.

datalen size of output array.

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

Return value: Returns GSASL_OK iff successful.

Deprecated: Use [gssapi_random\(\)](#) or [gssapi_nonce\(\)](#) instead.

Definition at line 422 of file obsolete.c.

References [gssapi_nonce\(\)](#), and [gssapi_random\(\)](#).

7.29.4.39 void* gssapi_server_application_data_get (Gssapi_session * *sctx*)

[gssapi_server_application_data_get](#):

Parameters:

sctx libgssapi server handle.

Retrieve application specific data from libgssapi server handle. The application data is set using [gssapi_server_application_data_set\(\)](#). It is normally used by the application to maintain state between the main program and the callback.

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

Deprecated: Use [gssapi_callback_hook_get\(\)](#) or [gssapi_session_hook_get\(\)](#) instead.

Definition at line 403 of file obsolete.c.

References [gssapi_appinfo_get\(\)](#).

7.29.4.40 void gssapi_server_application_data_set (Gssapi_session * *sctx*, void * *application_data*)

[gssapi_server_application_data_set](#):

Parameters:

sctx libgssapi server handle.

application_data opaque pointer to application specific data.

Store application specific data in the libgssapi server handle. The application data can be later (for instance, inside a callback) be retrieved by calling [gssapi_server_application_data_get\(\)](#). It is normally used by the application to maintain state between the main program and the callback.

Deprecated: Use [gssapi_callback_hook_set\(\)](#) or [gssapi_session_hook_set\(\)](#) instead.

Definition at line 382 of file obsolete.c.

References [gssapi_appinfo_set\(\)](#).

7.29.4.41 Gssapi_server_callback_anonymous gssapi_server_callback_anonymous_get (Gssapi * *ctx*)

[gssapi_server_callback_anonymous_get](#):

Parameters:

ctx libgssapi handle.

Return value: Returns the callback earlier set by calling [gssapi_server_callback_anonymous_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1252 of file `obsolete.c`.

References `Gsasl::cbs_anonymous`.

Referenced by `gsasl_callback()`.

7.29.4.42 `void gsasl_server_callback_anonymous_set (Gsasl * ctx, Gsasl_server_callback_anonymous cb)`

`gsasl_server_callback_anonymous_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the server for deciding if user is permitted anonymous access. The function can be later retrieved using [gsasl_server_callback_anonymous_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1233 of file `obsolete.c`.

References `Gsasl::cbs_anonymous`.

7.29.4.43 `Gsasl_server_callback_cipher gsasl_server_callback_cipher_get (Gsasl * ctx)`

`gsasl_server_callback_cipher_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_server_callback_cipher_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1414 of file `obsolete.c`.

References `Gsasl::cbs_cipher`.

7.29.4.44 `void gsasl_server_callback_cipher_set (Gsasl * ctx, Gsasl_server_callback_cipher cb)`

`gsasl_server_callback_cipher_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the server to inform the client of the cipher suites supported. The DES and 3DES ciphers must be supported for interoperability. It is currently used by the DIGEST-MD5 mechanism. The function can be later retrieved using [gssapi_server_callback_cipher_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gssapi_callback_set\(\)](#) to set the application callback, and uses [gssapi_callback\(\)](#) or [gssapi_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1395 of file `obsolete.c`.

References `Gssapi::cbs_cipher`.

7.29.4.45 [Gssapi_server_callback_cram_md5](#) `gssapi_server_callback_cram_md5_get (Gssapi * ctx)`

`gssapi_server_callback_cram_md5_get`:

Parameters:

ctx libgssapi handle.

Return value: Returns the callback earlier set by calling [gssapi_server_callback_cram_md5_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gssapi_callback_set\(\)](#) to set the application callback, and uses [gssapi_callback\(\)](#) or [gssapi_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1134 of file `obsolete.c`.

References `Gssapi::cbs_cram_md5`.

7.29.4.46 `void gssapi_server_callback_cram_md5_set (Gssapi * ctx, Gssapi_server_callback_cram_md5 cb)`

`gssapi_server_callback_cram_md5_set`:

Parameters:

ctx libgssapi handle.

cb callback function

Specify the callback function to use in the server for deciding if user is authenticated using CRAM-MD5 challenge and response. The function can be later retrieved using [gssapi_server_callback_cram_md5_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gssapi_callback_set\(\)](#) to set the application callback, and uses [gssapi_callback\(\)](#) or [gssapi_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1115 of file `obsolete.c`.

References `Gssapi::cbs_cram_md5`.

7.29.4.47 [Gssapi_server_callback_digest_md5](#) `gssapi_server_callback_digest_md5_get (Gssapi * ctx)`

`gssapi_server_callback_digest_md5_get`:

Parameters:

ctx libgssapi handle.

Return value: Return the callback earlier set by calling [gsasl_server_callback_digest_md5_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1174 of file `obsolete.c`.

References `Gsasl::cbs_digest_md5`.

7.29.4.48 `void gsasl_server_callback_digest_md5_set (Gsasl * ctx, Gsasl_server_callback_digest_md5 cb)`

`gsasl_server_callback_digest_md5_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the server for retrieving the secret hash of the username, realm and password for use in the DIGEST-MD5 mechanism. The function can be later retrieved using [gsasl_server_callback_digest_md5_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1155 of file `obsolete.c`.

References `Gsasl::cbs_digest_md5`.

7.29.4.49 `Gsasl_server_callback_external gsasl_server_callback_external_get (Gsasl * ctx)`

`gsasl_server_callback_external_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_server_callback_external_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1213 of file `obsolete.c`.

References `Gsasl::cbs_external`.

Referenced by [gsasl_callback\(\)](#).

7.29.4.50 `void gsasl_server_callback_external_set (Gsasl * ctx, Gsasl_server_callback_external cb)`

`gsasl_server_callback_external_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the server for deciding if user is authenticated out of band. The function can be later retrieved using [gssapi_server_callback_external_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gssapi_callback_set\(\)](#) to set the application callback, and uses [gssapi_callback\(\)](#) or [gssapi_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1194 of file `obsolete.c`.

References `Gssapi::cbs_external`.

7.29.4.51 [Gssapi_server_callback_gssapi](#) `gssapi_server_callback_gssapi_get (Gssapi * ctx)`

`gssapi_server_callback_gssapi_get`:

Parameters:

ctx libgssapi handle.

Return value: Returns the callback earlier set by calling [gssapi_server_callback_gssapi_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gssapi_callback_set\(\)](#) to set the application callback, and uses [gssapi_callback\(\)](#) or [gssapi_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1502 of file `obsolete.c`.

References `Gssapi::cbs_gssapi`.

Referenced by `gssapi_callback()`.

7.29.4.52 `void gssapi_server_callback_gssapi_set (Gssapi * ctx, Gssapi_server_callback_gssapi cb)`

`gssapi_server_callback_gssapi_set`:

Parameters:

ctx libgssapi handle.

cb callback function

Specify the callback function to use in the server for checking if a GSSAPI user is authorized for username (by, e.g., calling `krb5_userok()`). The function should return `GSASL_OK` if the user should be permitted access, or an error code such as `GSASL_AUTHENTICATION_ERROR` on failure. The function can be later retrieved using [gssapi_server_callback_gssapi_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gssapi_callback_set\(\)](#) to set the application callback, and uses [gssapi_callback\(\)](#) or [gssapi_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1483 of file `obsolete.c`.

References `Gssapi::cbs_gssapi`.

7.29.4.53 [Gssapi_server_callback_maxbuf](#) `gssapi_server_callback_maxbuf_get (Gssapi * ctx)`

`gssapi_server_callback_maxbuf_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_server_callback_maxbuf_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1373 of file `obsolete.c`.

References `Gsasl::cbs_maxbuf`.

Referenced by `_gsasl_kerberos_v5_server_step()`.

7.29.4.54 void [gsasl_server_callback_maxbuf_set](#) ([Gsasl](#) * *ctx*, [Gsasl_server_callback_maxbuf](#) *cb*)

`gsasl_server_callback_maxbuf_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the server to inform the client of the largest buffer the server is able to receive when using the DIGEST-MD5 "auth-int" or "auth-conf" Quality of Protection (qop). If this directive is missing, the default value 65536 will be assumed. The function can be later retrieved using [gsasl_server_callback_maxbuf_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1354 of file `obsolete.c`.

References `Gsasl::cbs_maxbuf`.

7.29.4.55 [Gsasl_server_callback_qop](#) [gsasl_server_callback_qop_get](#) ([Gsasl](#) * *ctx*)

`gsasl_server_callback_qop_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_server_callback_qop_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1331 of file `obsolete.c`.

References `Gsasl::cbs_qop`.

Referenced by `_gsasl_kerberos_v5_server_step()`.

7.29.4.56 void gssapi_server_callback_qop_set (Gssapi * ctx, Gssapi_server_callback_qop cb)

gssapi_server_callback_qop_set:

Parameters:

ctx libgssapi handle.

cb callback function

Specify the callback function to use in the server to know which quality of protection it accepts. The quality of protection eventually used is selected by the client though. It is currently used by the DIGEST-MD5 mechanism. The function can be later retrieved using [gssapi_server_callback_qop_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gssapi_callback_set\(\)](#) to set the application callback, and uses [gssapi_callback\(\)](#) or [gssapi_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1313 of file obsolete.c.

References Gssapi::cbs_qop.

7.29.4.57 Gssapi_server_callback_realm gssapi_server_callback_realm_get (Gssapi * ctx)

gssapi_server_callback_realm_get:

Parameters:

ctx libgssapi handle.

Return value: Returns the callback earlier set by calling [gssapi_server_callback_realm_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gssapi_callback_set\(\)](#) to set the application callback, and uses [gssapi_callback\(\)](#) or [gssapi_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1291 of file obsolete.c.

References Gssapi::cbs_realm.

Referenced by [_gssapi_kerberos_v5_server_step\(\)](#).

7.29.4.58 void gssapi_server_callback_realm_set (Gssapi * ctx, Gssapi_server_callback_realm cb)

gssapi_server_callback_realm_set:

Parameters:

ctx libgssapi handle.

cb callback function

Specify the callback function to use in the server to know which realm it serves. The realm is used by the user to determine which username and password to use. The function can be later retrieved using [gssapi_server_callback_realm_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gssapi_callback_set\(\)](#) to set the application callback, and uses [gssapi_callback\(\)](#) or [gssapi_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1273 of file obsolete.c.

References Gsasl::cbs_realm.

7.29.4.59 [Gsasl_server_callback_retrieve](#) `gsasl_server_callback_retrieve_get (Gsasl * ctx)`

`gsasl_server_callback_retrieve_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_server_callback_retrieve_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1094 of file obsolete.c.

References Gsasl::cbs_retrieve.

Referenced by `_gsasl_kerberos_v5_server_step()`, and `gsasl_callback()`.

7.29.4.60 `void gsasl_server_callback_retrieve_set (Gsasl * ctx, Gsasl_server_callback_retrieve cb)`

`gsasl_server_callback_retrieve_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the server for deciding if user is authenticated using authentication identity, authorization identity and password. The function can be later retrieved using [gsasl_server_callback_retrieve_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1075 of file obsolete.c.

References Gsasl::cbs_retrieve.

7.29.4.61 [Gsasl_server_callback_securid](#) `gsasl_server_callback_securid_get (Gsasl * ctx)`

`gsasl_server_callback_securid_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_server_callback_securid_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1460 of file obsolete.c.

References Gssapi::cbs_secuid.

Referenced by gssapi_callback().

7.29.4.62 void gssapi_server_callback_secuid_set (Gssapi * ctx, Gssapi_server_callback_secuid cb)

gssapi_server_callback_secuid_set:

Parameters:

ctx libgssapi handle.

cb callback function

Specify the callback function to use in the server for validating a user via the SECURID mechanism. The function should return GSASL_OK if user authenticated successfully, GSASL_SECURID_SERVER_NEED_ADDITIONAL_PASSCODE if it wants another passcode, GSASL_SECURID_SERVER_NEED_NEW_PIN if it wants a PIN change, or an error. When (and only when) GSASL_SECURID_SERVER_NEED_NEW_PIN is returned, suggestpin can be populated with a PIN code the server suggests, and suggestpinlen set to the length of the PIN. The function can be later retrieved using [gssapi_server_callback_secuid_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gssapi_callback_set\(\)](#) to set the application callback, and uses [gssapi_callback\(\)](#) or [gssapi_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1441 of file obsolete.c.

References Gssapi::cbs_secuid.

7.29.4.63 Gssapi_server_callback_service gssapi_server_callback_service_get (Gssapi * ctx)

gssapi_server_callback_service_get:

Parameters:

ctx libgssapi handle.

Return value: Returns the callback earlier set by calling [gssapi_server_callback_service_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gssapi_callback_set\(\)](#) to set the application callback, and uses [gssapi_callback\(\)](#) or [gssapi_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1543 of file obsolete.c.

References Gssapi::cbs_service.

Referenced by [_gssapi_kerberos_v5_server_step\(\)](#).

7.29.4.64 void gssapi_server_callback_service_set (Gssapi * ctx, Gssapi_server_callback_service cb)

gssapi_server_callback_service_set:

Parameters:

ctx libgssapi handle.

cb callback function

Specify the callback function to use in the server to set the name of the service. The service buffer should be a registered GSSAPI host-based service name, hostname the name of the server. The function can be later retrieved using [gsasl_server_callback_service_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1524 of file `obsolete.c`.

References `Gsasl::cbs_service`.

7.29.4.65 [Gsasl_server_callback_validate](#) [gsasl_server_callback_validate_get](#) (`Gsasl * ctx`)

`gsasl_server_callback_validate_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_server_callback_validate_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1054 of file `obsolete.c`.

References `Gsasl::cbs_validate`.

Referenced by [gsasl_callback\(\)](#).

7.29.4.66 `void gsasl_server_callback_validate_set` (`Gsasl * ctx`, [Gsasl_server_callback_validate](#) *cb*)

`gsasl_server_callback_validate_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the server for deciding if user is authenticated using authentication identity, authorization identity and password. The function can be later retrieved using [gsasl_server_callback_validate_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1035 of file `obsolete.c`.

References `Gsasl::cbs_validate`.

7.29.4.67 `Gsasl*` [gsasl_server_ctx_get](#) (`Gsasl_session * sctx`)

`gsasl_server_ctx_get`:

Parameters:

sctx libgssapi server handle

Return value: Returns the libgssapi handle given a libgssapi server handle.

Deprecated: This function is not useful with the new 0.2.0 API.

Definition at line 362 of file obsolete.c.

References Gssapi_session::ctx.

Referenced by _gssapi_kerberos_v5_server_step().

7.29.4.68 void gssapi_server_finish (Gssapi_session * sctx)

gssapi_server_finish:

Parameters:

sctx libgssapi server handle.

Destroy a libgssapi server handle. The handle must not be used with other libgssapi functions after this call.

Deprecated: Use [gssapi_finish\(\)](#) instead.

Definition at line 293 of file obsolete.c.

References gssapi_finish().

7.29.4.69 int gssapi_server_listmech (Gssapi * ctx, char * out, size_t * outlen)

gssapi_server_listmech:

Parameters:

ctx libgssapi handle.

out output character array.

outlen input maximum size of output character array, on output contains actual length of output array.

Write SASL names, separated by space, of mechanisms supported by the libgssapi server to the output array. To find out how large the output array must be, call this function with out=NULL.

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

Deprecated: Use [gssapi_server_mechlist\(\)](#) instead.

Definition at line 83 of file obsolete.c.

References GSASL_OK, gssapi_server_mechlist(), and GSASL_TOO_SMALL_BUFFER.

7.29.4.70 int gssapi_server_step (Gssapi_session * sctx, const char * input, size_t input_len, char * output, size_t * output_len)

gssapi_server_step:

Parameters:

sctx libgssapi server handle.

input input byte array.
input_len size of input byte array.
output output byte array.
output_len size of output byte array.

Perform one step of SASL authentication in server. This reads data from client (specified with *input* and *input_len*), processes it (potentially invoking callbacks to the application), and writes data to client (into variables *output* and *output_len*).

The contents of the output buffer is unspecified if this functions returns anything other than GSASL_NEEDS_MORE.

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

Deprecated: Use [gsasl_step\(\)](#) instead.

Definition at line 191 of file obsolete.c.

7.29.4.71 `int gsasl_server_step_base64 (Gsasl_session * sctx, const char * b64input, char * b64output, size_t b64output_len)`

`gsasl_server_step_base64`:

Parameters:

sctx libgsasl server handle.
b64input input base64 encoded byte array.
b64output output base64 encoded byte array.
b64output_len size of output base64 encoded byte array.

This is a simple wrapper around [gsasl_server_step\(\)](#) that base64 decodes the input and base64 encodes the output.

Return value: See [gsasl_server_step\(\)](#).

Deprecated: Use [gsasl_step64\(\)](#) instead.

Definition at line 261 of file obsolete.c.

7.29.4.72 `const char* gsasl_server_suggest_mechanism (Gsasl * ctx, const char * meclist)`

`gsasl_server_suggest_mechanism`:

Parameters:

ctx libgsasl handle.
meclist input character array with SASL mechanism names, separated by invalid characters (e.g. SPC).

Return value: Returns name of "best" SASL mechanism supported by the libgsasl server which is present in the input string.

Deprecated: This function was never useful, since it is the client that chose which mechanism to use.

Definition at line 612 of file obsolete.c.

7.29.4.73 char* gsasl_stringprep_nfkc (const char * *in*, ssize_t *len*)

gsasl_stringprep_nfkc:

Parameters:

in a UTF-8 encoded string.

len length of *in*, in bytes, or -1 if *in* is nul-terminated.

Converts a string into canonical form, standardizing such issues as whether a character with an accent is represented as a base character and combining accent or as a single precomposed character.

The normalization mode is NFKC (ALL COMPOSE). It standardizes differences that do not affect the text content, such as the above-mentioned accent representation. It standardizes the "compatibility" characters in Unicode, such as SUPERSCRIPT THREE to the standard forms (in this case DIGIT THREE). Formatting information may be lost but for most text operations such characters should be considered the same. It returns a result with composed forms rather than a maximally decomposed form.

Return value: Return a newly allocated string, that is the NFKC normalized form of *in*, or NULL on error.

Deprecated: No replacement functionality in GNU SASL, use GNU Libidn instead. Note that in SASL, you most likely want to use SASLprep and not bare NFKC, see [gsasl_saslprep\(\)](#).

Definition at line 1579 of file obsolete.c.

7.29.4.74 char* gsasl_stringprep_saslprep (const char * *in*, int * *stringprep_rc*)

gsasl_stringprep_saslprep:

Parameters:

in input ASCII or UTF-8 string with data to prepare according to SASLprep.

stringprep_rc pointer to output variable with stringprep error code, or NULL to indicate that you don't care about it.

Process a Unicode string for comparison, according to the "SASLprep" stringprep profile. This function is intended to be used by Simple Authentication and Security Layer (SASL) mechanisms (such as PLAIN, CRAM-MD5, and DIGEST-MD5) as well as other protocols exchanging user names and/or passwords.

Return value: Return a newly allocated string that is the "SASLprep" processed form of the input string, or NULL on error, in which case contain the stringprep library error code.

Deprecated: Use [gsasl_saslprep\(\)](#) instead.

Definition at line 1610 of file obsolete.c.

7.29.4.75 char* gsasl_stringprep_trace (const char * *in*, int * *stringprep_rc*)

gsasl_stringprep_trace:

Parameters:

in input ASCII or UTF-8 string with data to prepare according to "trace".

stringprep_rc pointer to output variable with stringprep error code, or NULL to indicate that you don't care about it.

Process a Unicode string for use as trace information, according to the "trace" stringprep profile. The profile is designed for use with the SASL ANONYMOUS Mechanism.

Return value: Return a newly allocated string that is the "trace" processed form of the input string, or NULL on error, in which case contain the stringprep library error code.

Deprecated: No replacement functionality in GNU SASL, use GNU Libidn instead.

Definition at line 1644 of file obsolete.c.

7.29.5 Variable Documentation

7.29.5.1 `enum { ... }` [deprecated](#)

7.30 gssl-mech.h File Reference

Data Structures

- struct [Gssl_mechanism_functions](#)
- struct [Gssl_mechanism](#)

Typedefs

- typedef int(*) [Gssl_init_function](#) ([Gssl](#) *ctx)
- typedef void(*) [Gssl_done_function](#) ([Gssl](#) *ctx)
- typedef int(*) [Gssl_start_function](#) ([Gssl_session](#) *sctx, void **mech_data)
- typedef int(*) [Gssl_step_function](#) ([Gssl_session](#) *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)
- typedef void(*) [Gssl_finish_function](#) ([Gssl_session](#) *sctx, void *mech_data)
- typedef int(*) [Gssl_code_function](#) ([Gssl_session](#) *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)
- typedef [Gssl_mechanism_functions](#) [Gssl_mechanism_functions](#)
- typedef [Gssl_mechanism](#) [Gssl_mechanism](#)

Functions

- int [gssl_register](#) ([Gssl](#) *ctx, const [Gssl_mechanism](#) *mech)

7.30.1 Typedef Documentation

7.30.1.1 typedef int(*) [Gssl_code_function](#)([Gssl_session](#) *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)

Definition at line 34 of file gssl-mech.h.

7.30.1.2 typedef void(*) [Gssl_done_function](#)([Gssl](#) *ctx)

Definition at line 28 of file gssl-mech.h.

7.30.1.3 typedef void(*) [Gssl_finish_function](#)([Gssl_session](#) *sctx, void *mech_data)

Definition at line 33 of file gssl-mech.h.

7.30.1.4 typedef int(*) [Gssl_init_function](#)([Gssl](#) *ctx)

Definition at line 27 of file gssl-mech.h.

7.30.1.5 typedef struct [Gssl_mechanism](#) [Gssl_mechanism](#)

Definition at line 59 of file gssl-mech.h.

7.30.1.6 typedef struct [Gsasl_mechanism_functions](#) [Gsasl_mechanism_functions](#)

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

7.30.1.7 typedef int(*) [Gsasl_start_function](#)([Gsasl_session](#) *sctx, void **mech_data)

Definition at line 29 of file `gsasl-mech.h`.

7.30.1.8 typedef int(*) [Gsasl_step_function](#)([Gsasl_session](#) *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)

Definition at line 30 of file `gsasl-mech.h`.

7.30.2 Function Documentation

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

`gsasl_register`:

Parameters:

ctx pointer to libgsasl handle.

mech 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 38 of file `register.c`.

References `Gsasl_mechanism::client`, `Gsasl::client_mechs`, `GSASL_MALLOC_ERROR`, `GSASL_OK`, `Gsasl_mechanism_functions::init`, `Gsasl::n_client_mechs`, `Gsasl::n_server_mechs`, `Gsasl_mechanism::server`, and `Gsasl::server_mechs`.

7.31 gssapi.h File Reference

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

Defines

- #define `GSASL_VERSION` "0.2.15"

Typedefs

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

Enumerations

- enum { `GSASL_MIN_MECHANISM_SIZE` = 1, `GSASL_MAX_MECHANISM_SIZE` = 20 }
- enum `Gssapi_rc` {
 - `GSASL_OK` = 0, `GSASL_NEEDS_MORE` = 1, `GSASL_UNKNOWN_MECHANISM` = 2, `GSASL_MECHANISM_CALLED_TOO_MANY_TIMES` = 3,
 - `GSASL_MALLOC_ERROR` = 7, `GSASL_BASE64_ERROR` = 8, `GSASL_CRYPTO_ERROR` = 9, `GSASL_SASLPREP_ERROR` = 29,
 - `GSASL_MECHANISM_PARSE_ERROR` = 30, `GSASL_AUTHENTICATION_ERROR` = 31, `GSASL_INTEGRITY_ERROR` = 33, `GSASL_NO_CLIENT_CODE` = 35,
 - `GSASL_NO_SERVER_CODE` = 36, `GSASL_NO_CALLBACK` = 51, `GSASL_NO_ANONYMOUS_TOKEN` = 52, `GSASL_NO_AUTHID` = 53,
 - `GSASL_NO_AUTHZID` = 54, `GSASL_NO_PASSWORD` = 55, `GSASL_NO_PASSCODE` = 56, `GSASL_NO_PIN` = 57,
 - `GSASL_NO_SERVICE` = 58, `GSASL_NO_HOSTNAME` = 59, `GSASL_GSSAPI_RELEASE_BUFFER_ERROR` = 37, `GSASL_GSSAPI_IMPORT_NAME_ERROR` = 38,
 - `GSASL_GSSAPI_INIT_SEC_CONTEXT_ERROR` = 39, `GSASL_GSSAPI_ACCEPT_SEC_CONTEXT_ERROR` = 40, `GSASL_GSSAPI_UNWRAP_ERROR` = 41, `GSASL_GSSAPI_WRAP_ERROR` = 42,
 - `GSASL_GSSAPI_ACQUIRE_CRED_ERROR` = 43, `GSASL_GSSAPI_DISPLAY_NAME_ERROR` = 44, `GSASL_GSSAPI_UNSUPPORTED_PROTECTION_ERROR` = 45, `GSASL_KERBEROS_V5_INIT_ERROR` = 46,
 - `GSASL_KERBEROS_V5_INTERNAL_ERROR` = 47, `GSASL_SHISHI_ERROR` = `GSASL_KERBEROS_V5_INTERNAL_ERROR`, `GSASL_SECURID_SERVER_NEED_ADDITIONAL_PASSCODE` = 48, `GSASL_SECURID_SERVER_NEED_NEW_PIN` = 49 }
- enum `Gssapi_qop` { `GSASL_QOP_AUTH` = 1, `GSASL_QOP_AUTH_INT` = 2, `GSASL_QOP_AUTH_CONF` = 4 }

- enum Gsasl_cipher {
 GSASL_CIPHER_DES = 1, GSASL_CIPHER_3DES = 2, GSASL_CIPHER_RC4 = 4, GSASL_CIPHER_RC4_40 = 8,
 GSASL_CIPHER_RC4_56 = 16, GSASL_CIPHER_AES = 32 }
- enum Gsasl_saslprep_flags { GSASL_ALLOW_UNASSIGNED = 1 }
- enum Gsasl_property {
 GSASL_AUTHID = 1, GSASL_AUTHZID = 2, GSASL_PASSWORD = 3, GSASL_ANONYMOUS_TOKEN = 4,
 GSASL_SERVICE = 5, GSASL_HOSTNAME = 6, GSASL_GSSAPI_DISPLAY_NAME = 7, GSASL_PASSCODE = 8,
 GSASL_SUGGESTED_PIN = 9, GSASL_PIN = 10, GSASL_REALM = 11, GSASL_VALIDATE_SIMPLE = 500,
 GSASL_VALIDATE_EXTERNAL = 501, GSASL_VALIDATE_ANONYMOUS = 502, GSASL_VALIDATE_GSSAPI = 503, GSASL_VALIDATE_SECURID = 504 }

Functions

- int gsasl_init (Gsasl **ctx)
- void gsasl_done (Gsasl *ctx)
- const char * gsasl_check_version (const char *req_version)
- 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)
- void gsasl_property_set (Gsasl_session *sctx, Gsasl_property prop, const char *data)
- void gsasl_property_set_raw (Gsasl_session *sctx, Gsasl_property prop, const char *data, size_t len)
- const char * gsasl_property_get (Gsasl_session *sctx, Gsasl_property prop)
- const char * gsasl_property_fast (Gsasl_session *sctx, Gsasl_property prop)
- int gsasl_client_mechlist (Gsasl *ctx, char **out)
- int gsasl_client_support_p (Gsasl *ctx, const char *name)
- const char * gsasl_client_suggest_mechanism (Gsasl *ctx, const char *mechlist)
- int gsasl_server_mechlist (Gsasl *ctx, char **out)
- int gsasl_server_support_p (Gsasl *ctx, const char *name)
- int gsasl_client_start (Gsasl *ctx, const char *mech, Gsasl_session **sctx)
- int gsasl_server_start (Gsasl *ctx, const char *mech, Gsasl_session **sctx)
- 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)
- void gsasl_finish (Gsasl_session *sctx)
- 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)
- const char * gsasl_strerror (int err)
- int gsasl_saslprep (const char *in, Gsasl_saslprep_flags flags, char **out, int *stringprepc)
- int gsasl_simple_getpass (const char *filename, const char *username, char **key)
- int gsasl_base64_to (const char *in, size_t inlen, char **out, size_t *outlen)

- int [gssapi_base64_from](#) (const char *in, size_t inlen, char **out, size_t *outlen)
- int [gssapi_nonce](#) (char *data, size_t datalen)
- int [gssapi_random](#) (char *data, size_t datalen)
- int [gssapi_md5](#) (const char *in, size_t inlen, char *out[16])
- int [gssapi_hmac_md5](#) (const char *key, size_t keylen, const char *in, size_t inlen, char *outhash[16])

Variables

- const char * [GSSAPI_VALID_MECHANISM_CHARACTERS](#)

7.31.1 Define Documentation

7.31.1.1 #define GSSAPI_VERSION "0.2.15"

GSSAPI_VERSION

String defined via CPP denoting the header file version number. Used together with `stringprep_check_version()` to verify header file and run-time library consistency.

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

7.31.2 Typedef Documentation

7.31.2.1 typedef struct [Gssapi](#) [Gssapi](#)

Definition at line 124 of file `gssapi.h`.

7.31.2.2 typedef int(*) [Gssapi_callback_function](#)([Gssapi](#) *ctx, [Gssapi_session](#) *sctx, [Gssapi_property](#) prop)

`Gssapi_callback_function`:

Parameters:

- ctx* libgssapi handle.
- sctx* session handle, may be NULL.
- prop* 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 `GSSAPI_AUTHENTICATION_ERROR` or `GSSAPI_OK` depending on whether the policy permitted the operation.

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

Since: 0.2.0

Definition at line 174 of file `gssapi.h`.

7.31.2.3 typedef struct [Gsasl_session](#) [Gsasl_session](#)

Definition at line 125 of file gsasl.h.

7.31.3 Enumeration Type Documentation

7.31.3.1 anonymous enum

Enumerator:

GSASL_MIN_MECHANISM_SIZE
GSASL_MAX_MECHANISM_SIZE

Definition at line 49 of file gsasl.h.

7.31.3.2 enum [Gsasl_cipher](#)

Enumerator:

GSASL_CIPHER_DES
GSASL_CIPHER_3DES
GSASL_CIPHER_RC4
GSASL_CIPHER_RC4_40
GSASL_CIPHER_RC4_56
GSASL_CIPHER_AES

Definition at line 107 of file gsasl.h.

7.31.3.3 enum [Gsasl_property](#)

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_VALIDATE_SIMPLE
GSASL_VALIDATE_EXTERNAL
GSASL_VALIDATE_ANONYMOUS
GSASL_VALIDATE_GSSAPI
GSASL_VALIDATE_SECURID

Definition at line 128 of file gsasl.h.

7.31.3.4 enum [Gssapi_qop](#)

Enumerator:

GSASL_QOP_AUTH
GSASL_QOP_AUTH_INT
GSASL_QOP_AUTH_CONF

Definition at line 99 of file gssapi.h.

7.31.3.5 enum [Gssapi_rc](#)

Enumerator:

GSASL_OK
GSASL_NEEDS_MORE
GSASL_UNKNOWN_MECHANISM
GSASL_MECHANISM_CALLED_TOO_MANY_TIMES
GSASL_MALLOC_ERROR
GSASL_BASE64_ERROR
GSASL_CRYPTO_ERROR
GSASL_SASLPREP_ERROR
GSASL_MECHANISM_PARSE_ERROR
GSASL_AUTHENTICATION_ERROR
GSASL_INTEGRITY_ERROR
GSASL_NO_CLIENT_CODE
GSASL_NO_SERVER_CODE
GSASL_NO_CALLBACK
GSASL_NO_ANONYMOUS_TOKEN
GSASL_NO_AUTHID
GSASL_NO_AUTHZID
GSASL_NO_PASSWORD
GSASL_NO_PASSCODE
GSASL_NO_PIN
GSASL_NO_SERVICE
GSASL_NO_HOSTNAME
GSASL_GSSAPI_RELEASE_BUFFER_ERROR
GSASL_GSSAPI_IMPORT_NAME_ERROR
GSASL_GSSAPI_INIT_SEC_CONTEXT_ERROR
GSASL_GSSAPI_ACCEPT_SEC_CONTEXT_ERROR
GSASL_GSSAPI_UNWRAP_ERROR
GSASL_GSSAPI_WRAP_ERROR
GSASL_GSSAPI_ACQUIRE_CRED_ERROR
GSASL_GSSAPI_DISPLAY_NAME_ERROR

GSASL_GSSAPI_UNSUPPORTED_PROTECTION_ERROR
GSASL_KERBEROS_V5_INIT_ERROR
GSASL_KERBEROS_V5_INTERNAL_ERROR
GSASL_SHISHI_ERROR
GSASL_SECURID_SERVER_NEED_ADDITIONAL_PASSCODE
GSASL_SECURID_SERVER_NEED_NEW_PIN

Definition at line 57 of file gsasl.h.

7.31.3.6 enum [Gsasl_saslprep_flags](#)

Enumerator:

GSASL_ALLOW_UNASSIGNED

Definition at line 118 of file gsasl.h.

7.31.4 Function Documentation

7.31.4.1 int [gsasl_base64_from](#) (const char * *in*, size_t *inlen*, char ** *out*, size_t * *outlen*)

[gsasl_base64_from](#):

Parameters:

in input byte array
inlen size of input byte array
out pointer to newly allocated output byte array
outlen 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 74 of file base64.c.

References GSASL_BASE64_ERROR, GSASL_MALLOC_ERROR, and GSASL_OK.

Referenced by [gsasl_step64](#)().

7.31.4.2 int [gsasl_base64_to](#) (const char * *in*, size_t *inlen*, char ** *out*, size_t * *outlen*)

[gsasl_base64_to](#):

Parameters:

in input byte array
inlen size of input byte array
out pointer to newly allocated output byte array

outlen pointer to size of newly allocated output byte array

Encode data as base64. The 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 44 of file base64.c.

References GSASL_MALLOC_ERROR, and GSASL_OK.

Referenced by `_gsasl_digest_md5_client_start()`, `_gsasl_digest_md5_server_start()`, and `gsasl_step64()`.

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

`gsasl_callback`:

Parameters:

ctx handle received from `gsasl_init()`, may be NULL to derive it from .

sctx session handle.

prop enumerated value of `Gsasl_property` type.

Invoke the application callback. The 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.

Note that if no callback has been set by the application, but the obsolete callback interface has been used, this function will translate the old callback interface into the new. This interface should be sufficient to invoke all callbacks, both new and old.

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

Since: 0.2.0

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

References `Gsasl_session::anonymous_token`, `Gsasl_session::authid`, `Gsasl_session::authzid`, `Gsasl::cb`, `Gsasl_session::ctx`, `GSASL_NO_CALLBACK`, `GSASL_OK`, `GSASL_PASSWORD`, `gsasl_property_set()`, `gsasl_property_set_raw()`, `gsasl_server_callback_anonymous_get()`, `gsasl_server_callback_external_get()`, `gsasl_server_callback_gssapi_get()`, `gsasl_server_callback_retrieve_get()`, `gsasl_server_callback_securid_get()`, `gsasl_server_callback_validate_get()`, `GSASL_SUGGESTED_PIN`, `GSASL_VALIDATE_ANONYMOUS`, `GSASL_VALIDATE_EXTERNAL`, `GSASL_VALIDATE_GSSAPI`, `GSASL_VALIDATE_SECURID`, `GSASL_VALIDATE_SIMPLE`, `Gsasl_session::gssapi_display_name`, `Gsasl_session::hostname`, `Gsasl_session::passcode`, `Gsasl_session::password`, and `Gsasl_session::pin`.

Referenced by `_gsasl_anonymous_server_step()`, `_gsasl_digest_md5_client_step()`, `_gsasl_external_server_step()`, `_gsasl_gssapi_server_step()`, `_gsasl_login_server_step()`, `_gsasl_plain_server_step()`, `_gsasl_securid_server_step()`, and `gsasl_property_get()`.

7.31.4.4 `void* gsas_callback_hook_get (Gsasl * ctx)`

`gsasl_callback_hook_get`:

Parameters:

ctx 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 206 of file callback.c.

References Gsasl::application_hook.

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

gsasl_callback_hook_set:

Parameters:

ctx libgsasl handle.

hook 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 186 of file callback.c.

References Gsasl::application_hook.

7.31.4.6 void gsasl_callback_set (Gsasl * ctx, Gsasl_callback_function cb)

gsasl_callback_set:

Parameters:

ctx handle received from [gsasl_init\(\)](#).

cb 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 44 of file callback.c.

References Gsasl::cb.

7.31.4.7 `const char* gsas_check_version (const char * req_version)`

gsasl_check_version:

Parameters:

req_version version string to compare with, or NULL.

Check library version.

See GSASL_VERSION for a suitable string.

Return value: Check that the the version of the library is at minimum the one given as a string in 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 42 of file version.c.

7.31.4.8 `int gsas_client_mechlist (Gsasl * ctx, char ** out)`

gsasl_client_mechlist:

Parameters:

ctx libgsasl handle.

out newly allocated output character array.

Return a newly allocated string containing SASL names, separated by space, of mechanisms supported by the libgsasl client. 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 74 of file listmech.c.

References Gsasl::client_mechs, and Gsasl::n_client_mechs.

Referenced by gsas_client_listmech().

7.31.4.9 `int gsas_client_start (Gsasl * ctx, const char * mech, Gsasl_session ** sctx)`

gsasl_client_start:

Parameters:

ctx libgsasl handle.

mech name of SASL mechanism.

sctx pointer to client handle.

This functions initiates a client SASL authentication. This function must be called before any other gsas_client_*(*) function is called.

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

Definition at line 119 of file xstart.c.

References Gsasl::client_mechs, and Gsasl::n_client_mechs.

Referenced by gsas_client_suggest_mechanism().

7.31.4.10 `const char* gsasl_client_suggest_mechanism (Gsasl * ctx, const char * mechlist)`

gsasl_client_suggest_mechanism:

Parameters:

ctx libgsasl handle.

mechlist 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.

Definition at line 37 of file suggest.c.

References Gsasl::client_mechs, gsasl_client_start(), gsasl_finish(), GSASL_OK, GSASL_VALID_MECHANISM_CHARACTERS, Gsasl::n_client_mechs, and Gsasl_mechanism::name.

7.31.4.11 `int gsasl_client_support_p (Gsasl * ctx, const char * name)`

gsasl_client_support_p:

Parameters:

ctx libgsasl handle.

name 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 49 of file supportp.c.

References Gsasl::client_mechs, and Gsasl::n_client_mechs.

7.31.4.12 `int gsasl_decode (Gsasl_session * sctx, const char * input, size_t input_len, char ** output, size_t * output_len)`

gsasl_decode:

Parameters:

sctx libgsasl session handle.

input input byte array.

input_len size of input byte array.

output newly allocated output byte array.

output_len size of output byte array.

Decode data according to negotiated SASL mechanism. This might mean that data is integrity or privacy protected.

The buffer is allocated by this function, and it is the responsibility of caller to deallocate it by calling free().

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

Definition at line 96 of file xcode.c.

References `Gssapi_mechanism::client`, `Gssapi_session::clientp`, `Gssapi_mechanism_functions::decode`, `Gssapi_session::mech`, and `Gssapi_mechanism::server`.

Referenced by `gssapi_decode_inline()`.

7.31.4.13 void gssapi_done (`Gssapi` * *ctx*)

`gssapi_done`:

Parameters:

ctx libgssapi handle.

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

Definition at line 33 of file done.c.

References `Gssapi_mechanism::client`, `Gssapi::client_mechs`, `Gssapi_mechanism_functions::done`, and `Gssapi::n_client_mechs`.

Referenced by `gssapi_init()`.

7.31.4.14 int gssapi_encode (`Gssapi_session` * *sctx*, const char * *input*, size_t *input_len*, char ** *output*, size_t * *output_len*)

`gssapi_encode`:

Parameters:

sctx libgssapi session handle.

input input byte array.

input_len size of input byte array.

output newly allocated output byte array.

output_len size of output byte array.

Encode data according to negotiated SASL mechanism. This might mean that data is integrity or privacy protected.

The buffer is allocated by this function, and it is the responsibility of caller to deallocate it by calling `free()`.

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

Definition at line 64 of file xcode.c.

References `Gssapi_mechanism::client`, `Gssapi_session::clientp`, `Gssapi_mechanism_functions::encode`, `Gssapi_session::mech`, and `Gssapi_mechanism::server`.

Referenced by `gssapi_encode_inline()`.

7.31.4.15 void gssapi_finish (`Gssapi_session` * *sctx*)

`gssapi_finish`:

Parameters:

sctx 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 33 of file xfinish.c.

References Gsasl_session::anonymous_token, Gsasl_session::authid, Gsasl_session::authzid, Gsasl_mechanism::client, Gsasl_session::clientp, Gsasl_mechanism_functions::finish, Gsasl_session::gssapi_display_name, Gsasl_session::hostname, Gsasl_session::mech, Gsasl_session::mech_data, Gsasl_session::passcode, Gsasl_session::password, Gsasl_session::pin, Gsasl_session::realm, Gsasl_mechanism::server, Gsasl_session::service, and Gsasl_session::suggestedpin.

Referenced by gsasl_client_finish(), gsasl_client_suggest_mechanism(), and gsasl_server_finish().

7.31.4.16 int gsasl_hmac_md5 (const char * key, size_t keylen, const char * in, size_t inlen, char * outhash[16])

gsasl_hmac_md5:

Parameters:

key input character array with key to use.

keylen length of input character array with key to use.

in input character array of data to hash.

inlen length of input character array of data to hash.

outhash newly allocated character array with keyed hash of data.

Compute keyed checksum of data using HMAC-MD5. The buffer must be deallocated by the caller.

Return value: Returns GSASL_OK iff successful.

Definition at line 92 of file crypto.c.

References GSASL_MALLOC_ERROR.

7.31.4.17 int gsasl_init (Gsasl ** ctx)

gsasl_init:

Parameters:

ctx 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 124 of file init.c.

References Gsasl_session::ctx, GSASL_CRYPTO_ERROR, gsasl_done(), GSASL_MALLOC_ERROR, and GSASL_OK.

7.31.4.18 int gsas_md5 (const char * in, size_t inlen, char * out[16])

gsasl_md5:

Parameters:

- in* input character array of data to hash.
- inlen* length of input character array of data to hash.
- out* newly allocated character array with hash of data.

Compute hash of data using MD5. The buffer must be deallocated by the caller.

Return value: Returns GSASL_OK iff successful.

Definition at line 70 of file crypto.c.

References GSASL_MALLOC_ERROR.

Referenced by `_gsasl_digest_md5_client_step()`, and `_gsasl_digest_md5_server_step()`.

7.31.4.19 int gsas_nonce (char * data, size_t datalen)

gsasl_nonce:

Parameters:

- data* output array to be filled with unpredictable random data.
- datalen* size of output array.

Store unpredictable data of given size in the provided buffer.

Return value: Returns GSASL_OK iff successful.

Definition at line 37 of file crypto.c.

Referenced by `_gsasl_digest_md5_client_start()`, `_gsasl_digest_md5_server_start()`, and `gsasl_randomize()`.

7.31.4.20 const char* gsas_property_fast (Gsas_session * sctx, Gsas_property prop)

gsasl_property_fast:

Parameters:

- sctx* session handle.
- prop* enumerated value of Gsas_property type, indicating the type of data in .

Retrieve the data stored in the session handle for given property .

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 171 of file property.c.

Referenced by `_gsasl_digest_md5_client_step()`, and `gsasl_property_get()`.

7.31.4.21 `const char* gsasl_property_get (Gsasl_session * sctx, Gsasl_property prop)`

`gsasl_property_get`:

Parameters:

sctx session handle.

prop enumerated value of `Gsasl_property` type, indicating the type of data in .

Retrieve the data stored in the session handle for given property , 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.

If no value is known, and no callback is specified or if the callback fail to return data, and if any obsolete callback functions has been set by the application, this function will try to call these obsolete callbacks, and store the returned data as the corresponding property. This behaviour of this function will be removed when the obsolete callback interfaces are removed.

Return value: Return data for property, or NULL if no value known.

Since: 0.2.0

Definition at line 208 of file `property.c`.

References `Gsasl_session::ctx`, `GSASL_ANONYMOUS_TOKEN`, `GSASL_AUTHID`, `GSASL_AUTHZID`, `gsasl_callback()`, `gsasl_client_callback_anonymous_get()`, `gsasl_client_callback_authentication_id_get()`, `gsasl_client_callback_authorization_id_get()`, `gsasl_client_callback_passcode_get()`, `gsasl_client_callback_password_get()`, `gsasl_client_callback_pin_get()`, `gsasl_client_callback_realm_get()`, `gsasl_client_callback_service_get()`, `GSASL_HOSTNAME`, `GSASL_OK`, `GSASL_PASSCODE`, `GSASL_PASSWORD`, `GSASL_PIN`, `gsasl_property_fast()`, `gsasl_property_set()`, `GSASL_REALM`, `GSASL_SERVICE`, and `Gsasl_session::suggestedpin`.

Referenced by `_gsasl_anonymous_client_step()`, `_gsasl_cram_md5_client_step()`, `_gsasl_cram_md5_server_step()`, `_gsasl_digest_md5_client_step()`, `_gsasl_digest_md5_server_step()`, `_gsasl_external_client_step()`, `_gsasl_gssapi_client_step()`, `_gsasl_gssapi_server_start()`, `_gsasl_login_client_step()`, `_gsasl_login_server_step()`, `_gsasl_ntlm_client_step()`, `_gsasl_plain_client_step()`, `_gsasl_plain_server_step()`, `_gsasl_secured_client_step()`, and `_gsasl_secured_server_step()`.

7.31.4.22 `void gsasl_property_set (Gsasl_session * sctx, Gsasl_property prop, const char * data)`

`gsasl_property_set`:

Parameters:

sctx session handle.

prop enumerated value of `Gsasl_property` type, indicating the type of data in .

data zero terminated character string to store.

Make a copy of and store it in the session handle for the indicated property .

You can immediately deallocate after calling this function, without affecting the data stored in the session handle.

Since: 0.2.0

Definition at line 102 of file property.c.

References `gssapi_property_set_raw()`.

Referenced by `_gssapi_cram_md5_server_step()`, `_gssapi_digest_md5_client_step()`, `_gssapi_digest_md5_server_step()`, `_gssapi_external_server_step()`, `_gssapi_login_server_step()`, `_gssapi_plain_server_step()`, `_gssapi_securid_server_step()`, `gssapi_callback()`, and `gssapi_property_get()`.

7.31.4.23 void gssapi_property_set_raw (Gssapi_session * *sctx*, Gssapi_property *prop*, const char * *data*, size_t *len*)

`gssapi_property_set_raw`:

Parameters:

sctx session handle.

prop enumerated value of Gssapi_property type, indicating the type of data in .

data character string to store.

len length of character string to store.

Make a copy of *data* and store a zero terminated version of it in the session handle for the indicated property .

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 `gssapi_property_set`.

Since: 0.2.0

Definition at line 128 of file property.c.

Referenced by `_gssapi_anonymous_server_step()`, `_gssapi_external_server_step()`, `_gssapi_gssapi_server_step()`, `_gssapi_securid_client_step()`, `gssapi_callback()`, and `gssapi_property_set()`.

7.31.4.24 int gssapi_random (char * *data*, size_t *datalen*)

`gssapi_random`:

Parameters:

data output array to be filled with strong random data.

datalen 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 53 of file crypto.c.

Referenced by `gssapi_randomize()`.

7.31.4.25 int gssapi_saslprep (const char * *in*, Gssapi_saslprep_flags *flags*, char ** *out*, int * *stringprepc*)

`gssapi_saslprep` - prepare internationalized string

Parameters:

- in* a UTF-8 encoded string.
- flags* any SASLprep flag, e.g., GSASL_ALLOW_UNASSIGNED.
- out* on exit, contains newly allocated output string.
- stringpreprc* if non-NULL, will hold precise stringprep return code.

Prepare string using SASLprep. On success, the variable must be deallocated by the caller.

Return value: Returns GSASL_OK on success, or GSASL_SASLPREP_ERROR on error.

Since: 0.2.3

Definition at line 48 of file saslprep.c.

References GSASL_ALLOW_UNASSIGNED, GSASL_MALLOC_ERROR, GSASL_OK, and GSASL_SASLPREP_ERROR.

Referenced by _gsasl_cram_md5_client_step(), _gsasl_cram_md5_server_step(), and _gsasl_plain_server_step().

7.31.4.26 int gsasl_server_mechlist (Gsasl * ctx, char ** out)

gsasl_server_mechlist:

Parameters:

- ctx* libgsasl handle.
- out* newly allocated output character array.

Return a newly allocated string containing SASL names, separated by space, of mechanisms supported by the libgsasl server. 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 93 of file listmech.c.

References Gsasl::n_server_mechs, and Gsasl::server_mechs.

Referenced by gsasl_server_listmech().

7.31.4.27 int gsasl_server_start (Gsasl * ctx, const char * mech, Gsasl_session ** sctx)

gsasl_server_start:

Parameters:

- ctx* libgsasl handle.
- mech* name of SASL mechanism.
- sctx* 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 137 of file xstart.c.

References Gsasl::n_server_mechs, and Gsasl::server_mechs.

7.31.4.28 int gssapi_server_support_p (Gssapi * ctx, const char * name)

gssapi_server_support_p:

Parameters:

- ctx* libgssapi handle.
- name* 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 66 of file supportp.c.

References Gssapi::n_server_mechs, and Gssapi::server_mechs.

7.31.4.29 void* gssapi_session_hook_get (Gssapi_session * sctx)

gssapi_session_hook_get:

Parameters:

- sctx* 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 246 of file callback.c.

References Gssapi_session::application_hook.

7.31.4.30 void gssapi_session_hook_set (Gssapi_session * sctx, void * hook)

gssapi_session_hook_set:

Parameters:

- sctx* libgssapi session handle.
- hook* 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 226 of file callback.c.

References Gssapi_session::application_hook.

7.31.4.31 `int gsasl_simple_getpass (const char * filename, const char * username, char ** key)`

`gsasl_simple_getpass`:

Parameters:

filename filename of file containing passwords.

username username string.

key newly allocated output character array.

Retrieve password for user from specified file. The buffer 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 47 of file md5pwd.c.

References GSASL_AUTHENTICATION_ERROR, GSASL_MALLOC_ERROR, and GSASL_OK.

7.31.4.32 `int gsasl_step (Gsasl_session * sctx, const char * input, size_t input_len, char ** output, size_t * output_len)`

`gsasl_step`:

Parameters:

sctx libgsasl session handle.

input input byte array.

input_len size of input byte array.

output newly allocated output byte array.

output_len pointer to output variable with size of output byte array.

Perform one step of SASL authentication. This reads data from the other end (from and), processes it (potentially invoking callbacks to the application), and writes data to server (into newly allocated variable and that indicate the length of).

The contents of the 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 buffer is allocated by this function, and it is the responsibility of caller to deallocate it by calling free ().

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

Definition at line 51 of file xstep.c.

References Gsasl_mechanism::client, Gsasl_session::clientp, Gsasl_session::mech, Gsasl_session::mech_data, Gsasl_mechanism::server, and Gsasl_mechanism_functions::step.

Referenced by `gsasl_step64()`.

7.31.4.33 int gssapi_step64 (Gssapi_session * sctx, const char * b64input, char ** b64output)

gssapi_step64:

Parameters:

sctx libgssapi client handle.

b64input input base64 encoded byte array.

b64output newly allocated output base64 encoded byte array.

This is a simple wrapper around [gssapi_step\(\)](#) that base64 decodes the input and base64 encodes the output.

The contents of the 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 buffer is allocated by this function, and it is the responsibility of caller to deallocate it by calling free ().

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

Definition at line 86 of file xstep.c.

References GSASL_BASE64_ERROR, gssapi_base64_from(), gssapi_base64_to(), GSASL_NEEDS_MORE, GSASL_OK, and gssapi_step().

7.31.4.34 const char* gssapi_strerror (int err)

gssapi_strerror:

Parameters:

err libgssapi error code

Convert return code to human readable string.

Return value: Returns a pointer to a statically allocated string containing a description of the error with the error value . This string can be used to output a diagnostic message to the user.

Definition at line 42 of file error.c.

References GSASL_AUTHENTICATION_ERROR, GSASL_BASE64_ERROR, GSASL_CANNOT_GET_CTX, GSASL_CRYPTO_ERROR, GSASL_FCLOSE_ERROR, GSASL_FOPEN_ERROR, GSASL_GSSAPI_ACCEPT_SEC_CONTEXT_ERROR, GSASL_GSSAPI_ACQUIRE_CRED_ERROR, GSASL_GSSAPI_DISPLAY_NAME_ERROR, GSASL_GSSAPI_IMPORT_NAME_ERROR, GSASL_GSSAPI_INIT_SEC_CONTEXT_ERROR, GSASL_GSSAPI_RELEASE_BUFFER_ERROR, GSASL_GSSAPI_UNSUPPORTED_PROTECTION_ERROR, GSASL_GSSAPI_UNWRAP_ERROR, GSASL_GSSAPI_WRAP_ERROR, GSASL_INTEGRITY_ERROR, GSASL_INVALID_HANDLE, GSASL_MALLOC_ERROR, GSASL_MECHANISM_CALLED_TOO_MANY_TIMES, GSASL_MECHANISM_PARSE_ERROR, GSASL_NEED_CLIENT_ANONYMOUS_CALLBACK, GSASL_NEED_CLIENT_AUTHENTICATION_ID_CALLBACK, GSASL_NEED_CLIENT_AUTHORIZATION_ID_CALLBACK, GSASL_NEED_CLIENT_PASSCODE_CALLBACK, GSASL_NEED_CLIENT_PASSWORD_CALLBACK, GSASL_NEED_CLIENT_PIN_CALLBACK, GSASL_NEED_CLIENT_SERVICE_CALLBACK, GSASL_NEED_SERVER_ANONYMOUS_CALLBACK, GSASL_NEED_SERVER_CRAM_MD5_CALLBACK, GSASL_NEED_SERVER_DIGEST_MD5_CALLBACK, GSASL_NEED_SERVER_EXTERNAL_CALLBACK, GSASL_NEED_SERVER_GSSAPI_CALLBACK, GSASL_NEED_SERVER_REALM_CALLBACK, GSASL_NEED_SERVER_RETRIEVE_CALLBACK, GSASL_NEED_SERVER_SECURID_CALLBACK, GSASL_NEED_SERVER_SERVICE_CALLBACK, GSASL_NEED_SERVER_VALIDATE_CALLBACK, GSASL_

NEEDS_MORE, GSASL_NO_ANONYMOUS_TOKEN, GSASL_NO_AUTHID, GSASL_NO_AUTHZID, GSASL_NO_CALLBACK, GSASL_NO_CLIENT_CODE, GSASL_NO_HOSTNAME, GSASL_NO_MORE_REALMS, GSASL_NO_PASSCODE, GSASL_NO_PASSWORD, GSASL_NO_PIN, GSASL_NO_SERVER_CODE, GSASL_NO_SERVICE, GSASL_OK, GSASL_SASLPREP_ERROR, GSASL_TOO_SMALL_BUFFER, GSASL_UNICODE_NORMALIZATION_ERROR, and GSASL_UNKNOWN_MECHANISM.

7.31.5 Variable Documentation

7.31.5.1 `const char*` [GSASL_VALID_MECHANISM_CHARACTERS](#)

GSASL_VALID_MECHANISM_CHARACTERS:

A zero-terminated character array, or string, with all ASCII characters that may be used within a SASL mechanism name.

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

Referenced by `gsasl_client_suggest_mechanism()`.

7.32 `init.c` File Reference

```
#include "internal.h"
#include <gc.h>
#include "cram-md5/cram-md5.h"
#include "external/external.h"
#include "gssapi/x-gssapi.h"
#include "anonymous/anonymous.h"
#include "plain/plain.h"
#include "securid/securid.h"
#include "digest-md5/digest-md5.h"
#include "login/login.h"
#include "ntlm/x-ntlm.h"
#include "kerberos_v5/kerberos_v5.h"
```

Functions

- `int gssasl_init (Gssasl **ctx)`

Variables

- `const char * GSASL_VALID_MECHANISM_CHARACTERS`

7.32.1 Function Documentation

7.32.1.1 `int gssasl_init (Gssasl ** ctx)`

`gssasl_init`:

Parameters:

`ctx` pointer to libgssasl handle.

This functions initializes libgssasl. The handle pointed to by `ctx` is valid for use with other libgssasl functions iff this function is successful. It also register all builtin SASL mechanisms, using `gssasl_register()`.

Return value: `GSASL_OK` iff successful, otherwise `GSASL_MALLOC_ERROR`.

Definition at line 124 of file `init.c`.

References `Gssasl_session::ctx`, `GSASL_CRYPTO_ERROR`, `gssasl_done()`, `GSASL_MALLOC_ERROR`, and `GSASL_OK`.

7.32.2 Variable Documentation

7.32.2.1 `const char* GSASL_VALID_MECHANISM_CHARACTERS`

Initial value:

```
"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-_"
```

GSASL_VALID_MECHANISM_CHARACTERS:

A zero-terminated character array, or string, with all ASCII characters that may be used within a SASL mechanism name.

Definition at line 47 of file init.c.

Referenced by `gsasl_client_suggest_mechanism()`.

7.33 internal.h File Reference

```
#include "gsasl.h"  
#include <stdlib.h>  
#include <string.h>
```

Data Structures

- struct [Gsasl](#)
- struct [Gsasl_session](#)

7.34 kerberos_v5.c File Reference

```
#include "kerberos_v5.h"  
#include <shishi.h>
```

Defines

- #define `DEBUG` 0
- #define `BITMAP_LEN` 1
- #define `MAXBUF_LEN` 4
- #define `RANDOM_LEN` 16
- #define `MUTUAL` (1 << 3)
- #define `SERVER_HELLO_LEN` `BITMAP_LEN + MAXBUF_LEN + RANDOM_LEN`
- #define `CLIENT_HELLO_LEN` `BITMAP_LEN + MAXBUF_LEN`
- #define `MAXBUF_DEFAULT` 65536

7.34.1 Define Documentation

7.34.1.1 #define `BITMAP_LEN` 1

Definition at line 37 of file `kerberos_v5.c`.

Referenced by `_gsasl_kerberos_v5_client_step()`, and `_gsasl_kerberos_v5_server_step()`.

7.34.1.2 #define `CLIENT_HELLO_LEN` `BITMAP_LEN + MAXBUF_LEN`

Definition at line 43 of file `kerberos_v5.c`.

Referenced by `_gsasl_kerberos_v5_client_step()`, and `_gsasl_kerberos_v5_server_step()`.

7.34.1.3 #define `DEBUG` 0

Definition at line 35 of file `kerberos_v5.c`.

7.34.1.4 #define `MAXBUF_DEFAULT` 65536

Definition at line 45 of file `kerberos_v5.c`.

Referenced by `_gsasl_kerberos_v5_client_step()`, and `_gsasl_kerberos_v5_server_step()`.

7.34.1.5 #define `MAXBUF_LEN` 4

Definition at line 38 of file `kerberos_v5.c`.

Referenced by `_gsasl_kerberos_v5_client_step()`, and `_gsasl_kerberos_v5_server_step()`.

7.34.1.6 #define `MUTUAL` (1 << 3)

Definition at line 40 of file `kerberos_v5.c`.

Referenced by `_gsasl_kerberos_v5_client_step()`, and `_gsasl_kerberos_v5_server_step()`.

7.34.1.7 #define RANDOM_LEN 16

Definition at line 39 of file kerberos_v5.c.

Referenced by `_gsasl_kerberos_v5_server_start()`, and `_gsasl_kerberos_v5_server_step()`.

7.34.1.8 #define SERVER_HELLO_LEN BITMAP_LEN + MAXBUF_LEN + RANDOM_LEN

Definition at line 42 of file kerberos_v5.c.

Referenced by `_gsasl_kerberos_v5_client_step()`, and `_gsasl_kerberos_v5_server_step()`.

7.35 kerberos_v5.h File Reference

```
#include "internal.h"
```

Defines

- `#define _GSASL_KERBEROS_V5_NAME "KERBEROS_V5"`

Functions

- `int _gsasl_kerberos_v5_client_init (Gsasl *ctx)`
- `int _gsasl_kerberos_v5_client_start (Gsasl_session *sctx, void **mech_data)`
- `int _gsasl_kerberos_v5_client_step (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char *output, size_t *output_len)`
- `int _gsasl_kerberos_v5_client_encode (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char *output, size_t *output_len)`
- `int _gsasl_kerberos_v5_client_decode (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char *output, size_t *output_len)`
- `int _gsasl_kerberos_v5_client_finish (Gsasl_session *sctx, void *mech_data)`
- `int _gsasl_kerberos_v5_server_init (Gsasl *ctx)`
- `int _gsasl_kerberos_v5_server_start (Gsasl_session *sctx, void **mech_data)`
- `int _gsasl_kerberos_v5_server_step (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char *output, size_t *output_len)`
- `int _gsasl_kerberos_v5_server_encode (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char *output, size_t *output_len)`
- `int _gsasl_kerberos_v5_server_decode (Gsasl_session *sctx, void *mech_data, const char *input, size_t input_len, char *output, size_t *output_len)`
- `int _gsasl_kerberos_v5_server_finish (Gsasl_session *sctx, void *mech_data)`

7.35.1 Define Documentation

7.35.1.1 `#define _GSASL_KERBEROS_V5_NAME "KERBEROS_V5"`

Definition at line 31 of file `kerberos_v5.h`.

7.35.2 Function Documentation

7.35.2.1 `int _gsasl_kerberos_v5_client_decode (Gsasl_session * sctx, void * mech_data, const char * input, size_t input_len, char * output, size_t * output_len)`

Definition at line 414 of file `kerberos_v5/client.c`.

References `_Gsasl_kerberos_v5_client_state::clientqop`, `GSASL_INTEGRITY_ERROR`, `GSASL_MALLOC_ERROR`, `GSASL_OK`, `GSASL_QOP_AUTH_CONF`, `GSASL_QOP_AUTH_INT`, and `_Gsasl_kerberos_v5_client_state::sessionkey`.

7.35.2.2 `int _gsasl_kerberos_v5_client_encode (Gsasl_session * sctx, void * mech_data, const char * input, size_t input_len, char * output, size_t * output_len)`

7.35.2.3 `int _gsasl_kerberos_v5_client_finish (Gsasl_session * sctx, void * mech_data)`

Definition at line 444 of file kerberos_v5/client.c.

References GSASL_OK, and `_Gsasl_kerberos_v5_client_state::sh`.

7.35.2.4 `int _gsasl_kerberos_v5_client_init (Gsasl * ctx)`

7.35.2.5 `int _gsasl_kerberos_v5_client_start (Gsasl_session * sctx, void ** mech_data)`

Definition at line 57 of file kerberos_v5/client.c.

References GSASL_KERBEROS_V5_INIT_ERROR, GSASL_MALLOC_ERROR, GSASL_OK, and GSASL_QOP_AUTH_INT.

7.35.2.6 `int _gsasl_kerberos_v5_client_step (Gsasl_session * sctx, void * mech_data, const char * input, size_t input_len, char * output, size_t * output_len)`

Definition at line 89 of file kerberos_v5/client.c.

References `_Gsasl_kerberos_v5_client_state::ap`, `_Gsasl_kerberos_v5_client_state::as`, `BITMAP_LEN`, `CLIENT_HELLO_LEN`, `_Gsasl_kerberos_v5_client_state::clientmaxbuf`, `_Gsasl_kerberos_v5_client_state::clientqop`, `GSASL_AUTHENTICATION_ERROR`, `GSASL_CANNOT_GET_CTX`, `gsasl_client_callback_authentication_id_get()`, `gsasl_client_callback_authorization_id_get()`, `gsasl_client_callback_maxbuf_get()`, `gsasl_client_callback_password_get()`, `gsasl_client_callback_qop_get()`, `gsasl_client_callback_realm_get()`, `gsasl_client_callback_service_get()`, `gsasl_client_ctx_get()`, `GSASL_KERBEROS_V5_INTERNAL_ERROR`, `GSASL_MECHANISM_CALLED_TOO_MANY_TIMES`, `GSASL_MECHANISM_PARSE_ERROR`, `GSASL_NEED_CLIENT_PASSWORD_CALLBACK`, `GSASL_NEED_CLIENT_SERVICE_CALLBACK`, `GSASL_NEEDS_MORE`, `GSASL_OK`, `GSASL_QOP_AUTH`, `GSASL_QOP_AUTH_CONF`, `GSASL_QOP_AUTH_INT`, `GSASL_TOO_SMALL_BUFFER`, `MAXBUF_DEFAULT`, `MAXBUF_LEN`, `MUTUAL`, `SERVER_HELLO_LEN`, `_Gsasl_kerberos_v5_client_state::serverhello`, `_Gsasl_kerberos_v5_client_state::servermaxbuf`, `_Gsasl_kerberos_v5_client_state::servermutual`, `_Gsasl_kerberos_v5_client_state::serverqops`, `_Gsasl_kerberos_v5_client_state::sessionkey`, `_Gsasl_kerberos_v5_client_state::sh`, `_Gsasl_kerberos_v5_client_state::step`, `STEP_FIRST`, `STEP_NONINFRA_SEND_APREQ`, `STEP_NONINFRA_SEND_ASREQ`, `STEP_NONINFRA_WAIT_APREP`, `STEP_NONINFRA_WAIT_ASREP`, and `STEP_SUCCESS`.

7.35.2.7 `int _gsasl_kerberos_v5_server_decode (Gsasl_session * sctx, void * mech_data, const char * input, size_t input_len, char * output, size_t * output_len)`

Definition at line 535 of file kerberos_v5/server.c.

References `_Gsasl_kerberos_v5_server_state::clientqop`, `GSASL_INTEGRITY_ERROR`, `GSASL_KERBEROS_V5_INTERNAL_ERROR`, `GSASL_MALLOC_ERROR`, `GSASL_OK`, `GSASL_QOP_AUTH_CONF`, `GSASL_QOP_AUTH_INT`, `_Gsasl_kerberos_v5_server_state::safe`, `_Gsasl_kerberos_v5_server_state::sessionkey`, and `_Gsasl_kerberos_v5_server_state::sh`.

7.35.2.8 `int _gsasl_kerberos_v5_server_encode (Gsasl_session * sctx, void * mech_data, const char * input, size_t input_len, char * output, size_t * output_len)`

Definition at line 488 of file `kerberos_v5/server.c`.

References `_Gsasl_kerberos_v5_server_state::clientqop`, `GSASL_INTEGRITY_ERROR`, `GSASL_KERBEROS_V5_INTERNAL_ERROR`, `GSASL_MALLOC_ERROR`, `GSASL_OK`, `GSASL_QOP_AUTH_CONF`, `GSASL_QOP_AUTH_INT`, `_Gsasl_kerberos_v5_server_state::safe`, `_Gsasl_kerberos_v5_server_state::sessionkey`, and `_Gsasl_kerberos_v5_server_state::sh`.

7.35.2.9 `int _gsasl_kerberos_v5_server_finish (Gsasl_session * sctx, void * mech_data)`

Definition at line 586 of file `kerberos_v5/server.c`.

References `GSASL_OK`, `_Gsasl_kerberos_v5_server_state::password`, `_Gsasl_kerberos_v5_server_state::random`, `_Gsasl_kerberos_v5_server_state::sh`, and `_Gsasl_kerberos_v5_server_state::username`.

7.35.2.10 `int _gsasl_kerberos_v5_server_init (Gsasl * ctx)`

7.35.2.11 `int _gsasl_kerberos_v5_server_start (Gsasl_session * sctx, void ** mech_data)`

Definition at line 65 of file `kerberos_v5/server.c`.

References `GSASL_KERBEROS_V5_INIT_ERROR`, `GSASL_KERBEROS_V5_INTERNAL_ERROR`, `GSASL_MALLOC_ERROR`, `GSASL_OK`, `GSASL_QOP_AUTH`, `GSASL_QOP_AUTH_INT`, and `RANDOM_LEN`.

7.35.2.12 `int _gsasl_kerberos_v5_server_step (Gsasl_session * sctx, void * mech_data, const char * input, size_t input_len, char * output, size_t * output_len)`

Definition at line 106 of file `kerberos_v5/server.c`.

References `_Gsasl_kerberos_v5_server_state::ap`, `_Gsasl_kerberos_v5_server_state::as`, `BITMAP_LEN`, `CLIENT_HELLO_LEN`, `_Gsasl_kerberos_v5_server_state::clientmaxbuf`, `_Gsasl_kerberos_v5_server_state::clientmutual`, `_Gsasl_kerberos_v5_server_state::clientqop`, `_Gsasl_kerberos_v5_server_state::firststep`, `GSASL_AUTHENTICATION_ERROR`, `GSASL_CANNOT_GET_CTX`, `GSASL_KERBEROS_V5_INTERNAL_ERROR`, `GSASL_MALLOC_ERROR`, `GSASL_NEED_SERVER_RETRIEVE_CALLBACK`, `GSASL_NEED_SERVER_SERVICE_CALLBACK`, `GSASL_NEEDS_MORE`, `GSASL_OK`, `GSASL_QOP_AUTH`, `GSASL_QOP_AUTH_CONF`, `GSASL_QOP_AUTH_INT`, `gsasl_server_callback_maxbuf_get()`, `gsasl_server_callback_qop_get()`, `gsasl_server_callback_realm_get()`, `gsasl_server_callback_retrieve_get()`, `gsasl_server_callback_service_get()`, `gsasl_server_ctx_get()`, `GSASL_TOO_SMALL_BUFFER`, `MAXBUF_DEFAULT`, `MAXBUF_LEN`, `MUTUAL`, `_Gsasl_kerberos_v5_server_state::password`, `_Gsasl_kerberos_v5_server_state::random`, `RANDOM_LEN`, `SERVER_HELLO_LEN`, `_Gsasl_kerberos_v5_server_state::serverhello`, `_Gsasl_kerberos_v5_server_state::serverhostname`, `_Gsasl_kerberos_v5_server_state::servermaxbuf`, `_Gsasl_kerberos_v5_server_state::serverqops`, `_Gsasl_kerberos_v5_server_state::serverrealm`, `_Gsasl_kerberos_v5_server_state::serverservice`, `_Gsasl_kerberos_v5_server_state::sessionkey`, `_Gsasl_kerberos_v5_server_state::sessiontkkey`, `_Gsasl_kerberos_v5_server_state::sh`, `_Gsasl_kerberos_v5_server_state::userkey`, `_Gsasl_kerberos_v5_server_state::username`, and `_Gsasl_kerberos_v5_server_state::userrealm`.

7.36 listmech.c File Reference

```
#include "internal.h"
```

Functions

- int [gsasl_client_mechlist](#) (Gsasl *ctx, char **out)
- int [gsasl_server_mechlist](#) (Gsasl *ctx, char **out)

7.36.1 Function Documentation

7.36.1.1 int [gsasl_client_mechlist](#) (Gsasl * ctx, char ** out)

[gsasl_client_mechlist](#):

Parameters:

- ctx* libgsasl handle.
- out* newly allocated output character array.

Return a newly allocated string containing SASL names, separated by space, of mechanisms supported by the libgsasl client. 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 74 of file listmech.c.

References [Gsasl::client_mechs](#), and [Gsasl::n_client_mechs](#).

Referenced by [gsasl_client_listmech\(\)](#).

7.36.1.2 int [gsasl_server_mechlist](#) (Gsasl * ctx, char ** out)

[gsasl_server_mechlist](#):

Parameters:

- ctx* libgsasl handle.
- out* newly allocated output character array.

Return a newly allocated string containing SASL names, separated by space, of mechanisms supported by the libgsasl server. 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 93 of file listmech.c.

References [Gsasl::n_server_mechs](#), and [Gsasl::server_mechs](#).

Referenced by [gsasl_server_listmech\(\)](#).

7.37 login.h File Reference

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

Defines

- `#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`

7.37.1 Define Documentation

7.37.1.1 `#define GSASL_LOGIN_NAME "LOGIN"`

Definition at line 28 of file login.h.

7.37.2 Function Documentation

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

Definition at line 102 of file login/client.c.

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

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

References `GSASL_MALLOC_ERROR`, and `GSASL_OK`.

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

Definition at line 58 of file login/client.c.

References `GSASL_AUTHID`, `GSASL_MECHANISM_CALLED_TOO_MANY_TIMES`, `GSASL_NEEDS_MORE`, `GSASL_NO_AUTHID`, `GSASL_NO_PASSWORD`, `GSASL_OK`, `GSASL_PASSWORD`, `gsasl_property_get()`, and `_Gsasl_login_client_state::step`.

7.37.2.4 void `_gsasl_login_server_finish` ([Gsasl_session](#) * *sctx*, void * *mech_data*)

Definition at line 149 of file login/server.c.

References `_Gsasl_login_server_state::password`, and `_Gsasl_login_server_state::username`.

7.37.2.5 int `_gsasl_login_server_start` ([Gsasl_session](#) * *sctx*, void ** *mech_data*)

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

References `GSASL_MALLOC_ERROR`, and `GSASL_OK`.

7.37.2.6 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 61 of file login/server.c.

References `CHALLENGE_PASSWORD`, `CHALLENGE_USERNAME`, `GSASL_AUTHENTICATION_ERROR`, `GSASL_AUTHID`, `GSASL_AUTHZID`, `gsasl_callback()`, `GSASL_MALLOC_ERROR`, `GSASL_MECHANISM_CALLED_TOO_MANY_TIMES`, `GSASL_MECHANISM_PARSE_ERROR`, `GSASL_NEEDS_MORE`, `GSASL_NO_CALLBACK`, `GSASL_OK`, `GSASL_PASSWORD`, `gsasl_property_get()`, `gsasl_property_set()`, `GSASL_VALIDATE_SIMPLE`, `_Gsasl_login_server_state::password`, `_Gsasl_login_server_state::step`, and `_Gsasl_login_server_state::username`.

7.37.3 Variable Documentation

7.37.3.1 [Gsasl_mechanism](#) `gsasl_login_mechanism`

Definition at line 30 of file login/mechinfo.c.

7.38 md5pwd.c File Reference

```
#include "internal.h"
```

Functions

- int [gsasl_simple_getpass](#) (const char *filename, const char *username, char **key)

7.38.1 Function Documentation

7.38.1.1 int [gsasl_simple_getpass](#) (const char * *filename*, const char * *username*, char ** *key*)

[gsasl_simple_getpass](#):

Parameters:

filename filename of file containing passwords.

username username string.

key newly allocated output character array.

Retrieve password for user from specified file. The buffer 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 47 of file md5pwd.c.

References GSASL_AUTHENTICATION_ERROR, GSASL_MALLOC_ERROR, and GSASL_OK.

7.39 mechinfo.c File Reference

```
#include "anonymous.h"
```

Variables

- [Gsasl_mechanism gsasl_anonymous_mechanism](#)

7.39.1 Variable Documentation

7.39.1.1 [Gsasl_mechanism gsasl_anonymous_mechanism](#)

Initial value:

```
{
  GSASL_ANONYMOUS_NAME,
  {
    NULL,
    NULL,
    NULL,

    NULL,

    NULL,
    NULL,
    NULL}
,
{
  NULL,
  NULL,
  NULL,

  NULL,

  NULL,
  NULL,
  NULL}
}
```

Definition at line 30 of file anonymous/mechinfo.c.

7.40 mechinfo.c File Reference

```
#include "cram-md5.h"
```

Variables

- [Gsasl_mechanism gsasl_cram_md5_mechanism](#)

7.40.1 Variable Documentation

7.40.1.1 [Gsasl_mechanism gsasl_cram_md5_mechanism](#)

Definition at line 30 of file cram-md5/mechinfo.c.

7.41 mechinfo.c File Reference

```
#include "digest-md5.h"
```

Variables

- [Gsasl_mechanism gsasl_digest_md5_mechanism](#)

7.41.1 Variable Documentation

7.41.1.1 [Gsasl_mechanism gsasl_digest_md5_mechanism](#)

Definition at line 30 of file digest-md5/mechinfo.c.

7.42 mechinfo.c File Reference

```
#include "external.h"
```

Variables

- [Gsasl_mechanism gsasl_external_mechanism](#)

7.42.1 Variable Documentation

7.42.1.1 [Gsasl_mechanism gsasl_external_mechanism](#)

Initial value:

```
{
  GSASL_EXTERNAL_NAME,
  {
    NULL,
    NULL,
    NULL,

    NULL,

    NULL,
    NULL,
    NULL}
,
{
  NULL,
  NULL,
  NULL,

  NULL,

  NULL,
  NULL,
  NULL}
}
```

Definition at line 30 of file external/mechinfo.c.

7.43 mechinfo.c File Reference

```
#include "x-gssapi.h"
```

Variables

- [Gsasl_mechanism gsasl_gssapi_mechanism](#)

7.43.1 Variable Documentation

7.43.1.1 [Gsasl_mechanism gsasl_gssapi_mechanism](#)

Definition at line 30 of file gssapi/mechinfo.c.

7.44 mechinfo.c File Reference

```
#include "login.h"
```

Variables

- [Gsasl_mechanism gsasl_login_mechanism](#)

7.44.1 Variable Documentation

7.44.1.1 [Gsasl_mechanism gsasl_login_mechanism](#)

Definition at line 30 of file login/mechinfo.c.

7.45 mechinfo.c File Reference

```
#include "x-ntlm.h"
```

Variables

- [Gsasl_mechanism gsasl_ntlm_mechanism](#)

7.45.1 Variable Documentation

7.45.1.1 [Gsasl_mechanism gsasl_ntlm_mechanism](#)

Definition at line 30 of file ntlm/mechinfo.c.

7.46 mechinfo.c File Reference

```
#include "plain.h"
```

Variables

- [Gsasl_mechanism gsasl_plain_mechanism](#)

7.46.1 Variable Documentation

7.46.1.1 [Gsasl_mechanism gsasl_plain_mechanism](#)

Initial value:

```
{
  GSASL_PLAIN_NAME,
  {
    NULL,
    NULL,
    NULL,

    NULL,

    NULL,
    NULL,
    NULL}
,
{
  NULL,
  NULL,
  NULL,

  NULL,

  NULL,
  NULL,
  NULL}
}
```

Definition at line 30 of file plain/mechinfo.c.

7.47 mechinfo.c File Reference

```
#include "securid.h"
```

Variables

- [Gsasl_mechanism gsasl_securid_mechanism](#)

7.47.1 Variable Documentation

7.47.1.1 [Gsasl_mechanism gsasl_securid_mechanism](#)

Definition at line 30 of file securid/mechinfo.c.

7.48 ntlm.c File Reference

```
#include <stdlib.h>
#include <string.h>
#include "x-ntlm.h"
#include <ntlm.h>
```

Data Structures

- struct [_Gsasl_ntlm_state](#)

Typedefs

- typedef [_Gsasl_ntlm_state](#) [_Gsasl_ntlm_state](#)

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)

7.48.1 Typedef Documentation

7.48.1.1 typedef struct [_Gsasl_ntlm_state](#) [_Gsasl_ntlm_state](#)

Definition at line 42 of file ntlm.c.

7.48.2 Function Documentation

7.48.2.1 void [_gsasl_ntlm_client_finish](#) ([Gsasl_session](#) * *sctx*, void * *mech_data*)

Definition at line 135 of file ntlm.c.

7.48.2.2 int [_gsasl_ntlm_client_start](#) ([Gsasl_session](#) * *sctx*, void ** *mech_data*)

Definition at line 45 of file ntlm.c.

References [GSASL_MALLOC_ERROR](#), and [GSASL_OK](#).

7.48.2.3 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 61 of file ntlm.c.

References GSASL_AUTHID, GSASL_MALLOC_ERROR, GSASL_MECHANISM_CALLED_TOO_MANY_TIMES, GSASL_MECHANISM_PARSE_ERROR, GSASL_NEEDS_MORE, GSASL_NO_AUTHID, GSASL_NO_PASSWORD, GSASL_OK, GSASL_PASSWORD, gsasl_property_get(), GSASL_REALM, and _Gsasl_ntlm_state::step.

7.49 obsolete.c File Reference

```
#include "internal.h"
#include <ctype.h>
#include <stdio.h>
```

Defines

- #define `Assert(Cond)` if (!(Cond)) abort()

Functions

- int `gsasl_client_listmech` (Gsasl *ctx, char *out, size_t *outlen)
- int `gsasl_server_listmech` (Gsasl *ctx, char *out, size_t *outlen)
- int `gsasl_client_step` (Gsasl_session *sctx, const char *input, size_t input_len, char *output, size_t *output_len)
- int `gsasl_server_step` (Gsasl_session *sctx, const char *input, size_t input_len, char *output, size_t *output_len)
- int `gsasl_client_step_base64` (Gsasl_session *sctx, const char *b64input, char *b64output, size_t b64output_len)
- int `gsasl_server_step_base64` (Gsasl_session *sctx, const char *b64input, char *b64output, size_t b64output_len)
- void `gsasl_client_finish` (Gsasl_session *sctx)
- void `gsasl_server_finish` (Gsasl_session *sctx)
- Gsasl * `gsasl_client_ctx_get` (Gsasl_session *sctx)
- void `gsasl_client_application_data_set` (Gsasl_session *sctx, void *application_data)
- void * `gsasl_client_application_data_get` (Gsasl_session *sctx)
- Gsasl * `gsasl_server_ctx_get` (Gsasl_session *sctx)
- void `gsasl_server_application_data_set` (Gsasl_session *sctx, void *application_data)
- void * `gsasl_server_application_data_get` (Gsasl_session *sctx)
- int `gsasl_randomize` (int strong, char *data, size_t datalen)
- Gsasl * `gsasl_ctx_get` (Gsasl_session *sctx)
- int `gsasl_encode_inline` (Gsasl_session *sctx, const char *input, size_t input_len, char *output, size_t *output_len)
- int `gsasl_decode_inline` (Gsasl_session *sctx, const char *input, size_t input_len, char *output, size_t *output_len)
- void `gsasl_application_data_set` (Gsasl *ctx, void *appdata)
- void * `gsasl_application_data_get` (Gsasl *ctx)
- void `gsasl_appinfo_set` (Gsasl_session *sctx, void *appdata)
- void * `gsasl_appinfo_get` (Gsasl_session *sctx)
- const char * `gsasl_server_suggest_mechanism` (Gsasl *ctx, const char *mechlist)
- void `gsasl_client_callback_authentication_id_set` (Gsasl *ctx, Gsasl_client_callback_authentication_id cb)
- Gsasl_client_callback_authentication_id `gsasl_client_callback_authentication_id_get` (Gsasl *ctx)
- void `gsasl_client_callback_authorization_id_set` (Gsasl *ctx, Gsasl_client_callback_authorization_id cb)
- Gsasl_client_callback_authorization_id `gsasl_client_callback_authorization_id_get` (Gsasl *ctx)
- void `gsasl_client_callback_password_set` (Gsasl *ctx, Gsasl_client_callback_password cb)
- Gsasl_client_callback_password `gsasl_client_callback_password_get` (Gsasl *ctx)

- void `gsasl_client_callback_passcode_set` (Gsasl *ctx, Gsasl_client_callback_passcode cb)
- `Gsasl_client_callback_passcode` `gsasl_client_callback_passcode_get` (Gsasl *ctx)
- void `gsasl_client_callback_pin_set` (Gsasl *ctx, Gsasl_client_callback_pin cb)
- `Gsasl_client_callback_pin` `gsasl_client_callback_pin_get` (Gsasl *ctx)
- void `gsasl_client_callback_service_set` (Gsasl *ctx, Gsasl_client_callback_service cb)
- `Gsasl_client_callback_service` `gsasl_client_callback_service_get` (Gsasl *ctx)
- void `gsasl_client_callback_anonymous_set` (Gsasl *ctx, Gsasl_client_callback_anonymous cb)
- `Gsasl_client_callback_anonymous` `gsasl_client_callback_anonymous_get` (Gsasl *ctx)
- void `gsasl_client_callback_qop_set` (Gsasl *ctx, Gsasl_client_callback_qop cb)
- `Gsasl_client_callback_qop` `gsasl_client_callback_qop_get` (Gsasl *ctx)
- void `gsasl_client_callback_maxbuf_set` (Gsasl *ctx, Gsasl_client_callback_maxbuf cb)
- `Gsasl_client_callback_maxbuf` `gsasl_client_callback_maxbuf_get` (Gsasl *ctx)
- void `gsasl_client_callback_realm_set` (Gsasl *ctx, Gsasl_client_callback_realm cb)
- `Gsasl_client_callback_realm` `gsasl_client_callback_realm_get` (Gsasl *ctx)
- void `gsasl_server_callback_validate_set` (Gsasl *ctx, Gsasl_server_callback_validate cb)
- `Gsasl_server_callback_validate` `gsasl_server_callback_validate_get` (Gsasl *ctx)
- void `gsasl_server_callback_retrieve_set` (Gsasl *ctx, Gsasl_server_callback_retrieve cb)
- `Gsasl_server_callback_retrieve` `gsasl_server_callback_retrieve_get` (Gsasl *ctx)
- void `gsasl_server_callback_cram_md5_set` (Gsasl *ctx, Gsasl_server_callback_cram_md5 cb)
- `Gsasl_server_callback_cram_md5` `gsasl_server_callback_cram_md5_get` (Gsasl *ctx)
- void `gsasl_server_callback_digest_md5_set` (Gsasl *ctx, Gsasl_server_callback_digest_md5 cb)
- `Gsasl_server_callback_digest_md5` `gsasl_server_callback_digest_md5_get` (Gsasl *ctx)
- void `gsasl_server_callback_external_set` (Gsasl *ctx, Gsasl_server_callback_external cb)
- `Gsasl_server_callback_external` `gsasl_server_callback_external_get` (Gsasl *ctx)
- void `gsasl_server_callback_anonymous_set` (Gsasl *ctx, Gsasl_server_callback_anonymous cb)
- `Gsasl_server_callback_anonymous` `gsasl_server_callback_anonymous_get` (Gsasl *ctx)
- void `gsasl_server_callback_realm_set` (Gsasl *ctx, Gsasl_server_callback_realm cb)
- `Gsasl_server_callback_realm` `gsasl_server_callback_realm_get` (Gsasl *ctx)
- void `gsasl_server_callback_qop_set` (Gsasl *ctx, Gsasl_server_callback_qop cb)
- `Gsasl_server_callback_qop` `gsasl_server_callback_qop_get` (Gsasl *ctx)
- void `gsasl_server_callback_maxbuf_set` (Gsasl *ctx, Gsasl_server_callback_maxbuf cb)
- `Gsasl_server_callback_maxbuf` `gsasl_server_callback_maxbuf_get` (Gsasl *ctx)
- void `gsasl_server_callback_cipher_set` (Gsasl *ctx, Gsasl_server_callback_cipher cb)
- `Gsasl_server_callback_cipher` `gsasl_server_callback_cipher_get` (Gsasl *ctx)
- void `gsasl_server_callback_securid_set` (Gsasl *ctx, Gsasl_server_callback_securid cb)
- `Gsasl_server_callback_securid` `gsasl_server_callback_securid_get` (Gsasl *ctx)
- void `gsasl_server_callback_gssapi_set` (Gsasl *ctx, Gsasl_server_callback_gssapi cb)
- `Gsasl_server_callback_gssapi` `gsasl_server_callback_gssapi_get` (Gsasl *ctx)
- void `gsasl_server_callback_service_set` (Gsasl *ctx, Gsasl_server_callback_service cb)
- `Gsasl_server_callback_service` `gsasl_server_callback_service_get` (Gsasl *ctx)
- char * `gsasl_stringprep_nfkc` (const char *in, ssize_t len)
- char * `gsasl_stringprep_saslprep` (const char *in, int *stringprep_rc)
- char * `gsasl_stringprep_trace` (const char *in, int *stringprep_rc)
- int `gsasl_md5pwd_get_password` (const char *filename, const char *username, char *key, size_t *keylen)
- int `gsasl_base64_encode` (char const *src, size_t srclength, char *target, size_t targsize)
- int `gsasl_base64_decode` (char const *src, char *target, size_t targsize)

7.49.1 Define Documentation

7.49.1.1 #define Assert(Cond) if (!(Cond)) abort()

Definition at line 1784 of file obsolete.c.

Referenced by `gsasl_base64_encode()`.

7.49.2 Function Documentation

7.49.2.1 void* gsasl_appinfo_get (Gsasl_session * sctx)

`gsasl_appinfo_get`:

Parameters:

sctx libgsasl session handle.

Retrieve application specific data from libgsasl session handle. The application data is set using `gsasl_appinfo_set()`. It is normally used by the application to maintain state between the main program and the callback.

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

Deprecated: Use `gsasl_callback_hook_get()` instead.

Definition at line 594 of file obsolete.c.

References `Gsasl_session::application_data`.

Referenced by `gsasl_client_application_data_get()`, and `gsasl_server_application_data_get()`.

7.49.2.2 void gsasl_appinfo_set (Gsasl_session * sctx, void * appdata)

`gsasl_appinfo_set`:

Parameters:

sctx libgsasl session handle.

appdata 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_appinfo_get()`. It is normally used by the application to maintain state between the main program and the callback.

Deprecated: Use `gsasl_callback_hook_set()` instead.

Definition at line 575 of file obsolete.c.

References `Gsasl_session::application_data`.

Referenced by `gsasl_client_application_data_set()`, and `gsasl_server_application_data_set()`.

7.49.2.3 void* gsasl_application_data_get (Gsasl * ctx)

`gsasl_application_data_get`:

Parameters:

ctx libgsasl handle.

Retrieve application specific data from libgsasl handle. The application data is set using [gsasl_application_data_set\(\)](#). It is normally used by the application to maintain state between the main program and the callback.

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

Deprecated: Use [gsasl_callback_hook_get\(\)](#) instead.

Definition at line 556 of file obsolete.c.

References Gsasl::application_hook.

7.49.2.4 void gsasl_application_data_set (Gsasl * ctx, void * appdata)

gsasl_application_data_set:

Parameters:

ctx libgsasl handle.

appdata 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_application_data_get\(\)](#). It is normally used by the application to maintain state between the main program and the callback.

Deprecated: Use [gsasl_callback_hook_set\(\)](#) instead.

Definition at line 537 of file obsolete.c.

References Gsasl::application_hook.

7.49.2.5 int gsasl_base64_decode (char const * src, char * target, size_t targsize)

gsasl_base64_decode:

Parameters:

src input byte array

target output byte array

targsize size of output byte array

Decode Base64 data. Skips all whitespace anywhere. Converts characters, four at a time, starting at (or after) *src* from Base64 numbers into three 8 bit bytes in the *target* area.

Return value: Returns the number of data bytes stored at the target, or -1 on error.

Deprecated: Use [gsasl_base64_from\(\)](#) instead.

Definition at line 1886 of file obsolete.c.

7.49.2.6 int gsasl_base64_encode (char const * src, size_t srclength, char * target, size_t targsize)

gsasl_base64_encode:

Parameters:

src input byte array
srclength size of input byte array
target output byte array
targsize size of output byte array

Encode data as base64. Converts characters, three at a time, starting at *src* into four base64 characters in the *target* area until the entire input buffer is encoded.

Return value: Returns the number of data bytes stored at the *target*, or -1 on error.

Deprecated: Use [gsasl_base64_to\(\)](#) instead.

Definition at line 1807 of file `obsolete.c`.

References `Assert`.

7.49.2.7 void* gsasl_client_application_data_get (Gsasl_session * sctx)

`gsasl_client_application_data_get`:

Parameters:

sctx libgsasl client handle.

Retrieve application specific data from libgsasl client handle. The application data is set using [gsasl_client_application_data_set\(\)](#). It is normally used by the application to maintain state between the main program and the callback.

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

Deprecated: Use [gsasl_callback_hook_get\(\)](#) or [gsasl_session_hook_get\(\)](#) instead.

Definition at line 348 of file `obsolete.c`.

References `gsasl_appinfo_get()`.

7.49.2.8 void gsasl_client_application_data_set (Gsasl_session * sctx, void * application_data)

`gsasl_client_application_data_set`:

Parameters:

sctx libgsasl client handle.
application_data opaque pointer to application specific data.

Store application specific data in the libgsasl client handle. The application data can be later (for instance, inside a callback) be retrieved by calling [gsasl_client_application_data_get\(\)](#). It is normally used by the application to maintain state between the main program and the callback.

Deprecated: Use [gsasl_callback_hook_set\(\)](#) or [gsasl_session_hook_set\(\)](#) instead.

Definition at line 327 of file `obsolete.c`.

References `gsasl_appinfo_set()`.

7.49.2.9 [Gsasl_client_callback_anonymous](#) [gsasl_client_callback_anonymous_get](#) ([Gsasl](#) * *ctx*)

[gsasl_client_callback_anonymous_get](#):

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_client_callback_anonymous_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 895 of file [obsolete.c](#).

References [Gsasl::cbc_anonymous](#).

Referenced by [gsasl_property_get\(\)](#).

7.49.2.10 [void gsasl_client_callback_anonymous_set](#) ([Gsasl](#) * *ctx*, [Gsasl_client_callback_anonymous](#) *cb*)

[gsasl_client_callback_anonymous_set](#):

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the client to set the anonymous token, which usually is the users email address. The function can be later retrieved using [gsasl_client_callback_anonymous_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 876 of file [obsolete.c](#).

References [Gsasl::cbc_anonymous](#).

7.49.2.11 [Gsasl_client_callback_authentication_id](#) [gsasl_client_callback_authentication_id_get](#) ([Gsasl](#) * *ctx*)

[gsasl_client_callback_authentication_id_get](#):

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_client_callback_authentication_id_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 652 of file [obsolete.c](#).

References [Gsasl::cbc_authentication_id](#).

Referenced by [_gsasl_kerberos_v5_client_step\(\)](#), and [gsasl_property_get\(\)](#).

7.49.2.12 void gsasl_client_callback_authentication_id_set (Gsasl * ctx, Gsasl_client_callback_authentication_id cb)

gsasl_client_callback_authentication_id_set:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the client to set the authentication identity. The function can be later retrieved using [gsasl_client_callback_authentication_id_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 632 of file obsolete.c.

References Gsasl::cbc_authentication_id.

7.49.2.13 Gsasl_client_callback_authorization_id gsasl_client_callback_authorization_id_get (Gsasl * ctx)

gsasl_client_callback_authorization_id_get:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_client_callback_authorization_id_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 692 of file obsolete.c.

References Gsasl::cbc_authorization_id.

Referenced by [_gsasl_kerberos_v5_client_step\(\)](#), and [gsasl_property_get\(\)](#).

7.49.2.14 void gsasl_client_callback_authorization_id_set (Gsasl * ctx, Gsasl_client_callback_authorization_id cb)

gsasl_client_callback_authorization_id_set:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the client to set the authorization identity. The function can be later retrieved using [gsasl_client_callback_authorization_id_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 672 of file obsolete.c.

References Gsasl::cbc_authorization_id.

7.49.2.15 **Gsasl_client_callback_maxbuf** `gsasl_client_callback_maxbuf_get (Gsasl * ctx)`

`gsasl_client_callback_maxbuf_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling `gsasl_client_callback_maxbuf_set()`.

Deprecated: This function is part of the old callback interface. The new interface uses `gsasl_callback_set()` to set the application callback, and uses `gsasl_callback()` or `gsasl_property_get()` to invoke the callback for certain properties.

Definition at line 975 of file obsolete.c.

References Gsasl::cbc_maxbuf.

Referenced by `_gsasl_kerberos_v5_client_step()`.

7.49.2.16 **void gsasl_client_callback_maxbuf_set (Gsasl * ctx, Gsasl_client_callback_maxbuf cb)**

`gsasl_client_callback_maxbuf_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the client to inform the server of the largest buffer the client is able to receive when using the DIGEST-MD5 "auth-int" or "auth-conf" Quality of Protection (qop). If this directive is missing, the default value 65536 will be assumed. The function can be later retrieved using `gsasl_client_callback_maxbuf_get()`.

Deprecated: This function is part of the old callback interface. The new interface uses `gsasl_callback_set()` to set the application callback, and uses `gsasl_callback()` or `gsasl_property_get()` to invoke the callback for certain properties.

Definition at line 956 of file obsolete.c.

References Gsasl::cbc_maxbuf.

7.49.2.17 **Gsasl_client_callback_passcode** `gsasl_client_callback_passcode_get (Gsasl * ctx)`

`gsasl_client_callback_passcode_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling `gsasl_client_callback_passcode_set()`.

Deprecated: This function is part of the old callback interface. The new interface uses `gsasl_callback_set()` to set the application callback, and uses `gsasl_callback()` or `gsasl_property_get()` to invoke the callback for certain properties.

Definition at line 772 of file obsolete.c.

References Gsasl::cbc_passcode.

Referenced by gsasl_property_get().

7.49.2.18 void gsasl_client_callback_passcode_set (Gsasl * ctx, Gsasl_client_callback_passcode cb)

gsasl_client_callback_passcode_set:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the client to set the passcode. The function can be later retrieved using [gsasl_client_callback_passcode_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 752 of file obsolete.c.

References Gsasl::cbc_passcode.

7.49.2.19 Gsasl_client_callback_password gsasl_client_callback_password_get (Gsasl * ctx)

gsasl_client_callback_password_get:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_client_callback_password_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 732 of file obsolete.c.

References Gsasl::cbc_password.

Referenced by [_gsasl_kerberos_v5_client_step\(\)](#), and [gsasl_property_get\(\)](#).

7.49.2.20 void gsasl_client_callback_password_set (Gsasl * ctx, Gsasl_client_callback_password cb)

gsasl_client_callback_password_set:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the client to set the password. The function can be later retrieved using [gsasl_client_callback_password_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 712 of file `obsolete.c`.

References `Gsasl::cbc_password`.

7.49.2.21 [Gsasl_client_callback_pin](#) [gsasl_client_callback_pin_get](#) (**Gsasl** * *ctx*)

`gsasl_client_callback_pin_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_client_callback_pin_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 813 of file `obsolete.c`.

References `Gsasl::cbc_pin`.

Referenced by [gsasl_property_get\(\)](#).

7.49.2.22 **void** [gsasl_client_callback_pin_set](#) (**Gsasl** * *ctx*, [Gsasl_client_callback_pin](#) *cb*)

`gsasl_client_callback_pin_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the client to chose a new pin, possibly suggested by the server, for the SECURID mechanism. This is not normally invoked, but only when the server requests it. The function can be later retrieved using [gsasl_client_callback_pin_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 794 of file `obsolete.c`.

References `Gsasl::cbc_pin`.

7.49.2.23 [Gsasl_client_callback_qop](#) [gsasl_client_callback_qop_get](#) (**Gsasl** * *ctx*)

`gsasl_client_callback_qop_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_client_callback_qop_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 933 of file `obsolete.c`.

References `Gsasl::cbc_qop`.

Referenced by `_gsasl_kerberos_v5_client_step()`.

7.49.2.24 `void gsasl_client_callback_qop_set (Gsasl * ctx, Gsasl_client_callback_qop cb)`

`gsasl_client_callback_qop_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the client to determine the qop to use after looking at what the server offered. The function can be later retrieved using [gsasl_client_callback_qop_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 915 of file `obsolete.c`.

References `Gsasl::cbc_qop`.

7.49.2.25 `Gsasl_client_callback_realm gsasl_client_callback_realm_get (Gsasl * ctx)`

`gsasl_client_callback_realm_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_client_callback_realm_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1014 of file `obsolete.c`.

References `Gsasl::cbc_realm`.

Referenced by `_gsasl_kerberos_v5_client_step()`, and `gsasl_property_get()`.

7.49.2.26 `void gsasl_client_callback_realm_set (Gsasl * ctx, Gsasl_client_callback_realm cb)`

`gsasl_client_callback_realm_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the client to know which realm it belongs to. The realm is used by the server to determine which username and password to use. The function can be later retrieved using [gsasl_client_callback_realm_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 996 of file obsolete.c.

References `Gsasl::cbc_realm`.

7.49.2.27 [Gsasl_client_callback_service](#) [gsasl_client_callback_service_get](#) (`Gsasl * ctx`)

[gsasl_client_callback_service_get](#):

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_client_callback_service_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 855 of file obsolete.c.

References `Gsasl::cbc_service`.

Referenced by [_gsasl_kerberos_v5_client_step\(\)](#), and [gsasl_property_get\(\)](#).

7.49.2.28 `void` [gsasl_client_callback_service_set](#) (`Gsasl * ctx`, [Gsasl_client_callback_service](#) *cb*)

[gsasl_client_callback_service_set](#):

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the client to set the name of the service. The service buffer should be a registered GSSAPI host-based service name, hostname the name of the server. Servicename is used by DIGEST-MD5 and should be the name of generic server in case of a replicated service. The function can be later retrieved using [gsasl_client_callback_service_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 836 of file obsolete.c.

References `Gsasl::cbc_service`.

7.49.2.29 `Gsasl*` [gsasl_client_ctx_get](#) (`Gsasl_session * sctx`)

[gsasl_client_ctx_get](#):

Parameters:

sctx libgsasl client handle

Return value: Returns the libgsasl handle given a libgsasl client handle.

Deprecated: This function is not useful with the new 0.2.0 API.

Definition at line 307 of file obsolete.c.

References Gsasl_session::ctx.

Referenced by _gsasl_kerberos_v5_client_step().

7.49.2.30 void gsasl_client_finish (Gsasl_session * sctx)

gsasl_client_finish:

Parameters:

sctx libgsasl client handle.

Destroy a libgsasl client handle. The handle must not be used with other libgsasl functions after this call.

Deprecated: Use [gsasl_finish\(\)](#) instead.

Definition at line 278 of file obsolete.c.

References gsasl_finish().

7.49.2.31 int gsasl_client_listmech (Gsasl * ctx, char * out, size_t * outlen)

gsasl_client_listmech:

Parameters:

ctx libgsasl handle.

out output character array.

outlen input maximum size of output character array, on output contains actual length of output array.

Write SASL names, separated by space, of mechanisms supported by the libgsasl client to the output array. To find out how large the output array must be, call this function with out=NULL.

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

Deprecated: Use [gsasl_client_mechlist\(\)](#) instead.

Definition at line 41 of file obsolete.c.

References gsasl_client_mechlist(), GSASL_OK, and GSASL_TOO_SMALL_BUFFER.

7.49.2.32 int gsasl_client_step (Gsasl_session * sctx, const char * input, size_t input_len, char * output, size_t * output_len)

gsasl_client_step:

Parameters:

sctx libgsasl client handle.

input input byte array.

input_len size of input byte array.

output output byte array.

output_len size of output byte array.

Perform one step of SASL authentication in client. This reads data from server (specified with *input* and *input_len*), processes it (potentially invoking callbacks to the application), and writes data to server (into variables *output* and *output_len*).

The contents of the output buffer is unspecified if this functions returns anything other than GSASL_NEEDS_MORE.

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

Deprecated: Use [gsasl_step\(\)](#) instead.

Definition at line 161 of file `obsolete.c`.

7.49.2.33 `int gsasl_client_step_base64 (Gsasl_session * sctx, const char * b64input, char * b64output, size_t b64output_len)`

`gsasl_client_step_base64`:

Parameters:

sctx libgsasl client handle.

b64input input base64 encoded byte array.

b64output output base64 encoded byte array.

b64output_len size of output base64 encoded byte array.

This is a simple wrapper around [gsasl_client_step\(\)](#) that base64 decodes the input and base64 encodes the output.

Return value: See [gsasl_client_step\(\)](#).

Deprecated: Use [gsasl_step64\(\)](#) instead.

Definition at line 239 of file `obsolete.c`.

7.49.2.34 `Gsasl* gsasl_ctx_get (Gsasl_session * sctx)`

`gsasl_ctx_get`:

Parameters:

sctx libgsasl session handle

Return value: Returns the libgsasl handle given a libgsasl session handle.

Deprecated: This function is not useful with the new 0.2.0 API.

Definition at line 438 of file `obsolete.c`.

References `Gsasl_session::ctx`.

7.49.2.35 `int gsasl_decode_inline (Gsasl_session * sctx, const char * input, size_t input_len, char * output, size_t * output_len)`

gsasl_decode_inline:

Parameters:

- sctx* libgsasl session handle.
- input* input byte array.
- input_len* size of input byte array.
- output* output byte array.
- output_len* size of output byte array.

Decode data according to negotiated SASL mechanism. This might mean that data is integrity or privacy protected.

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

Deprecated: Use [gsasl_decode\(\)](#) instead.

Since: 0.2.0

Definition at line 502 of file obsolete.c.

References [gsasl_decode\(\)](#), [GSASL_OK](#), and [GSASL_TOO_SMALL_BUFFER](#).

7.49.2.36 `int gsasl_encode_inline (Gsasl_session * sctx, const char * input, size_t input_len, char * output, size_t * output_len)`

gsasl_encode_inline:

Parameters:

- sctx* libgsasl session handle.
- input* input byte array.
- input_len* size of input byte array.
- output* output byte array.
- output_len* size of output byte array.

Encode data according to negotiated SASL mechanism. This might mean that data is integrity or privacy protected.

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

Deprecated: Use [gsasl_encode\(\)](#) instead.

Since: 0.2.0

Definition at line 462 of file obsolete.c.

References [gsasl_encode\(\)](#), [GSASL_OK](#), and [GSASL_TOO_SMALL_BUFFER](#).

7.49.2.37 `int gsasl_md5pwd_get_password (const char * filename, const char * username, char * key, size_t * keylen)`

gsasl_md5pwd_get_password:

Parameters:

filename filename of file containing passwords.

username username string.

key output character array.

keylen input maximum size of output character array, on output contains actual length of output array.

Retrieve password for user from specified file. To find out how large the output array must be, call this function with out=NULL.

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.

Deprecated: Use [gsasl_simple_getpass\(\)](#) instead.

Definition at line 1685 of file obsolete.c.

References GSASL_AUTHENTICATION_ERROR, GSASL_FCLOSE_ERROR, GSASL_FOPEN_ERROR, GSASL_OK, and GSASL_TOO_SMALL_BUFFER.

7.49.2.38 int gsasl_randomize (int *strong*, char * *data*, size_t *datalen*)

gsasl_randomize:

Parameters:

strong 0 iff operation should not block, non-0 for very strong randomness.

data output array to be filled with random data.

datalen size of output array.

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

Return value: Returns GSASL_OK iff successful.

Deprecated: Use [gsasl_random\(\)](#) or [gsasl_nonce\(\)](#) instead.

Definition at line 422 of file obsolete.c.

References [gsasl_nonce\(\)](#), and [gsasl_random\(\)](#).

7.49.2.39 void* gsasl_server_application_data_get (Gsasl_session * *sctx*)

gsasl_server_application_data_get:

Parameters:

sctx libgsasl server handle.

Retrieve application specific data from libgsasl server handle. The application data is set using [gsasl_server_application_data_set\(\)](#). It is normally used by the application to maintain state between the main program and the callback.

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

Deprecated: Use [gsasl_callback_hook_get\(\)](#) or [gsasl_session_hook_get\(\)](#) instead.

Definition at line 403 of file `obsolete.c`.

References [gsasl_appinfo_get\(\)](#).

7.49.2.40 void [gsasl_server_application_data_set](#) ([Gsasl_session](#) * *sctx*, void * *application_data*)

[gsasl_server_application_data_set](#):

Parameters:

sctx libgsasl server handle.

application_data opaque pointer to application specific data.

Store application specific data in the libgsasl server handle. The application data can be later (for instance, inside a callback) be retrieved by calling [gsasl_server_application_data_get\(\)](#). It is normally used by the application to maintain state between the main program and the callback.

Deprecated: Use [gsasl_callback_hook_set\(\)](#) or [gsasl_session_hook_set\(\)](#) instead.

Definition at line 382 of file `obsolete.c`.

References [gsasl_appinfo_set\(\)](#).

7.49.2.41 [Gsasl_server_callback_anonymous](#) [gsasl_server_callback_anonymous_get](#) ([Gsasl](#) * *ctx*)

[gsasl_server_callback_anonymous_get](#):

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_server_callback_anonymous_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1252 of file `obsolete.c`.

References `Gsasl::cbs_anonymous`.

Referenced by [gsasl_callback\(\)](#).

7.49.2.42 void [gsasl_server_callback_anonymous_set](#) ([Gsasl](#) * *ctx*, [Gsasl_server_callback_anonymous_cb](#))

[gsasl_server_callback_anonymous_set](#):

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the server for deciding if user is permitted anonymous access. The function can be later retrieved using [gsasl_server_callback_anonymous_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1233 of file `obsolete.c`.

References `Gsasl::cbs_anonymous`.

7.49.2.43 [Gsasl_server_callback_cipher](#) `gsasl_server_callback_cipher_get (Gsasl * ctx)`

`gsasl_server_callback_cipher_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_server_callback_cipher_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1414 of file `obsolete.c`.

References `Gsasl::cbs_cipher`.

7.49.2.44 `void gsasl_server_callback_cipher_set (Gsasl * ctx, Gsasl_server_callback_cipher cb)`

`gsasl_server_callback_cipher_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the server to inform the client of the cipher suites supported. The DES and 3DES ciphers must be supported for interoperability. It is currently used by the DIGEST-MD5 mechanism. The function can be later retrieved using [gsasl_server_callback_cipher_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1395 of file `obsolete.c`.

References `Gsasl::cbs_cipher`.

7.49.2.45 [Gsasl_server_callback_cram_md5](#) `gsasl_server_callback_cram_md5_get (Gsasl * ctx)`

`gsasl_server_callback_cram_md5_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_server_callback_cram_md5_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1134 of file `obsolete.c`.

References `Gsasl::cbs_cram_md5`.

7.49.2.46 `void gsasl_server_callback_cram_md5_set (Gsasl * ctx, Gsasl_server_callback_cram_md5 cb)`

`gsasl_server_callback_cram_md5_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the server for deciding if user is authenticated using CRAM-MD5 challenge and response. The function can be later retrieved using [gsasl_server_callback_cram_md5_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1115 of file `obsolete.c`.

References `Gsasl::cbs_cram_md5`.

7.49.2.47 `Gsasl_server_callback_digest_md5 gsasl_server_callback_digest_md5_get (Gsasl * ctx)`

`gsasl_server_callback_digest_md5_get`:

Parameters:

ctx libgsasl handle.

Return value: Return the callback earlier set by calling [gsasl_server_callback_digest_md5_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1174 of file `obsolete.c`.

References `Gsasl::cbs_digest_md5`.

7.49.2.48 `void gsasl_server_callback_digest_md5_set (Gsasl * ctx, Gsasl_server_callback_digest_md5 cb)`

`gsasl_server_callback_digest_md5_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the server for retrieving the secret hash of the username, realm and password for use in the DIGEST-MD5 mechanism. The function can be later retrieved using [gsasl_server_callback_digest_md5_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1155 of file `obsolete.c`.

References `Gsasl::cbs_digest_md5`.

7.49.2.49 [Gsasl_server_callback_external](#) [gsasl_server_callback_external_get](#) (`Gsasl * ctx`)

`gsasl_server_callback_external_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_server_callback_external_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1213 of file `obsolete.c`.

References `Gsasl::cbs_external`.

Referenced by [gsasl_callback\(\)](#).

7.49.2.50 `void` [gsasl_server_callback_external_set](#) (`Gsasl * ctx`, [Gsasl_server_callback_external cb](#))

`gsasl_server_callback_external_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the server for deciding if user is authenticated out of band. The function can be later retrieved using [gsasl_server_callback_external_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1194 of file `obsolete.c`.

References `Gsasl::cbs_external`.

7.49.2.51 [Gsasl_server_callback_gssapi](#) [gsasl_server_callback_gssapi_get](#) (`Gsasl * ctx`)

`gsasl_server_callback_gssapi_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_server_callback_gssapi_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1502 of file `obsolete.c`.

References `Gsasl::cbs_gssapi`.

Referenced by [gsasl_callback\(\)](#).

7.49.2.52 `void gsasl_server_callback_gssapi_set (Gsasl * ctx, Gsasl_server_callback_gssapi cb)`

`gsasl_server_callback_gssapi_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the server for checking if a GSSAPI user is authorized for username (by, e.g., calling `krb5_userok()`). The function should return `GSASL_OK` if the user should be permitted access, or an error code such as `GSASL_AUTHENTICATION_ERROR` on failure. The function can be later retrieved using [gsasl_server_callback_gssapi_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1483 of file `obsolete.c`.

References `Gsasl::cbs_gssapi`.

7.49.2.53 `Gsasl_server_callback_maxbuf gsasl_server_callback_maxbuf_get (Gsasl * ctx)`

`gsasl_server_callback_maxbuf_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_server_callback_maxbuf_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1373 of file `obsolete.c`.

References `Gsasl::cbs_maxbuf`.

Referenced by [_gsasl_kerberos_v5_server_step\(\)](#).

7.49.2.54 `void gsasl_server_callback_maxbuf_set (Gsasl * ctx, Gsasl_server_callback_maxbuf cb)`

`gsasl_server_callback_maxbuf_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the server to inform the client of the largest buffer the server is able to receive when using the DIGEST-MD5 "auth-int" or "auth-conf" Quality of Protection (qop). If this directive is missing, the default value 65536 will be assumed. The function can be later retrieved using [gsasl_server_callback_maxbuf_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1354 of file obsolete.c.

References Gsasl::cbs_maxbuf.

7.49.2.55 [Gsasl_server_callback_qop](#) [gsasl_server_callback_qop_get](#) ([Gsasl](#) * *ctx*)

[gsasl_server_callback_qop_get](#):

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_server_callback_qop_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1331 of file obsolete.c.

References Gsasl::cbs_qop.

Referenced by [_gsasl_kerberos_v5_server_step\(\)](#).

7.49.2.56 [void gsasl_server_callback_qop_set](#) ([Gsasl](#) * *ctx*, [Gsasl_server_callback_qop](#) *cb*)

[gsasl_server_callback_qop_set](#):

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the server to know which quality of protection it accepts. The quality of protection eventually used is selected by the client though. It is currently used by the DIGEST-MD5 mechanism. The function can be later retrieved using [gsasl_server_callback_qop_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1313 of file obsolete.c.

References Gsasl::cbs_qop.

7.49.2.57 Gsasl_server_callback_realm `gsasl_server_callback_realm_get (Gsasl * ctx)`

`gsasl_server_callback_realm_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling `gsasl_server_callback_realm_set()`.

Deprecated: This function is part of the old callback interface. The new interface uses `gsasl_callback_set()` to set the application callback, and uses `gsasl_callback()` or `gsasl_property_get()` to invoke the callback for certain properties.

Definition at line 1291 of file `obsolete.c`.

References `Gsasl::cbs_realm`.

Referenced by `_gsasl_kerberos_v5_server_step()`.

7.49.2.58 void gsasl_server_callback_realm_set (Gsasl * ctx, Gsasl_server_callback_realm cb)

`gsasl_server_callback_realm_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the server to know which realm it serves. The realm is used by the user to determine which username and password to use. The function can be later retrieved using `gsasl_server_callback_realm_get()`.

Deprecated: This function is part of the old callback interface. The new interface uses `gsasl_callback_set()` to set the application callback, and uses `gsasl_callback()` or `gsasl_property_get()` to invoke the callback for certain properties.

Definition at line 1273 of file `obsolete.c`.

References `Gsasl::cbs_realm`.

7.49.2.59 Gsasl_server_callback_retrieve `gsasl_server_callback_retrieve_get (Gsasl * ctx)`

`gsasl_server_callback_retrieve_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling `gsasl_server_callback_retrieve_set()`.

Deprecated: This function is part of the old callback interface. The new interface uses `gsasl_callback_set()` to set the application callback, and uses `gsasl_callback()` or `gsasl_property_get()` to invoke the callback for certain properties.

Definition at line 1094 of file `obsolete.c`.

References `Gsasl::cbs_retrieve`.

Referenced by `_gsasl_kerberos_v5_server_step()`, and `gsasl_callback()`.

7.49.2.60 void gsasl_server_callback_retrieve_set (Gsasl * ctx, Gsasl_server_callback_retrieve cb)

gsasl_server_callback_retrieve_set:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the server for deciding if user is authenticated using authentication identity, authorization identity and password. The function can be later retrieved using [gsasl_server_callback_retrieve_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1075 of file obsolete.c.

References Gsasl::cbs_retrieve.

7.49.2.61 Gsasl_server_callback_securid gsasl_server_callback_securid_get (Gsasl * ctx)

gsasl_server_callback_securid_get:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_server_callback_securid_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1460 of file obsolete.c.

References Gsasl::cbs_securid.

Referenced by [gsasl_callback\(\)](#).

7.49.2.62 void gsasl_server_callback_securid_set (Gsasl * ctx, Gsasl_server_callback_securid cb)

gsasl_server_callback_securid_set:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the server for validating a user via the SECURID mechanism. The function should return GSASL_OK if user authenticated successfully, GSASL_SECURID_SERVER_NEED_ADDITIONAL_PASSCODE if it wants another passcode, GSASL_SECURID_SERVER_NEED_NEW_PIN if it wants a PIN change, or an error. When (and only when) GSASL_SECURID_SERVER_NEED_NEW_PIN is returned, suggestpin can be populated with a PIN code the server suggests, and suggestpinlen set to the length of the PIN. The function can be later retrieved using [gsasl_server_callback_securid_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1441 of file `obsolete.c`.

References `Gsasl::cbs_securid`.

7.49.2.63 [Gsasl_server_callback_service](#) [gsasl_server_callback_service_get](#) (`Gsasl * ctx`)

`gsasl_server_callback_service_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_server_callback_service_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1543 of file `obsolete.c`.

References `Gsasl::cbs_service`.

Referenced by `_gsasl_kerberos_v5_server_step()`.

7.49.2.64 `void` [gsasl_server_callback_service_set](#) (`Gsasl * ctx`, [Gsasl_server_callback_service cb](#))

`gsasl_server_callback_service_set`:

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the server to set the name of the service. The service buffer should be a registered GSSAPI host-based service name, hostname the name of the server. The function can be later retrieved using [gsasl_server_callback_service_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1524 of file `obsolete.c`.

References `Gsasl::cbs_service`.

7.49.2.65 [Gsasl_server_callback_validate](#) [gsasl_server_callback_validate_get](#) (`Gsasl * ctx`)

`gsasl_server_callback_validate_get`:

Parameters:

ctx libgsasl handle.

Return value: Returns the callback earlier set by calling [gsasl_server_callback_validate_set\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1054 of file `obsolete.c`.

References `Gsasl::cbs_validate`.

Referenced by [gsasl_callback\(\)](#).

7.49.2.66 void [gsasl_server_callback_validate_set](#) ([Gsasl](#) * *ctx*, [Gsasl_server_callback_validate](#) *cb*)

[gsasl_server_callback_validate_set](#):

Parameters:

ctx libgsasl handle.

cb callback function

Specify the callback function to use in the server for deciding if user is authenticated using authentication identity, authorization identity and password. The function can be later retrieved using [gsasl_server_callback_validate_get\(\)](#).

Deprecated: This function is part of the old callback interface. The new interface uses [gsasl_callback_set\(\)](#) to set the application callback, and uses [gsasl_callback\(\)](#) or [gsasl_property_get\(\)](#) to invoke the callback for certain properties.

Definition at line 1035 of file `obsolete.c`.

References `Gsasl::cbs_validate`.

7.49.2.67 [Gsasl](#)* [gsasl_server_ctx_get](#) ([Gsasl_session](#) * *sctx*)

[gsasl_server_ctx_get](#):

Parameters:

sctx libgsasl server handle

Return value: Returns the libgsasl handle given a libgsasl server handle.

Deprecated: This function is not useful with the new 0.2.0 API.

Definition at line 362 of file `obsolete.c`.

References `Gsasl_session::ctx`.

Referenced by [_gsasl_kerberos_v5_server_step\(\)](#).

7.49.2.68 void [gsasl_server_finish](#) ([Gsasl_session](#) * *sctx*)

[gsasl_server_finish](#):

Parameters:

sctx libgsasl server handle.

Destroy a libgsasl server handle. The handle must not be used with other libgsasl functions after this call.

Deprecated: Use [gsasl_finish\(\)](#) instead.

Definition at line 293 of file `obsolete.c`.

References [gsasl_finish\(\)](#).

7.49.2.69 `int gsasl_server_listmech (Gsasl * ctx, char * out, size_t * outlen)`

`gsasl_server_listmech`:

Parameters:

ctx libgsasl handle.

out output character array.

outlen input maximum size of output character array, on output contains actual length of output array.

Write SASL names, separated by space, of mechanisms supported by the libgsasl server to the output array. To find out how large the output array must be, call this function with `out=NULL`.

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

Deprecated: Use [gsasl_server_mechlist\(\)](#) instead.

Definition at line 83 of file `obsolete.c`.

References `GSASL_OK`, [gsasl_server_mechlist\(\)](#), and `GSASL_TOO_SMALL_BUFFER`.

7.49.2.70 `int gsasl_server_step (Gsasl_session * sctx, const char * input, size_t input_len, char * output, size_t * output_len)`

`gsasl_server_step`:

Parameters:

sctx libgsasl server handle.

input input byte array.

input_len size of input byte array.

output output byte array.

output_len size of output byte array.

Perform one step of SASL authentication in server. This reads data from client (specified with `input` and `input_len`), processes it (potentially invoking callbacks to the application), and writes data to client (into variables `output` and `output_len`).

The contents of the output buffer is unspecified if this functions returns anything other than `GSASL_NEEDS_MORE`.

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

Deprecated: Use [gsasl_step\(\)](#) instead.

Definition at line 191 of file `obsolete.c`.

7.49.2.71 `int gsasl_server_step_base64 (Gsasl_session * sctx, const char * b64input, char * b64output, size_t b64output_len)`

gsasl_server_step_base64:

Parameters:

- sctx* libgsasl server handle.
- b64input* input base64 encoded byte array.
- b64output* output base64 encoded byte array.
- b64output_len* size of output base64 encoded byte array.

This is a simple wrapper around `gsasl_server_step()` that base64 decodes the input and base64 encodes the output.

Return value: See `gsasl_server_step()`.

Deprecated: Use `gsasl_step64()` instead.

Definition at line 261 of file `obsolete.c`.

7.49.2.72 `const char* gsasl_server_suggest_mechanism (Gsasl * ctx, const char * meclist)`

gsasl_server_suggest_mechanism:

Parameters:

- ctx* libgsasl handle.
- meclist* input character array with SASL mechanism names, separated by invalid characters (e.g. SPC).

Return value: Returns name of "best" SASL mechanism supported by the libgsasl server which is present in the input string.

Deprecated: This function was never useful, since it is the client that chose which mechanism to use.

Definition at line 612 of file `obsolete.c`.

7.49.2.73 `char* gsasl_stringprep_nfkc (const char * in, ssize_t len)`

gsasl_stringprep_nfkc:

Parameters:

- in* a UTF-8 encoded string.
- len* length of *in*, in bytes, or -1 if *in* is nul-terminated.

Converts a string into canonical form, standardizing such issues as whether a character with an accent is represented as a base character and combining accent or as a single precomposed character.

The normalization mode is NFKC (ALL COMPOSE). It standardizes differences that do not affect the text content, such as the above-mentioned accent representation. It standardizes the "compatibility" characters in Unicode, such as SUPERSCRIPT THREE to the standard forms (in this case DIGIT THREE). Formatting information may be lost but for most text operations such characters should be considered the same. It returns a result with composed forms rather than a maximally decomposed form.

Return value: Return a newly allocated string, that is the NFKC normalized form of , o NULL on error.

Deprecated: No replacement functionality in GNU SASL, use GNU Libidn instead. Note that in SASL, you most likely want to use SASLprep and not bare NFKC, see [gsasl_saslprep\(\)](#).

Definition at line 1579 of file obsolete.c.

7.49.2.74 `char* gsasl_stringprep_saslprep (const char * in, int * stringprep_rc)`

`gsasl_stringprep_saslprep`:

Parameters:

in input ASCII or UTF-8 string with data to prepare according to SASLprep.

stringprep_rc pointer to output variable with stringprep error code, or NULL to indicate that you don't care about it.

Process a Unicode string for comparison, according to the "SASLprep" stringprep profile. This function is intended to be used by Simple Authentication and Security Layer (SASL) mechanisms (such as PLAIN, CRAM-MD5, and DIGEST-MD5) as well as other protocols exchanging user names and/or passwords.

Return value: Return a newly allocated string that is the "SASLprep" processed form of the input string, or NULL on error, in which case contain the stringprep library error code.

Deprecated: Use [gsasl_saslprep\(\)](#) instead.

Definition at line 1610 of file obsolete.c.

7.49.2.75 `char* gsasl_stringprep_trace (const char * in, int * stringprep_rc)`

`gsasl_stringprep_trace`:

Parameters:

in input ASCII or UTF-8 string with data to prepare according to "trace".

stringprep_rc pointer to output variable with stringprep error code, or NULL to indicate that you don't care about it.

Process a Unicode string for use as trace information, according to the "trace" stringprep profile. The profile is designed for use with the SASL ANONYMOUS Mechanism.

Return value: Return a newly allocated string that is the "trace" processed form of the input string, or NULL on error, in which case contain the stringprep library error code.

Deprecated: No replacement functionality in GNU SASL, use GNU Libidn instead.

Definition at line 1644 of file obsolete.c.

7.50 parser.c File Reference

```
#include "parser.h"
#include <stdlib.h>
#include <string.h>
#include "validate.h"
```

Defines

- #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)

7.50.1 Define Documentation

7.50.1.1 #define [DEFAULT_ALGORITHM](#) "md5-sess"

Definition at line 40 of file parser.c.

7.50.1.2 #define [DEFAULT_CHARSET](#) "utf-8"

Definition at line 39 of file parser.c.

7.50.2 Enumeration Type Documentation

7.50.2.1 anonymous enum

Enumerator:

CHALLENGE_REALM
CHALLENGE_NONCE
CHALLENGE_QOP
CHALLENGE_STALE
CHALLENGE_MAXBUF
CHALLENGE_CHARSET
CHALLENGE_ALGORITHM
CHALLENGE_CIPHER

Definition at line 42 of file parser.c.

7.50.2.2 anonymous enum

Enumerator:

QOP_AUTH
QOP_AUTH_INT
QOP_AUTH_CONF

Definition at line 69 of file parser.c.

7.50.2.3 anonymous enum

Enumerator:

CIPHER_DES
CIPHER_3DES
CIPHER_RC4
CIPHER_RC4_40
CIPHER_RC4_56
CIPHER_AES_CBC

Definition at line 89 of file parser.c.

7.50.2.4 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 315 of file parser.c.

7.50.2.5 anonymous enum

Enumerator:

RESPONSEAUTH_RSPAUTH

Definition at line 521 of file parser.c.

7.50.3 Function Documentation

7.50.3.1 `int digest_md5_parse_challenge (const char * challenge, size_t len, digest_md5_challenge * out)`

Definition at line 569 of file parser.c.

Referenced by `_gsasl_digest_md5_client_step()`, and `main()`.

7.50.3.2 `int digest_md5_parse_finish (const char * finish, size_t len, digest_md5_finish * out)`

Definition at line 611 of file parser.c.

Referenced by `_gsasl_digest_md5_client_step()`, and `main()`.

7.50.3.3 `int digest_md5_parse_response (const char * response, size_t len, digest_md5_response * out)`

Definition at line 590 of file parser.c.

Referenced by `_gsasl_digest_md5_server_step()`, and `main()`.

7.51 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)

7.51.1 Function Documentation

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

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

7.51.1.2 int [digest_md5_parse_challenge](#) (const char * *challenge*, size_t *len*, [digest_md5_challenge](#) * *out*)

Definition at line 569 of file `parser.c`.

Referenced by `_gsasl_digest_md5_client_step()`, and `main()`.

7.51.1.3 int [digest_md5_parse_finish](#) (const char * *finish*, size_t *len*, [digest_md5_finish](#) * *out*)

Definition at line 611 of file `parser.c`.

Referenced by `_gsasl_digest_md5_client_step()`, and `main()`.

7.51.1.4 int [digest_md5_parse_response](#) (const char * *response*, size_t *len*, [digest_md5_response](#) * *out*)

Definition at line 590 of file `parser.c`.

Referenced by `_gsasl_digest_md5_server_step()`, and `main()`.

7.52 plain.h File Reference

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

Defines

- #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](#)

7.52.1 Define Documentation

7.52.1.1 #define GSASL_PLAIN_NAME "PLAIN"

Definition at line 28 of file plain.h.

7.52.2 Function Documentation

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

Definition at line 37 of file plain/client.c.

References [GSASL_AUTHID](#), [GSASL_AUTHZID](#), [GSASL_MALLOC_ERROR](#), [GSASL_NO_AUTHID](#), [GSASL_NO_PASSWORD](#), [GSASL_OK](#), [GSASL_PASSWORD](#), and [gsasl_property_get\(\)](#).

7.52.2.2 int _gsasl_plain_server_step ([Gsasl_session](#) * sctx, void * mech_data, const char * input, size_t input_len, char ** output, size_t * output_len)

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

References [GSASL_ALLOW_UNASSIGNED](#), [GSASL_AUTHENTICATION_ERROR](#), [GSASL_AUTHID](#), [GSASL_AUTHZID](#), [gsasl_callback\(\)](#), [GSASL_MALLOC_ERROR](#), [GSASL_MECHANISM_PARSE_ERROR](#), [GSASL_NEEDS_MORE](#), [GSASL_NO_CALLBACK](#), [GSASL_NO_PASSWORD](#), [GSASL_OK](#), [GSASL_PASSWORD](#), [gsasl_property_get\(\)](#), [gsasl_property_set\(\)](#), [gsasl_saslprep\(\)](#), and [GSASL_VALIDATE_SIMPLE](#).

7.52.3 Variable Documentation

7.52.3.1 [Gsasl_mechanism](#) [gsasl_plain_mechanism](#)

Definition at line 30 of file plain/mechinfo.c.

7.53 printer.c File Reference

```
#include "printer.h"
#include <stdlib.h>
#include <vasprintf.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)

7.53.1 Function Documentation

7.53.1.1 char* [digest_md5_print_challenge](#) ([digest_md5_challenge](#) * c)

Definition at line 43 of file printer.c.

References [DIGEST_MD5_CIPHER_3DES](#), [DIGEST_MD5_CIPHER_AES_CBC](#), [DIGEST_MD5_CIPHER_DES](#), [DIGEST_MD5_CIPHER_RC4](#), [DIGEST_MD5_CIPHER_RC4_40](#), [DIGEST_MD5_CIPHER_RC4_56](#), [DIGEST_MD5_QOP_AUTH](#), [DIGEST_MD5_QOP_AUTH_CONF](#), [DIGEST_MD5_QOP_AUTH_INT](#), and [digest_md5_validate_challenge\(\)](#).

Referenced by [_gsasl_digest_md5_server_step\(\)](#), and [main\(\)](#).

7.53.1.2 char* [digest_md5_print_finish](#) ([digest_md5_finish](#) * finish)

Definition at line 172 of file printer.c.

References [digest_md5_validate_finish\(\)](#).

Referenced by [_gsasl_digest_md5_server_step\(\)](#).

7.53.1.3 char* [digest_md5_print_response](#) ([digest_md5_response](#) * r)

Definition at line 102 of file printer.c.

References [DIGEST_MD5_CIPHER_3DES](#), [DIGEST_MD5_CIPHER_AES_CBC](#), [DIGEST_MD5_CIPHER_DES](#), [DIGEST_MD5_CIPHER_RC4](#), [DIGEST_MD5_CIPHER_RC4_40](#), [DIGEST_MD5_CIPHER_RC4_56](#), [DIGEST_MD5_QOP_AUTH](#), [DIGEST_MD5_QOP_AUTH_CONF](#), [DIGEST_MD5_QOP_AUTH_INT](#), and [digest_md5_validate_response\(\)](#).

Referenced by [_gsasl_digest_md5_client_step\(\)](#), and [main\(\)](#).

7.54 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)

7.54.1 Function Documentation

7.54.1.1 char* [digest_md5_print_challenge](#) ([digest_md5_challenge](#) * *challenge*)

Definition at line 43 of file printer.c.

References [DIGEST_MD5_CIPHER_3DES](#), [DIGEST_MD5_CIPHER_AES_CBC](#), [DIGEST_MD5_CIPHER_DES](#), [DIGEST_MD5_CIPHER_RC4](#), [DIGEST_MD5_CIPHER_RC4_40](#), [DIGEST_MD5_CIPHER_RC4_56](#), [DIGEST_MD5_QOP_AUTH](#), [DIGEST_MD5_QOP_AUTH_CONF](#), [DIGEST_MD5_QOP_AUTH_INT](#), and [digest_md5_validate_challenge\(\)](#).

Referenced by [_gsasl_digest_md5_server_step\(\)](#), and [main\(\)](#).

7.54.1.2 char* [digest_md5_print_finish](#) ([digest_md5_finish](#) * *out*)

Definition at line 172 of file printer.c.

References [digest_md5_validate_finish\(\)](#).

Referenced by [_gsasl_digest_md5_server_step\(\)](#).

7.54.1.3 char* [digest_md5_print_response](#) ([digest_md5_response](#) * *response*)

Definition at line 102 of file printer.c.

References [DIGEST_MD5_CIPHER_3DES](#), [DIGEST_MD5_CIPHER_AES_CBC](#), [DIGEST_MD5_CIPHER_DES](#), [DIGEST_MD5_CIPHER_RC4](#), [DIGEST_MD5_CIPHER_RC4_40](#), [DIGEST_MD5_CIPHER_RC4_56](#), [DIGEST_MD5_QOP_AUTH](#), [DIGEST_MD5_QOP_AUTH_CONF](#), [DIGEST_MD5_QOP_AUTH_INT](#), and [digest_md5_validate_response\(\)](#).

Referenced by [_gsasl_digest_md5_client_step\(\)](#), and [main\(\)](#).

7.55 property.c File Reference

```
#include "internal.h"
```

Functions

- void [gsasl_property_set](#) ([Gsasl_session](#) *sctx, [Gsasl_property](#) prop, const char *data)
- void [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)

7.55.1 Function Documentation

7.55.1.1 const char* [gsasl_property_fast](#) ([Gsasl_session](#) * *sctx*, [Gsasl_property](#) *prop*)

[gsasl_property_fast](#):

Parameters:

sctx session handle.

prop enumerated value of [Gsasl_property](#) type, indicating the type of data in .

Retrieve the data stored in the session handle for given property .

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 171 of file [property.c](#).

Referenced by [_gsasl_digest_md5_client_step\(\)](#), and [gsasl_property_get\(\)](#).

7.55.1.2 const char* [gsasl_property_get](#) ([Gsasl_session](#) * *sctx*, [Gsasl_property](#) *prop*)

[gsasl_property_get](#):

Parameters:

sctx session handle.

prop enumerated value of [Gsasl_property](#) type, indicating the type of data in .

Retrieve the data stored in the session handle for given property , 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.

If no value is known, and no callback is specified or if the callback fail to return data, and if any obsolete callback functions has been set by the application, this function will try to call these obsolete callbacks,

and store the returned data as the corresponding property. This behaviour of this function will be removed when the obsolete callback interfaces are removed.

Return value: Return data for property, or NULL if no value known.

Since: 0.2.0

Definition at line 208 of file property.c.

References Gsasl_session::ctx, GSASL_ANONYMOUS_TOKEN, GSASL_AUTHID, GSASL_AUTHZID, gsasl_callback(), gsasl_client_callback_anonymous_get(), gsasl_client_callback_authentication_id_get(), gsasl_client_callback_authorization_id_get(), gsasl_client_callback_passcode_get(), gsasl_client_callback_password_get(), gsasl_client_callback_pin_get(), gsasl_client_callback_realm_get(), gsasl_client_callback_service_get(), GSASL_HOSTNAME, GSASL_OK, GSASL_PASSCODE, GSASL_PASSWORD, GSASL_PIN, gsasl_property_fast(), gsasl_property_set(), GSASL_REALM, GSASL_SERVICE, and Gsasl_session::suggestedpin.

Referenced by _gsasl_anonymous_client_step(), _gsasl_cram_md5_client_step(), _gsasl_cram_md5_server_step(), _gsasl_digest_md5_client_step(), _gsasl_digest_md5_server_step(), _gsasl_external_client_step(), _gsasl_gssapi_client_step(), _gsasl_gssapi_server_start(), _gsasl_login_client_step(), _gsasl_login_server_step(), _gsasl_ntlm_client_step(), _gsasl_plain_client_step(), _gsasl_plain_server_step(), _gsasl_securid_client_step(), and _gsasl_securid_server_step().

7.55.1.3 void gsasl_property_set (Gsasl_session * sctx, Gsasl_property prop, const char * data)

gsasl_property_set:

Parameters:

sctx session handle.

prop enumerated value of Gsasl_property type, indicating the type of data in .

data zero terminated character string to store.

Make a copy of and store it in the session handle for the indicated property .

You can immediately deallocate after calling this function, without affecting the data stored in the session handle.

Since: 0.2.0

Definition at line 102 of file property.c.

References gsasl_property_set_raw().

Referenced by _gsasl_cram_md5_server_step(), _gsasl_digest_md5_client_step(), _gsasl_digest_md5_server_step(), _gsasl_external_server_step(), _gsasl_login_server_step(), _gsasl_plain_server_step(), _gsasl_securid_server_step(), gsasl_callback(), and gsasl_property_get().

7.55.1.4 void gsasl_property_set_raw (Gsasl_session * sctx, Gsasl_property prop, const char * data, size_t len)

gsasl_property_set_raw:

Parameters:

sctx session handle.

prop enumerated value of Gsasl_property type, indicating the type of data in .

data character string to store.

len length of character string to store.

Make a copy of *sized* and store a zero terminated version of it in the session handle for the indicated property .

You can immediately deallocate 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`.

Since: 0.2.0

Definition at line 128 of file `property.c`.

Referenced by `_gsasl_anonymous_server_step()`, `_gsasl_external_server_step()`, `_gsasl_gssapi_server_step()`, `_gsasl_secured_client_step()`, `gsasl_callback()`, and `gsasl_property_set()`.

7.56 register.c File Reference

```
#include "internal.h"
```

Functions

- `int gsasl_register (Gsasl *ctx, const Gsasl_mechanism *mech)`

7.56.1 Function Documentation

7.56.1.1 `int gsasl_register (Gsasl * ctx, const Gsasl_mechanism * mech)`

`gsasl_register`:

Parameters:

ctx pointer to libgsasl handle.

mech 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 38 of file register.c.

References `Gsasl_mechanism::client`, `Gsasl::client_mechs`, `GSASL_MALLOC_ERROR`, `GSASL_OK`, `Gsasl_mechanism_functions::init`, `Gsasl::n_client_mechs`, `Gsasl::n_server_mechs`, `Gsasl_mechanism::server`, and `Gsasl::server_mechs`.

7.57 saslprep.c File Reference

```
#include "internal.h"
```

Functions

- int `gsasl_saslprep` (const char *in, `Gsasl_saslprep_flags` flags, char **out, int *stringprepc)

7.57.1 Function Documentation

7.57.1.1 int `gsasl_saslprep` (const char * *in*, `Gsasl_saslprep_flags` *flags*, char ** *out*, int * *stringprepc*)

`gsasl_saslprep` - prepare internationalized string

Parameters:

in a UTF-8 encoded string.

flags any SASLprep flag, e.g., `GSASL_ALLOW_UNASSIGNED`.

out on exit, contains newly allocated output string.

stringprepc if non-NULL, will hold precise stringprep return code.

Prepare string using SASLprep. On success, the variable must be deallocated by the caller.

Return value: Returns `GSASL_OK` on success, or `GSASL_SASLPREP_ERROR` on error.

Since: 0.2.3

Definition at line 48 of file `saslprep.c`.

References `GSASL_ALLOW_UNASSIGNED`, `GSASL_MALLOC_ERROR`, `GSASL_OK`, and `GSASL_SASLPREP_ERROR`.

Referenced by `_gsasl_cram_md5_client_step()`, `_gsasl_cram_md5_server_step()`, and `_gsasl_plain_server_step()`.

7.58 securid.h File Reference

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

Defines

- #define [GSASL_SECURID_NAME](#) "SECURID"

Functions

- int [_gsasl_secuid_client_start](#) ([Gsasl_session](#) *sctx, void **mech_data)
- int [_gsasl_secuid_client_step](#) ([Gsasl_session](#) *sctx, void *mech_data, const char *input, size_t input_len, char **output, size_t *output_len)
- void [_gsasl_secuid_client_finish](#) ([Gsasl_session](#) *sctx, void *mech_data)
- int [_gsasl_secuid_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_secuid_mechanism](#)

7.58.1 Define Documentation

7.58.1.1 #define GSASL_SECURID_NAME "SECURID"

Definition at line 28 of file securid.h.

7.58.2 Function Documentation

7.58.2.1 void _gsasl_secuid_client_finish ([Gsasl_session](#) * sctx, void * mech_data)

Definition at line 163 of file securid/client.c.

References [_Gsasl_login_client_state::step](#).

7.58.2.2 int _gsasl_secuid_client_start ([Gsasl_session](#) * sctx, void ** mech_data)

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

References [GSASL_MALLOC_ERROR](#), [GSASL_OK](#), and [_Gsasl_login_client_state::step](#).

7.58.2.3 int _gsasl_secuid_client_step ([Gsasl_session](#) * sctx, void * mech_data, const char * input, size_t input_len, char ** output, size_t * output_len)

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

References [GSASL_AUTHID](#), [GSASL_AUTHZID](#), [GSASL_MALLOC_ERROR](#), [GSASL_MECHANISM_CALLED_TOO_MANY_TIMES](#), [GSASL_NO_AUTHID](#), [GSASL_NO_PASSCODE](#), [GSASL_NO_PIN](#), [GSASL_OK](#), [GSASL_PASSCODE](#), [GSASL_PIN](#), [gsasl_property_get\(\)](#), [gsasl_property_set_raw\(\)](#), [GSASL_SUGGESTED_PIN](#), [PASSCODE](#), [PIN](#), and [_Gsasl_login_client_state::step](#).

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

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

References GSASL_AUTHID, GSASL_AUTHZID, gsasl_callback(), GSASL_MALLOC_ERROR, GSASL_MECHANISM_PARSE_ERROR, GSASL_NEEDS_MORE, GSASL_PASSCODE, GSASL_PIN, gsasl_property_get(), gsasl_property_set(), GSASL_SECURID_SERVER_NEED_ADDITIONAL_PASSCODE, GSASL_SECURID_SERVER_NEED_NEW_PIN, GSASL_SUGGESTED_PIN, GSASL_VALIDATE_SECURID, PASSCODE, and PIN.

7.58.3 Variable Documentation

7.58.3.1 `Gsasl_mechanism gsasl_securig_mechanism`

Definition at line 30 of file securid/mechinfo.c.

7.59 server.c File Reference

```
#include "anonymous.h"
```

Functions

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

7.59.1 Function Documentation

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

Definition at line 31 of file anonymous/server.c.

References GSASL_ANONYMOUS_TOKEN, gsasl_callback(), GSASL_MECHANISM_PARSE_ERROR, GSASL_NEEDS_MORE, gsasl_property_set_raw(), and GSASL_VALIDATE_ANONYMOUS.

7.60 server.c File Reference

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

Defines

- #define [MD5LEN](#) 16

Functions

- 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)

7.60.1 Define Documentation

7.60.1.1 #define MD5LEN 16

Definition at line 42 of file `cram-md5/server.c`.

Referenced by [_gsasl_cram_md5_server_step\(\)](#), [digest_md5_decode\(\)](#), and [digest_md5_encode\(\)](#).

7.60.2 Function Documentation

7.60.2.1 void [_gsasl_cram_md5_server_finish](#) ([Gsasl_session](#) *sctx, void *mech_data)

Definition at line 124 of file `cram-md5/server.c`.

7.60.2.2 int [_gsasl_cram_md5_server_start](#) ([Gsasl_session](#) *sctx, void **mech_data)

Definition at line 45 of file `cram-md5/server.c`.

References [cram_md5_challenge\(\)](#), [CRAM_MD5_CHALLENGE_LEN](#), [GSASL_MALLOC_ERROR](#), and [GSASL_OK](#).

7.60.2.3 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 61 of file `cram-md5/server.c`.

References [cram_md5_digest\(\)](#), [CRAM_MD5_DIGEST_LEN](#), [GSASL_AUTHENTICATION_ERROR](#), [GSASL_AUTHID](#), [GSASL_MALLOC_ERROR](#), [GSASL_MECHANISM_PARSE_ERROR](#), [GSASL_NEEDS_MORE](#), [GSASL_NO_PASSWORD](#), [GSASL_OK](#), [GSASL_PASSWORD](#), [gsasl_property_get\(\)](#), [gsasl_property_set\(\)](#), [gsasl_saslprep\(\)](#), and [MD5LEN](#).

7.61 server.c File Reference

```
#include "digest-md5.h"
#include <stdlib.h>
#include <string.h>
#include "tokens.h"
#include "parser.h"
#include "printer.h"
#include "free.h"
#include "session.h"
#include "digesthmac.h"
#include "validate.h"
```

Data Structures

- struct [_Gsasl_digest_md5_server_state](#)

Defines

- #define [NONCE_ENTROPY_BYTES](#) 16

Typedefs

- typedef [_Gsasl_digest_md5_server_state](#) [_Gsasl_digest_md5_server_state](#)

Functions

- 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)

7.61.1 Define Documentation

7.61.1.1 #define NONCE_ENTROPY_BYTES 16

Definition at line 45 of file digest-md5/server.c.

Referenced by [_gsasl_digest_md5_server_start\(\)](#).

7.61.2 Typedef Documentation

7.61.2.1 typedef struct [_Gsasl_digest_md5_server_state](#) [_Gsasl_digest_md5_server_state](#)

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

7.61.3 Function Documentation

7.61.3.1 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*)

Definition at line 272 of file digest-md5/server.c.

References [digest_md5_decode\(\)](#), [GSASL_INTEGRITY_ERROR](#), [GSASL_NEEDS_MORE](#), [GSASL_OK](#), [_Gsasl_digest_md5_server_state::kic](#), [digest_md5_response::qop](#), [_Gsasl_digest_md5_server_state::readseqnum](#), and [_Gsasl_digest_md5_server_state::response](#).

7.61.3.2 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*)

Definition at line 248 of file digest-md5/server.c.

References [digest_md5_encode\(\)](#), [GSASL_INTEGRITY_ERROR](#), [GSASL_NEEDS_MORE](#), [GSASL_OK](#), [_Gsasl_digest_md5_server_state::kis](#), [digest_md5_response::qop](#), [_Gsasl_digest_md5_server_state::response](#), and [_Gsasl_digest_md5_server_state::sendseqnum](#).

7.61.3.3 void [_gsasl_digest_md5_server_finish](#) ([Gsasl_session](#) * *sctx*, void * *mech_data*)

Definition at line 233 of file digest-md5/server.c.

References [_Gsasl_digest_md5_server_state::challenge](#), [digest_md5_free_challenge\(\)](#), [digest_md5_free_finish\(\)](#), [digest_md5_free_response\(\)](#), [_Gsasl_digest_md5_server_state::finish](#), and [_Gsasl_digest_md5_server_state::response](#).

7.61.3.4 int [_gsasl_digest_md5_server_start](#) ([Gsasl_session](#) * *sctx*, void ** *mech_data*)

Definition at line 63 of file digest-md5/server.c.

References [DIGEST_MD5_QOP_AUTH](#), [DIGEST_MD5_QOP_AUTH_INT](#), [gsasl_base64_to\(\)](#), [GSASL_MALLOC_ERROR](#), [gsasl_nonce\(\)](#), [GSASL_OK](#), and [NONCE_ENTROPY_BYTES](#).

7.61.3.5 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 97 of file digest-md5/server.c.

References [digest_md5_response::authzid](#), [_Gsasl_digest_md5_server_state::challenge](#), [digest_md5_response::cipher](#), [digest_md5_response::cnonce](#), [digest_md5_hmac\(\)](#), [DIGEST_MD5_LENGTH](#), [digest_md5_parse_response\(\)](#), [digest_md5_print_challenge\(\)](#), [digest_md5_print_finish\(\)](#), [DIGEST_MD5_RESPONSE_LENGTH](#), [digest_md5_validate\(\)](#), [digest_md5_response::digesturi](#), [_Gsasl_digest_md5_server_state::finish](#), [GSASL_AUTHENTICATION_ERROR](#), [GSASL_AUTHID](#), [GSASL_AUTHZID](#), [GSASL_MALLOC_ERROR](#), [gsasl_md5\(\)](#), [GSASL_MECHANISM_CALLED_TOO_](#)

MANY_TIMES, GSASL_MECHANISM_PARSE_ERROR, GSASL_NEEDS_MORE, GSASL_NO_PASSWORD, GSASL_OK, GSASL_PASSWORD, gsasl_property_get(), gsasl_property_set(), GSASL_REALM, digest_md5_response::nc, digest_md5_response::nonce, digest_md5_challenge::nrealms, digest_md5_response::qop, digest_md5_response::realm, digest_md5_challenge::realms, digest_md5_response::response, _Gsasl_digest_md5_server_state::response, digest_md5_finish::rspauth, _Gsasl_digest_md5_server_state::secret, _Gsasl_digest_md5_server_state::step, and digest_md5_response::username.

7.62 server.c File Reference

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

Functions

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

7.62.1 Function Documentation

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

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

References GSASL_AUTHZID, gsasl_callback(), GSASL_MECHANISM_PARSE_ERROR, GSASL_NEEDS_MORE, gsasl_property_set(), gsasl_property_set_raw(), and GSASL_VALIDATE_EXTERNAL.

7.63 server.c File Reference

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

Data Structures

- struct [_Gsasl_gssapi_server_state](#)

Typedefs

- typedef [_Gsasl_gssapi_server_state](#) [_Gsasl_gssapi_server_state](#)

Functions

- 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)

7.63.1 Typedef Documentation

7.63.1.1 typedef struct [_Gsasl_gssapi_server_state](#) [_Gsasl_gssapi_server_state](#)

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

7.63.2 Function Documentation

7.63.2.1 void [_gsasl_gssapi_server_finish](#) ([Gsasl_session](#) * sctx, void * mech_data)

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

References [_Gsasl_gssapi_server_state::client](#), [_Gsasl_gssapi_server_state::context](#), and [_Gsasl_gssapi_server_state::cred](#).

7.63.2.2 int [_gsasl_gssapi_server_start](#) ([Gsasl_session](#) * sctx, void ** mech_data)

Definition at line 59 of file gssapi/server.c.

References [GSASL_GSSAPI_ACQUIRE_CRED_ERROR](#), [GSASL_GSSAPI_IMPORT_NAME_ERROR](#), [GSASL_HOSTNAME](#), [GSASL_MALLOC_ERROR](#), [GSASL_NO_HOSTNAME](#), [GSASL_NO_SERVICE](#), [GSASL_OK](#), [gsasl_property_get\(\)](#), and [GSASL_SERVICE](#).

7.63.2.3 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 121 of file gssapi/server.c.

References `_Gsasl_gssapi_server_state::client`, `_Gsasl_gssapi_server_state::context`, `_Gsasl_gssapi_server_state::cred`, `GSASL_AUTHZID`, `gsasl_callback()`, `GSASL_GSSAPI_ACCEPT_SEC_CONTEXT_ERROR`, `GSASL_GSSAPI_DISPLAY_NAME`, `GSASL_GSSAPI_DISPLAY_NAME_ERROR`, `GSASL_GSSAPI_RELEASE_BUFFER_ERROR`, `GSASL_GSSAPI_UNSUPPORTED_PROTECTION_ERROR`, `GSASL_GSSAPI_UNWRAP_ERROR`, `GSASL_GSSAPI_WRAP_ERROR`, `GSASL_MALLOC_ERROR`, `GSASL_MECHANISM_CALLED_TOO_MANY_TIMES`, `GSASL_NEEDS_MORE`, `gsasl_property_set_raw()`, `GSASL_QOP_AUTH`, `GSASL_VALIDATE_GSSAPI`, and `_Gsasl_gssapi_server_state::step`.

7.64 server.c File Reference

```
#include "kerberos_v5.h"
#include "shared.h"
```

Data Structures

- [struct _Gsasl_kerberos_v5_server_state](#)

Functions

- [int _gsasl_kerberos_v5_server_init](#) ([Gsasl_ctx](#) *ctx)
- [int _gsasl_kerberos_v5_server_start](#) ([Gsasl_session](#) *sctx, void **mech_data)
- [int _gsasl_kerberos_v5_server_step](#) ([Gsasl_session](#) *sctx, void *mech_data, const char *input, size_t input_len, char *output, size_t *output_len)
- [int _gsasl_kerberos_v5_server_encode](#) ([Gsasl_session](#) *sctx, void *mech_data, const char *input, size_t input_len, char *output, size_t *output_len)
- [int _gsasl_kerberos_v5_server_decode](#) ([Gsasl_session](#) *sctx, void *mech_data, const char *input, size_t input_len, char *output, size_t *output_len)
- [int _gsasl_kerberos_v5_server_finish](#) ([Gsasl_session](#) *sctx, void *mech_data)

7.64.1 Function Documentation

7.64.1.1 [int _gsasl_kerberos_v5_server_decode](#) ([Gsasl_session](#) * sctx, void * mech_data, const char * input, size_t input_len, char * output, size_t * output_len)

Definition at line 535 of file kerberos_v5/server.c.

References [_Gsasl_kerberos_v5_server_state::clientqop](#), [GSASL_INTEGRITY_ERROR](#), [GSASL_KERBEROS_V5_INTERNAL_ERROR](#), [GSASL_MALLOC_ERROR](#), [GSASL_OK](#), [GSASL_QOP_AUTH_CONF](#), [GSASL_QOP_AUTH_INT](#), [_Gsasl_kerberos_v5_server_state::safe](#), [_Gsasl_kerberos_v5_server_state::sessionkey](#), and [_Gsasl_kerberos_v5_server_state::sh](#).

7.64.1.2 [int _gsasl_kerberos_v5_server_encode](#) ([Gsasl_session](#) * sctx, void * mech_data, const char * input, size_t input_len, char * output, size_t * output_len)

Definition at line 488 of file kerberos_v5/server.c.

References [_Gsasl_kerberos_v5_server_state::clientqop](#), [GSASL_INTEGRITY_ERROR](#), [GSASL_KERBEROS_V5_INTERNAL_ERROR](#), [GSASL_MALLOC_ERROR](#), [GSASL_OK](#), [GSASL_QOP_AUTH_CONF](#), [GSASL_QOP_AUTH_INT](#), [_Gsasl_kerberos_v5_server_state::safe](#), [_Gsasl_kerberos_v5_server_state::sessionkey](#), and [_Gsasl_kerberos_v5_server_state::sh](#).

7.64.1.3 [int _gsasl_kerberos_v5_server_finish](#) ([Gsasl_session](#) * sctx, void * mech_data)

Definition at line 586 of file kerberos_v5/server.c.

References [GSASL_OK](#), [_Gsasl_kerberos_v5_server_state::password](#), [_Gsasl_kerberos_v5_server_state::random](#), [_Gsasl_kerberos_v5_server_state::sh](#), and [_Gsasl_kerberos_v5_server_state::username](#).

7.64.1.4 `int _gsasl_kerberos_v5_server_init (Gsasl_ctx * ctx)`

Definition at line 56 of file `kerberos_v5/server.c`.

References `GSASL_OK`, and `GSASL_UNKNOWN_MECHANISM`.

7.64.1.5 `int _gsasl_kerberos_v5_server_start (Gsasl_session * sctx, void ** mech_data)`

Definition at line 65 of file `kerberos_v5/server.c`.

References `GSASL_KERBEROS_V5_INIT_ERROR`, `GSASL_KERBEROS_V5_INTERNAL_ERROR`, `GSASL_MALLOC_ERROR`, `GSASL_OK`, `GSASL_QOP_AUTH`, `GSASL_QOP_AUTH_INT`, and `RANDOM_LEN`.

7.64.1.6 `int _gsasl_kerberos_v5_server_step (Gsasl_session * sctx, void * mech_data, const char * input, size_t input_len, char * output, size_t * output_len)`

Definition at line 106 of file `kerberos_v5/server.c`.

References `_Gsasl_kerberos_v5_server_state::ap`, `_Gsasl_kerberos_v5_server_state::as`, `BITMAP_LEN`, `CLIENT_HELLO_LEN`, `_Gsasl_kerberos_v5_server_state::clientmaxbuf`, `_Gsasl_kerberos_v5_server_state::clientmutual`, `_Gsasl_kerberos_v5_server_state::clientqop`, `_Gsasl_kerberos_v5_server_state::firststep`, `GSASL_AUTHENTICATION_ERROR`, `GSASL_CANNOT_GET_CTX`, `GSASL_KERBEROS_V5_INTERNAL_ERROR`, `GSASL_MALLOC_ERROR`, `GSASL_NEED_SERVER_RETRIEVE_CALLBACK`, `GSASL_NEED_SERVER_SERVICE_CALLBACK`, `GSASL_NEEDS_MORE`, `GSASL_OK`, `GSASL_QOP_AUTH`, `GSASL_QOP_AUTH_CONF`, `GSASL_QOP_AUTH_INT`, `gsasl_server_callback_maxbuf_get()`, `gsasl_server_callback_qop_get()`, `gsasl_server_callback_realm_get()`, `gsasl_server_callback_retrieve_get()`, `gsasl_server_callback_service_get()`, `gsasl_server_ctx_get()`, `GSASL_TOO_SMALL_BUFFER`, `MAXBUF_DEFAULT`, `MAXBUF_LEN`, `MUTUAL`, `_Gsasl_kerberos_v5_server_state::password`, `_Gsasl_kerberos_v5_server_state::random`, `RANDOM_LEN`, `SERVER_HELLO_LEN`, `_Gsasl_kerberos_v5_server_state::serverhello`, `_Gsasl_kerberos_v5_server_state::serverhostname`, `_Gsasl_kerberos_v5_server_state::servermaxbuf`, `_Gsasl_kerberos_v5_server_state::serverqops`, `_Gsasl_kerberos_v5_server_state::serverrealm`, `_Gsasl_kerberos_v5_server_state::serverservice`, `_Gsasl_kerberos_v5_server_state::sessionkey`, `_Gsasl_kerberos_v5_server_state::sessiontkkey`, `_Gsasl_kerberos_v5_server_state::sh`, `_Gsasl_kerberos_v5_server_state::userkey`, `_Gsasl_kerberos_v5_server_state::username`, and `_Gsasl_kerberos_v5_server_state::userrealm`.

7.65 server.c File Reference

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

Data Structures

- struct [_Gsasl_login_server_state](#)

Defines

- #define [CHALLENGE_USERNAME](#) "User Name"
- #define [CHALLENGE_PASSWORD](#) "Password"

Functions

- 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)

7.65.1 Define Documentation

7.65.1.1 #define CHALLENGE_PASSWORD "Password"

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

Referenced by [_gsasl_login_server_step\(\)](#).

7.65.1.2 #define CHALLENGE_USERNAME "User Name"

Definition at line 43 of file login/server.c.

Referenced by [_gsasl_login_server_step\(\)](#).

7.65.2 Function Documentation

7.65.2.1 void _gsasl_login_server_finish ([Gsasl_session](#) * sctx, void * mech_data)

Definition at line 149 of file login/server.c.

References [_Gsasl_login_server_state::password](#), and [_Gsasl_login_server_state::username](#).

7.65.2.2 int _gsasl_login_server_start ([Gsasl_session](#) * sctx, void ** mech_data)

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

References [GSASL_MALLOC_ERROR](#), and [GSASL_OK](#).

7.65.2.3 `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 61 of file login/server.c.

References CHALLENGE_PASSWORD, CHALLENGE_USERNAME, GSASL_AUTHENTICATION_ERROR, GSASL_AUTHID, GSASL_AUTHZID, gsasl_callback(), GSASL_MALLOC_ERROR, GSASL_MECHANISM_CALLED_TOO_MANY_TIMES, GSASL_MECHANISM_PARSE_ERROR, GSASL_NEEDS_MORE, GSASL_NO_CALLBACK, GSASL_OK, GSASL_PASSWORD, gsasl_property_get(), gsasl_property_set(), GSASL_VALIDATE_SIMPLE, _Gsasl_login_server_state::password, _Gsasl_login_server_state::step, and _Gsasl_login_server_state::username.

7.66 server.c File Reference

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

Functions

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

7.66.1 Function Documentation

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

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

References `GSASL_ALLOW_UNASSIGNED`, `GSASL_AUTHENTICATION_ERROR`, `GSASL_AUTHID`, `GSASL_AUTHZID`, `gsasl_callback()`, `GSASL_MALLOC_ERROR`, `GSASL_MECHANISM_PARSE_ERROR`, `GSASL_NEEDS_MORE`, `GSASL_NO_CALLBACK`, `GSASL_NO_PASSWORD`, `GSASL_OK`, `GSASL_PASSWORD`, `gsasl_property_get()`, `gsasl_property_set()`, `gsasl_saslprep()`, and `GSASL_VALIDATE_SIMPLE`.

7.67 server.c File Reference

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

Defines

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

Functions

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

7.67.1 Define Documentation

7.67.1.1 #define `PASSCODE` "passcode"

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

7.67.1.2 #define `PIN` "pin"

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

7.67.2 Function Documentation

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

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

References `GSASL_AUTHID`, `GSASL_AUTHZID`, `gsasl_callback()`, `GSASL_MALLOC_ERROR`, `GSASL_MECHANISM_PARSE_ERROR`, `GSASL_NEEDS_MORE`, `GSASL_PASSCODE`, `GSASL_PIN`, `gsasl_property_get()`, `gsasl_property_set()`, `GSASL_SECURID_SERVER_NEED_ADDITIONAL_PASSCODE`, `GSASL_SECURID_SERVER_NEED_NEW_PIN`, `GSASL_SUGGESTED_PIN`, `GSASL_VALIDATE_SECURID`, `PASSCODE`, and `PIN`.

7.68 session.c File Reference

```
#include "session.h"
#include <stdlib.h>
#include <string.h>
#include <gc.h>
```

Defines

- #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])

7.68.1 Define Documentation

7.68.1.1 #define C2I(buf)

Value:

```
((buf[0] & 0xFF) |
 ((buf[1] & 0xFF) << 8) |
 ((buf[2] & 0xFF) << 16) |
 ((buf[3] & 0xFF) << 24))
```

Definition at line 116 of file session.c.

Referenced by [digest_md5_decode\(\)](#).

7.68.1.2 #define MAC_DATA_LEN 4

Definition at line 41 of file session.c.

Referenced by [digest_md5_decode\(\)](#), and [digest_md5_encode\(\)](#).

7.68.1.3 #define MAC_HMAC_LEN 10

Definition at line 42 of file session.c.

Referenced by [digest_md5_decode\(\)](#), and [digest_md5_encode\(\)](#).

7.68.1.4 #define MAC_MSG_TYPE "\x00\x01"

Definition at line 43 of file session.c.

Referenced by `digest_md5_decode()`, and `digest_md5_encode()`.

7.68.1.5 #define MAC_MSG_TYPE_LEN 2

Definition at line 44 of file session.c.

Referenced by `digest_md5_decode()`, and `digest_md5_encode()`.

7.68.1.6 #define MAC_SEQNUM_LEN 4

Definition at line 45 of file session.c.

Referenced by `digest_md5_decode()`, and `digest_md5_encode()`.

7.68.1.7 #define MD5LEN 16

Definition at line 39 of file session.c.

7.68.1.8 #define SASL_INTEGRITY_PREFIX_LENGTH 4

Definition at line 40 of file session.c.

Referenced by `digest_md5_decode()`.

7.68.2 Function Documentation**7.68.2.1 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 122 of file session.c.

References `C2I`, `DIGEST_MD5_QOP_AUTH_CONF`, `DIGEST_MD5_QOP_AUTH_INT`, `MAC_DATA_LEN`, `MAC_HMAC_LEN`, `MAC_MSG_TYPE`, `MAC_MSG_TYPE_LEN`, `MAC_SEQNUM_LEN`, `MD5LEN`, and `SASL_INTEGRITY_PREFIX_LENGTH`.

Referenced by `_gsasl_digest_md5_client_decode()`, and `_gsasl_digest_md5_server_decode()`.

7.68.2.2 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 48 of file session.c.

References `DIGEST_MD5_QOP_AUTH_CONF`, `DIGEST_MD5_QOP_AUTH_INT`, `MAC_DATA_LEN`, `MAC_HMAC_LEN`, `MAC_MSG_TYPE`, `MAC_MSG_TYPE_LEN`, `MAC_SEQNUM_LEN`, and `MD5LEN`.

Referenced by `_gsasl_digest_md5_client_encode()`, and `_gsasl_digest_md5_server_encode()`.

7.69 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])

7.69.1 Function Documentation

7.69.1.1 [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 122 of file session.c.

References C2I, DIGEST_MD5_QOP_AUTH_CONF, DIGEST_MD5_QOP_AUTH_INT, MAC_DATA_LEN, MAC_HMAC_LEN, MAC_MSG_TYPE, MAC_MSG_TYPE_LEN, MAC_SEQNUM_LEN, MD5LEN, and SASL_INTEGRITY_PREFIX_LENGTH.

Referenced by [_gsasl_digest_md5_client_decode\(\)](#), and [_gsasl_digest_md5_server_decode\(\)](#).

7.69.1.2 [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 48 of file session.c.

References DIGEST_MD5_QOP_AUTH_CONF, DIGEST_MD5_QOP_AUTH_INT, MAC_DATA_LEN, MAC_HMAC_LEN, MAC_MSG_TYPE, MAC_MSG_TYPE_LEN, MAC_SEQNUM_LEN, and MD5LEN.

Referenced by [_gsasl_digest_md5_client_encode\(\)](#), and [_gsasl_digest_md5_server_encode\(\)](#).

7.70 shared.h File Reference

```
#include "kerberos_v5.h"  
#include <shishi.h>
```

Defines

- #define [DEBUG](#) 0
- #define [BITMAP_LEN](#) 1
- #define [MAXBUF_LEN](#) 4
- #define [RANDOM_LEN](#) 16
- #define [MUTUAL](#) (1 << 3)
- #define [SERVER_HELLO_LEN](#) [BITMAP_LEN](#) + [MAXBUF_LEN](#) + [RANDOM_LEN](#)
- #define [CLIENT_HELLO_LEN](#) [BITMAP_LEN](#) + [MAXBUF_LEN](#)
- #define [MAXBUF_DEFAULT](#) 65536

7.70.1 Define Documentation

7.70.1.1 #define [BITMAP_LEN](#) 1

Definition at line 34 of file shared.h.

7.70.1.2 #define [CLIENT_HELLO_LEN](#) [BITMAP_LEN](#) + [MAXBUF_LEN](#)

Definition at line 40 of file shared.h.

7.70.1.3 #define [DEBUG](#) 0

Definition at line 32 of file shared.h.

7.70.1.4 #define [MAXBUF_DEFAULT](#) 65536

Definition at line 42 of file shared.h.

7.70.1.5 #define [MAXBUF_LEN](#) 4

Definition at line 35 of file shared.h.

7.70.1.6 #define [MUTUAL](#) (1 << 3)

Definition at line 37 of file shared.h.

7.70.1.7 #define [RANDOM_LEN](#) 16

Definition at line 36 of file shared.h.

7.70.1.8 #define SERVER_HELLO_LEN BITMAP_LEN + MAXBUF_LEN + RANDOM_LEN

Definition at line 39 of file shared.h.

7.71 suggest.c File Reference

```
#include "internal.h"
```

Functions

- const char * [gsasl_client_suggest_mechanism](#) ([Gsasl](#) *ctx, const char *mechlist)

7.71.1 Function Documentation

7.71.1.1 const char* [gsasl_client_suggest_mechanism](#) ([Gsasl](#) * ctx, const char * *mechlist*)

[gsasl_client_suggest_mechanism](#):

Parameters:

ctx libgsasl handle.

mechlist 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.

Definition at line 37 of file [suggest.c](#).

References [Gsasl::client_mechs](#), [gsasl_client_start\(\)](#), [gsasl_finish\(\)](#), [GSASL_OK](#), [GSASL_VALID_MECHANISM_CHARACTERS](#), [Gsasl::n_client_mechs](#), and [Gsasl_mechanism::name](#).

7.72 supportp.c File Reference

```
#include "internal.h"
```

Functions

- int [gsasl_client_support_p](#) ([Gsasl](#) *ctx, const char *name)
- int [gsasl_server_support_p](#) ([Gsasl](#) *ctx, const char *name)

7.72.1 Function Documentation

7.72.1.1 int [gsasl_client_support_p](#) ([Gsasl](#) * ctx, const char * name)

[gsasl_client_support_p](#):

Parameters:

- ctx* libgsasl handle.
- name* 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 49 of file supportp.c.

References [Gsasl::client_mechs](#), and [Gsasl::n_client_mechs](#).

7.72.1.2 int [gsasl_server_support_p](#) ([Gsasl](#) * ctx, const char * name)

[gsasl_server_support_p](#):

Parameters:

- ctx* libgsasl handle.
- name* 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 66 of file supportp.c.

References [Gsasl::n_server_mechs](#), and [Gsasl::server_mechs](#).

7.73 test-gc-hmac-md5.c File Reference

```
#include <stdio.h>
#include <string.h>
#include "gc.h"
```

Functions

- `int main (int argc, char *argv[])`

7.73.1 Function Documentation

7.73.1.1 `int main (int argc, char * argv[])`

Definition at line 29 of file test-gc-hmac-md5.c.

7.74 test-gc-md5.c File Reference

```
#include <stdio.h>
#include <string.h>
#include "gc.h"
```

Functions

- `int main (int argc, char *argv[])`

7.74.1 Function Documentation

7.74.1.1 `int main (int argc, char * argv[])`

Definition at line 29 of file test-gc-md5.c.

7.75 test-gc.c File Reference

```
#include <stdio.h>
#include <string.h>
#include "gc.h"
```

Functions

- int [main](#) (int argc, char *argv[])

7.75.1 Function Documentation

7.75.1.1 int main (int argc, char * argv[])

Definition at line 29 of file test-gc.c.

7.76 test-parser.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "parser.h"
#include "printer.h"
#include "digesthmac.h"
```

Functions

- `int main (int argc, char *argv[])`

7.76.1 Function Documentation

7.76.1.1 `int main (int argc, char * argv[])`

Definition at line 32 of file test-parser.c.

References `digest_md5_challenge::ciphers`, `digest_md5_response::nonce`, `digest_md5_hmac()`, `digest_md5_parse_challenge()`, `digest_md5_parse_finish()`, `digest_md5_parse_response()`, `digest_md5_print_challenge()`, `digest_md5_print_response()`, `DIGEST_MD5_QOP_AUTH`, `digest_md5_response::digesturi`, `digest_md5_response::nc`, `digest_md5_response::nonce`, `digest_md5_challenge::nonce`, `digest_md5_challenge::nrealms`, `digest_md5_challenge::qops`, `digest_md5_challenge::realms`, `digest_md5_response::response`, `digest_md5_finish::rspauth`, and `digest_md5_response::username`.

7.77 tokens.h File Reference

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

Data Structures

- struct [digest_md5_challenge](#)
- struct [digest_md5_response](#)
- struct [digest_md5_finish](#)

Defines

- #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 [digest_md5_challenge](#) [digest_md5_challenge](#)
- typedef [digest_md5_response](#) [digest_md5_response](#)
- typedef [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 }

7.77.1 Define Documentation

7.77.1.1 #define DIGEST_MD5_LENGTH 16

Definition at line 30 of file tokens.h.

Referenced by [_gsasl_digest_md5_client_step\(\)](#), and [_gsasl_digest_md5_server_step\(\)](#).

7.77.1.2 #define DIGEST_MD5_RESPONSE_LENGTH 32

Definition at line 95 of file tokens.h.

Referenced by [_gsasl_digest_md5_client_step\(\)](#), [_gsasl_digest_md5_server_step\(\)](#), [digest_md5_validate_finish\(\)](#), and [digest_md5_validate_response\(\)](#).

7.77.2 Typedef Documentation

7.77.2.1 typedef struct [digest_md5_challenge](#) [digest_md5_challenge](#)

Definition at line 93 of file tokens.h.

7.77.2.2 typedef enum [digest_md5_cipher](#) [digest_md5_cipher](#)

Definition at line 51 of file tokens.h.

7.77.2.3 typedef struct [digest_md5_finish](#) [digest_md5_finish](#)

Definition at line 150 of file tokens.h.

7.77.2.4 typedef enum [digest_md5_qop](#) [digest_md5_qop](#)

Definition at line 39 of file tokens.h.

7.77.2.5 typedef struct [digest_md5_response](#) [digest_md5_response](#)

Definition at line 141 of file tokens.h.

7.77.3 Enumeration Type Documentation

7.77.3.1 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 tokens.h.

7.77.3.2 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 tokens.h.

7.78 validate.c File Reference

```
#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)

7.78.1 Function Documentation

7.78.1.1 int [digest_md5_validate](#) ([digest_md5_challenge](#) *c, [digest_md5_response](#) *r)

Definition at line 116 of file `validate.c`.

References `digest_md5_response::cipher`, `digest_md5_challenge::ciphers`, `DIGEST_MD5_QOP_AUTH`, `DIGEST_MD5_QOP_AUTH_CONF`, `digest_md5_response::nc`, `digest_md5_challenge::nonce`, `digest_md5_response::nonce`, `digest_md5_response::qop`, `digest_md5_challenge::qops`, `digest_md5_challenge::utf8`, and `digest_md5_response::utf8`.

Referenced by `_gsasl_digest_md5_server_step()`.

7.78.1.2 int [digest_md5_validate_challenge](#) ([digest_md5_challenge](#) *c)

Definition at line 34 of file `validate.c`.

References `digest_md5_challenge::ciphers`, `DIGEST_MD5_QOP_AUTH_CONF`, `digest_md5_challenge::nonce`, and `digest_md5_challenge::qops`.

Referenced by `digest_md5_print_challenge()`.

7.78.1.3 int [digest_md5_validate_finish](#) ([digest_md5_finish](#) *f)

Definition at line 103 of file `validate.c`.

References `DIGEST_MD5_RESPONSE_LENGTH`, and `digest_md5_finish::rspauth`.

Referenced by `digest_md5_print_finish()`.

7.78.1.4 int [digest_md5_validate_response](#) ([digest_md5_response](#) *r)

Definition at line 53 of file `validate.c`.

References `digest_md5_response::cipher`, `digest_md5_response::cnonce`, `DIGEST_MD5_QOP_AUTH_CONF`, `DIGEST_MD5_RESPONSE_LENGTH`, `digest_md5_response::digesturi`, `digest_md5_response::nc`, `digest_md5_response::nonce`, `digest_md5_response::qop`, `digest_md5_response::response`, and `digest_md5_response::username`.

Referenced by `digest_md5_print_response()`.

7.79 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)

7.79.1 Function Documentation

7.79.1.1 int [digest_md5_validate](#) ([digest_md5_challenge](#) *c, [digest_md5_response](#) *r)

Definition at line 116 of file [validate.c](#).

References [digest_md5_response::cipher](#), [digest_md5_challenge::ciphers](#), [DIGEST_MD5_QOP_AUTH](#), [DIGEST_MD5_QOP_AUTH_CONF](#), [digest_md5_response::nc](#), [digest_md5_response::nonce](#), [digest_md5_challenge::nonce](#), [digest_md5_response::qop](#), [digest_md5_challenge::qops](#), [digest_md5_response::utf8](#), and [digest_md5_challenge::utf8](#).

Referenced by [_gsasl_digest_md5_server_step\(\)](#).

7.79.1.2 int [digest_md5_validate_challenge](#) ([digest_md5_challenge](#) *c)

Definition at line 34 of file [validate.c](#).

References [digest_md5_challenge::ciphers](#), [DIGEST_MD5_QOP_AUTH_CONF](#), [digest_md5_challenge::nonce](#), and [digest_md5_challenge::qops](#).

Referenced by [digest_md5_print_challenge\(\)](#).

7.79.1.3 int [digest_md5_validate_finish](#) ([digest_md5_finish](#) *f)

Definition at line 103 of file [validate.c](#).

References [DIGEST_MD5_RESPONSE_LENGTH](#), and [digest_md5_finish::rspauth](#).

Referenced by [digest_md5_print_finish\(\)](#).

7.79.1.4 int [digest_md5_validate_response](#) ([digest_md5_response](#) *r)

Definition at line 53 of file [validate.c](#).

References [digest_md5_response::cipher](#), [digest_md5_response::cnonce](#), [DIGEST_MD5_QOP_AUTH_CONF](#), [DIGEST_MD5_RESPONSE_LENGTH](#), [digest_md5_response::digesturi](#), [digest_md5_response::nc](#), [digest_md5_response::nonce](#), [digest_md5_response::qop](#), [digest_md5_response::response](#), and [digest_md5_response::username](#).

Referenced by [digest_md5_print_response\(\)](#).

7.80 version.c File Reference

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

Functions

- const char * [gsasl_check_version](#) (const char *req_version)

7.80.1 Function Documentation

7.80.1.1 const char* [gsasl_check_version](#) (const char * *req_version*)

[gsasl_check_version](#):

Parameters:

req_version version string to compare with, or NULL.

Check library version.

See GSASL_VERSION for a suitable string.

Return value: Check that the the version of the library is at minimum the one given as a string in 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 42 of file version.c.

7.81 x-gssapi.h File Reference

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

Defines

- #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](#)

7.81.1 Define Documentation

7.81.1.1 #define [GSASL_GSSAPI_NAME](#) "GSSAPI"

Definition at line 28 of file x-gssapi.h.

7.81.2 Function Documentation

7.81.2.1 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 330 of file gssapi/client.c.

References [_Gsasl_gssapi_client_state::context](#), [GSASL_GSSAPI_RELEASE_BUFFER_ERROR](#), [GSASL_GSSAPI_UNWRAP_ERROR](#), [GSASL_MALLOC_ERROR](#), [GSASL_OK](#), [GSASL_QOP_AUTH_CONF](#), [GSASL_QOP_AUTH_INT](#), [_Gsasl_gssapi_client_state::qop](#), and [_Gsasl_gssapi_client_state::step](#).

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

Definition at line 275 of file gssapi/client.c.

References `_Gssapi_client_state::context`, `GSASL_GSSAPI_RELEASE_BUFFER_ERROR`, `GSASL_GSSAPI_WRAP_ERROR`, `GSASL_MALLOC_ERROR`, `GSASL_OK`, `GSASL_QOP_AUTH_CONF`, `GSASL_QOP_AUTH_INT`, `_Gssapi_client_state::qop`, and `_Gssapi_client_state::step`.

7.81.2.3 `void _gssapi_client_finish (Gssapi_session * sctx, void * mech_data)`

Definition at line 257 of file gssapi/client.c.

References `_Gssapi_client_state::context`, and `_Gssapi_client_state::service`.

7.81.2.4 `int _gssapi_client_start (Gssapi_session * sctx, void ** mech_data)`

Definition at line 59 of file gssapi/client.c.

References `GSASL_MALLOC_ERROR`, `GSASL_OK`, and `GSASL_QOP_AUTH`.

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

Definition at line 78 of file gssapi/client.c.

References `_Gssapi_client_state::context`, `GSASL_AUTHID`, `GSASL_GSSAPI_IMPORT_NAME_ERROR`, `GSASL_GSSAPI_INIT_SEC_CONTEXT_ERROR`, `GSASL_GSSAPI_RELEASE_BUFFER_ERROR`, `GSASL_GSSAPI_UNSUPPORTED_PROTECTION_ERROR`, `GSASL_GSSAPI_UNWRAP_ERROR`, `GSASL_GSSAPI_WRAP_ERROR`, `GSASL_HOSTNAME`, `GSASL_MALLOC_ERROR`, `GSASL_MECHANISM_CALLED_TOO_MANY_TIMES`, `GSASL_MECHANISM_PARSE_ERROR`, `GSASL_NEEDS_MORE`, `GSASL_NO_AUTHID`, `GSASL_NO_HOSTNAME`, `GSASL_NO_SERVICE`, `GSASL_OK`, `gssapi_property_get()`, `GSASL_SERVICE`, `_Gssapi_client_state::qop`, `_Gssapi_client_state::service`, and `_Gssapi_client_state::step`.

7.81.2.6 `void _gssapi_server_finish (Gssapi_session * sctx, void * mech_data)`

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

References `_Gssapi_server_state::client`, `_Gssapi_server_state::context`, and `_Gssapi_server_state::cred`.

7.81.2.7 `int _gssapi_server_start (Gssapi_session * sctx, void ** mech_data)`

Definition at line 59 of file gssapi/server.c.

References `GSASL_GSSAPI_ACQUIRE_CRED_ERROR`, `GSASL_GSSAPI_IMPORT_NAME_ERROR`, `GSASL_HOSTNAME`, `GSASL_MALLOC_ERROR`, `GSASL_NO_HOSTNAME`, `GSASL_NO_SERVICE`, `GSASL_OK`, `gssapi_property_get()`, and `GSASL_SERVICE`.

7.81.2.8 `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 121 of file gssapi/server.c.

References `_Gsasl_gssapi_server_state::client`, `_Gsasl_gssapi_server_state::context`, `_Gsasl_gssapi_server_state::cred`, `GSASL_AUTHZID`, `gsasl_callback()`, `GSASL_GSSAPI_ACCEPT_SEC_CONTEXT_ERROR`, `GSASL_GSSAPI_DISPLAY_NAME`, `GSASL_GSSAPI_DISPLAY_NAME_ERROR`, `GSASL_GSSAPI_RELEASE_BUFFER_ERROR`, `GSASL_GSSAPI_UNSUPPORTED_PROTECTION_ERROR`, `GSASL_GSSAPI_UNWRAP_ERROR`, `GSASL_GSSAPI_WRAP_ERROR`, `GSASL_MALLOC_ERROR`, `GSASL_MECHANISM_CALLED_TOO_MANY_TIMES`, `GSASL_NEEDS_MORE`, `gsasl_property_set_raw()`, `GSASL_QOP_AUTH`, `GSASL_VALIDATE_GSSAPI`, and `_Gsasl_gssapi_server_state::step`.

7.81.3 Variable Documentation

7.81.3.1 `Gsasl_mechanism` `gsasl_gssapi_mechanism`

Definition at line 30 of file gssapi/mechinfo.c.

7.82 x-ntlm.h File Reference

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

Defines

- `#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`

7.82.1 Define Documentation

7.82.1.1 `#define GSASL_NTLM_NAME "NTLM"`

Definition at line 28 of file x-ntlm.h.

7.82.2 Function Documentation

7.82.2.1 `void _gsasl_ntlm_client_finish (Gsasl_session * sctx, void * mech_data)`

Definition at line 135 of file ntlm.c.

7.82.2.2 `int _gsasl_ntlm_client_start (Gsasl_session * sctx, void ** mech_data)`

Definition at line 45 of file ntlm.c.

References `GSASL_MALLOC_ERROR`, and `GSASL_OK`.

7.82.2.3 `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 61 of file ntlm.c.

References `GSASL_AUTHID`, `GSASL_MALLOC_ERROR`, `GSASL_MECHANISM_CALLED_TOO_MANY_TIMES`, `GSASL_MECHANISM_PARSE_ERROR`, `GSASL_NEEDS_MORE`, `GSASL_NO_AUTHID`, `GSASL_NO_PASSWORD`, `GSASL_OK`, `GSASL_PASSWORD`, `gsasl_property_get()`, `GSASL_REALM`, and `_Gsasl_ntlm_state::step`.

7.82.3 Variable Documentation

7.82.3.1 [Gsasl_mechanism](#) [gsasl_ntlm_mechanism](#)

Definition at line 30 of file ntlm/mechinfo.c.

7.83 xcode.c File Reference

```
#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*)

7.83.1 Function Documentation

7.83.1.1 int `gsasl_decode` (`Gsasl_session` **sctx*, const char **input*, size_t *input_len*, char ***output*, size_t **output_len*)

`gsasl_decode`:

Parameters:

- sctx* libgsasl session handle.
- input* input byte array.
- input_len* size of input byte array.
- output* newly allocated output byte array.
- output_len* size of output byte array.

Decode data according to negotiated SASL mechanism. This might mean that data is integrity or privacy protected.

The buffer is allocated by this function, and it is the responsibility of caller to deallocate it by calling `free()`.

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

Definition at line 96 of file `xcode.c`.

References `Gsasl_mechanism::client`, `Gsasl_session::clientp`, `Gsasl_mechanism_functions::decode`, `Gsasl_session::mech`, and `Gsasl_mechanism::server`.

Referenced by `gsasl_decode_inline()`.

7.83.1.2 int `gsasl_encode` (`Gsasl_session` **sctx*, const char **input*, size_t *input_len*, char ***output*, size_t **output_len*)

`gsasl_encode`:

Parameters:

- sctx* libgsasl session handle.
- input* input byte array.
- input_len* size of input byte array.
- output* newly allocated output byte array.
- output_len* size of output byte array.

Encode data according to negotiated SASL mechanism. This might mean that data is integrity or privacy protected.

The buffer is allocated by this function, and it is the responsibility of caller to deallocate it by calling `free()`.

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

Definition at line 64 of file `xcode.c`.

References `Gsasl_mechanism::client`, `Gsasl_session::clientp`, `Gsasl_mechanism_functions::encode`, `Gsasl_session::mech`, and `Gsasl_mechanism::server`.

Referenced by `gsasl_encode_inline()`.

7.84 xfinish.c File Reference

```
#include "internal.h"
```

Functions

- void `gsasl_finish` (`Gsasl_session *sctx`)

7.84.1 Function Documentation

7.84.1.1 void `gsasl_finish` (`Gsasl_session * sctx`)

`gsasl_finish`:

Parameters:

sctx 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 33 of file `xfinish.c`.

References `Gsasl_session::anonymous_token`, `Gsasl_session::authid`, `Gsasl_session::authzid`, `Gsasl_mechanism::client`, `Gsasl_session::clientp`, `Gsasl_mechanism_functions::finish`, `Gsasl_session::gssapi_display_name`, `Gsasl_session::hostname`, `Gsasl_session::mech`, `Gsasl_session::mech_data`, `Gsasl_session::passcode`, `Gsasl_session::password`, `Gsasl_session::pin`, `Gsasl_session::realm`, `Gsasl_mechanism::server`, `Gsasl_session::service`, and `Gsasl_session::suggestedpin`.

Referenced by `gsasl_client_finish()`, `gsasl_client_suggest_mechanism()`, and `gsasl_server_finish()`.

7.85 xstart.c File Reference

```
#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`)

7.85.1 Function Documentation

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

`gsasl_client_start`:

Parameters:

- ctx* libgsasl handle.
- mech* name of SASL mechanism.
- sctx* 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 119 of file `xstart.c`.

References `Gsasl::client_mechs`, and `Gsasl::n_client_mechs`.

Referenced by `gsasl_client_suggest_mechanism()`.

7.85.1.2 int `gsasl_server_start` (`Gsasl * ctx`, const char * `mech`, `Gsasl_session ** sctx`)

`gsasl_server_start`:

Parameters:

- ctx* libgsasl handle.
- mech* name of SASL mechanism.
- sctx* 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 137 of file `xstart.c`.

References `Gsasl::n_server_mechs`, and `Gsasl::server_mechs`.

7.86 xstep.c File Reference

```
#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*)

7.86.1 Function Documentation

7.86.1.1 int `gsasl_step` (`Gsasl_session` * *sctx*, const char * *input*, size_t *input_len*, char ** *output*, size_t * *output_len*)

`gsasl_step`:

Parameters:

sctx libgsasl session handle.

input input byte array.

input_len size of input byte array.

output newly allocated output byte array.

output_len 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 that indicate the length of `output`).

The contents of the 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 buffer is allocated by this function, and it is the responsibility of caller to deallocate it by calling `free()`.

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

Definition at line 51 of file `xstep.c`.

References `Gsasl_mechanism::client`, `Gsasl_session::clientp`, `Gsasl_session::mech`, `Gsasl_session::mech_data`, `Gsasl_mechanism::server`, and `Gsasl_mechanism_functions::step`.

Referenced by `gsasl_step64()`.

7.86.1.2 int `gsasl_step64` (`Gsasl_session` * *sctx*, const char * *b64input*, char ** *b64output*)

`gsasl_step64`:

Parameters:

sctx libgsasl client handle.

b64input input base64 encoded byte array.

b64output 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 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 buffer is allocated by this function, and it is the responsibility of caller to deallocate it by calling `free ()`.

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

Definition at line 86 of file xstep.c.

References GSASL_BASE64_ERROR, [gsasl_base64_from\(\)](#), [gsasl_base64_to\(\)](#), GSASL_NEEDS_MORE, GSASL_OK, and [gsasl_step\(\)](#).

Index

- [/home/jas/self/src/gsaslib/ Directory Reference, 25](#)
- [/home/jas/self/src/gsaslib/anonymous/ Directory Reference, 19](#)
- [/home/jas/self/src/gsaslib/cram-md5/ Directory Reference, 20](#)
- [/home/jas/self/src/gsaslib/digest-md5/ Directory Reference, 21](#)
- [/home/jas/self/src/gsaslib/external/ Directory Reference, 22](#)
- [/home/jas/self/src/gsaslib/gssapi/ Directory Reference, 23](#)
- [/home/jas/self/src/gsaslib/kerberos_v5/ Directory Reference, 24](#)
- [/home/jas/self/src/gsaslib/login/ Directory Reference, 26](#)
- [/home/jas/self/src/gsaslib/ntlm/ Directory Reference, 27](#)
- [/home/jas/self/src/gsaslib/plain/ Directory Reference, 28](#)
- [/home/jas/self/src/gsaslib/securid/ Directory Reference, 29](#)
- [/home/jas/self/src/gsaslib/src/ Directory Reference, 30](#)
- [/home/jas/self/src/gsaslib/tests/ Directory Reference, 31](#)
-
- [error.c, 108](#)
- [_GSASL_KERBEROS_V5_NAME](#)
 - [kerberos_v5.h, 176](#)
- [_Gssapi_digest_md5_client_state, 33](#)
 - [challenge, 33](#)
 - [digest-md5/client.c, 80](#)
 - [finish, 33](#)
 - [kcc, 33](#)
 - [kcs, 34](#)
 - [kic, 34](#)
 - [kis, 34](#)
 - [readseqnum, 34](#)
 - [response, 34](#)
 - [secret, 34](#)
 - [sendseqnum, 34](#)
 - [step, 34](#)
- [_Gssapi_digest_md5_server_state, 36](#)
 - [challenge, 36](#)
 - [digest-md5/server.c, 241](#)
 - [finish, 36](#)
 - [kcc, 36](#)
 - [kcs, 36](#)
 - [kic, 36](#)
 - [kis, 36](#)
 - [readseqnum, 37](#)
 - [response, 37](#)
 - [secret, 37](#)
 - [sendseqnum, 37](#)
 - [step, 37](#)
- [_Gssapi_client_state, 38](#)
 - [context, 38](#)
 - [gssapi/client.c, 83](#)
 - [qop, 38](#)
 - [service, 38](#)
 - [step, 38](#)
- [_Gssapi_server_state, 39](#)
 - [client, 39](#)
 - [context, 39](#)
 - [cred, 39](#)
 - [gssapi/server.c, 244](#)
 - [step, 39](#)
- [_Gssapi_kerberos_v5_client_state, 40](#)
 - [ap, 40](#)
 - [as, 40](#)
 - [clientmaxbuf, 40](#)
 - [clientqop, 40](#)
 - [safe, 40](#)
 - [serverhello, 41](#)
 - [servermaxbuf, 41](#)
 - [servermutual, 41](#)
 - [serverqops, 41](#)
 - [sessionkey, 41](#)
 - [sh, 41](#)
 - [step, 41](#)
 - [tkt, 41](#)
- [_Gssapi_kerberos_v5_server_state, 43](#)
 - [ap, 43](#)
 - [as, 43](#)
 - [clientmaxbuf, 43](#)
 - [clientmutual, 43](#)
 - [clientqop, 44](#)
 - [firststep, 44](#)
 - [password, 44](#)
 - [random, 44](#)

- safe, 44
- serverhello, 44
- serverhostname, 44
- servermaxbuf, 44
- serverqops, 45
- serverrealm, 45
- serverservice, 45
- sessionkey, 45
- sessionktkey, 45
- sh, 45
- userkey, 45
- username, 45
- userrealm, 46
- _Gsasl_login_client_state, 47
 - step, 47
- _Gsasl_login_server_state, 48
 - password, 48
 - step, 48
 - username, 48
- _Gsasl_ntlm_state, 49
 - ntlm.c, 192
 - step, 49
- __attribute__
 - gsasl-compat.h, 118
- _gsasl_anonymous_client_step
 - anonymous.h, 67
 - anonymous/client.c, 77
- _gsasl_anonymous_server_step
 - anonymous.h, 67
 - anonymous/server.c, 238
- _gsasl_cram_md5_client_step
 - cram-md5.h, 92
 - cram-md5/client.c, 78
- _gsasl_cram_md5_server_finish
 - cram-md5.h, 92
 - cram-md5/server.c, 239
- _gsasl_cram_md5_server_start
 - cram-md5.h, 92
 - cram-md5/server.c, 239
- _gsasl_cram_md5_server_step
 - cram-md5.h, 92
 - cram-md5/server.c, 239
- _gsasl_digest_md5_client_decode
 - digest-md5.h, 96
 - digest-md5/client.c, 80
- _gsasl_digest_md5_client_encode
 - digest-md5.h, 96
 - digest-md5/client.c, 80
- _gsasl_digest_md5_client_finish
 - digest-md5.h, 97
 - digest-md5/client.c, 80
- _gsasl_digest_md5_client_start
 - digest-md5.h, 97
 - digest-md5/client.c, 80
- _gsasl_digest_md5_client_step
 - digest-md5.h, 97
 - digest-md5/client.c, 80
- _gsasl_digest_md5_server_decode
 - digest-md5.h, 97
 - digest-md5/server.c, 241
- _gsasl_digest_md5_server_encode
 - digest-md5.h, 98
 - digest-md5/server.c, 241
- _gsasl_digest_md5_server_finish
 - digest-md5.h, 98
 - digest-md5/server.c, 241
- _gsasl_digest_md5_server_start
 - digest-md5.h, 98
 - digest-md5/server.c, 241
- _gsasl_digest_md5_server_step
 - digest-md5.h, 98
 - digest-md5/server.c, 241
- _gsasl_external_client_step
 - external.h, 110
 - external/client.c, 82
- _gsasl_external_server_step
 - external.h, 110
 - external/server.c, 243
- _gsasl_gssapi_client_decode
 - gssapi/client.c, 83
 - x-gssapi.h, 268
- _gsasl_gssapi_client_encode
 - gssapi/client.c, 83
 - x-gssapi.h, 268
- _gsasl_gssapi_client_finish
 - gssapi/client.c, 84
 - x-gssapi.h, 269
- _gsasl_gssapi_client_start
 - gssapi/client.c, 84
 - x-gssapi.h, 269
- _gsasl_gssapi_client_step
 - gssapi/client.c, 84
 - x-gssapi.h, 269
- _gsasl_gssapi_server_finish
 - gssapi/server.c, 244
 - x-gssapi.h, 269
- _gsasl_gssapi_server_start
 - gssapi/server.c, 244
 - x-gssapi.h, 269
- _gsasl_gssapi_server_step
 - gssapi/server.c, 244
 - x-gssapi.h, 269
- _gsasl_kerberos_v5_client_decode
 - kerberos_v5.h, 176
 - kerberos_v5/client.c, 86
- _gsasl_kerberos_v5_client_encode
 - kerberos_v5.h, 176
 - kerberos_v5/client.c, 86

- `_gsasl_kerberos_v5_client_finish`
 - `kerberos_v5.h`, 177
 - `kerberos_v5/client.c`, 86
- `_gsasl_kerberos_v5_client_init`
 - `kerberos_v5.h`, 177
 - `kerberos_v5/client.c`, 86
- `_gsasl_kerberos_v5_client_start`
 - `kerberos_v5.h`, 177
 - `kerberos_v5/client.c`, 86
- `_gsasl_kerberos_v5_client_step`
 - `kerberos_v5.h`, 177
 - `kerberos_v5/client.c`, 87
- `_gsasl_kerberos_v5_server_decode`
 - `kerberos_v5.h`, 177
 - `kerberos_v5/server.c`, 246
- `_gsasl_kerberos_v5_server_encode`
 - `kerberos_v5.h`, 177
 - `kerberos_v5/server.c`, 246
- `_gsasl_kerberos_v5_server_finish`
 - `kerberos_v5.h`, 178
 - `kerberos_v5/server.c`, 246
- `_gsasl_kerberos_v5_server_init`
 - `kerberos_v5.h`, 178
 - `kerberos_v5/server.c`, 246
- `_gsasl_kerberos_v5_server_start`
 - `kerberos_v5.h`, 178
 - `kerberos_v5/server.c`, 247
- `_gsasl_kerberos_v5_server_step`
 - `kerberos_v5.h`, 178
 - `kerberos_v5/server.c`, 247
- `_gsasl_login_client_finish`
 - `login.h`, 180
 - `login/client.c`, 88
- `_gsasl_login_client_start`
 - `login.h`, 180
 - `login/client.c`, 88
- `_gsasl_login_client_step`
 - `login.h`, 180
 - `login/client.c`, 88
- `_gsasl_login_server_finish`
 - `login.h`, 180
 - `login/server.c`, 248
- `_gsasl_login_server_start`
 - `login.h`, 181
 - `login/server.c`, 248
- `_gsasl_login_server_step`
 - `login.h`, 181
 - `login/server.c`, 248
- `_gsasl_ntlm_client_finish`
 - `ntlm.c`, 192
 - `x-ntlm.h`, 271
- `_gsasl_ntlm_client_start`
 - `ntlm.c`, 192
 - `x-ntlm.h`, 271
- `_gsasl_ntlm_client_step`
 - `ntlm.c`, 192
 - `x-ntlm.h`, 271
- `_gsasl_plain_client_step`
 - `plain.h`, 227
 - `plain/client.c`, 89
- `_gsasl_plain_server_step`
 - `plain.h`, 227
 - `plain/server.c`, 250
- `_gsasl_secured_client_finish`
 - `securid.h`, 236
 - `securid/client.c`, 90
- `_gsasl_secured_client_start`
 - `securid.h`, 236
 - `securid/client.c`, 90
- `_gsasl_secured_client_step`
 - `securid.h`, 236
 - `securid/client.c`, 90
- `_gsasl_secured_server_step`
 - `securid.h`, 236
 - `securid/server.c`, 251
- A2_POST
 - `digestmac.c`, 101
- A2_PRE
 - `digestmac.c`, 101
- `anonymous.h`, 67
 - `_gsasl_anonymous_client_step`, 67
 - `_gsasl_anonymous_server_step`, 67
 - `gsasl_anonymous_mechanism`, 68
 - `GSASL_ANONYMOUS_NAME`, 67
- `anonymous/client.c`
 - `_gsasl_anonymous_client_step`, 77
- `anonymous/mechinfo.c`
 - `gsasl_anonymous_mechanism`, 183
- `anonymous/server.c`
 - `_gsasl_anonymous_server_step`, 238
- `anonymous_token`
 - `Gsasl_session`, 64
- `ap`
 - `_Gsasl_kerberos_v5_client_state`, 40
 - `_Gsasl_kerberos_v5_server_state`, 43
- `application_data`
 - `Gsasl_session`, 64
- `application_hook`
 - `Gsasl`, 56
 - `Gsasl_session`, 64
- `as`
 - `_Gsasl_kerberos_v5_client_state`, 40
 - `_Gsasl_kerberos_v5_server_state`, 43
- `Assert`
 - `obsolete.c`, 196
- `authid`
 - `Gsasl_session`, 64

- authzid
 - digest_md5_response, [53](#)
 - Gsasl_session, [65](#)
- base64.c, [69](#)
 - gsasl_base64_from, [69](#)
 - gsasl_base64_to, [69](#)
- BITMAP_LEN
 - kerberos_v5.c, [174](#)
 - shared.h, [255](#)
- C2I
 - session.c, [252](#)
- callback.c, [71](#)
 - gsasl_callback, [71](#)
 - gsasl_callback_hook_get, [71](#)
 - gsasl_callback_hook_set, [72](#)
 - gsasl_callback_set, [72](#)
 - gsasl_session_hook_get, [73](#)
 - gsasl_session_hook_set, [73](#)
- cb
 - Gsasl, [56](#)
- cbc_anonymous
 - Gsasl, [57](#)
- cbc_authentication_id
 - Gsasl, [57](#)
- cbc_authorization_id
 - Gsasl, [57](#)
- cbc_maxbuf
 - Gsasl, [57](#)
- cbc_passcode
 - Gsasl, [57](#)
- cbc_password
 - Gsasl, [57](#)
- cbc_pin
 - Gsasl, [57](#)
- cbc_qop
 - Gsasl, [57](#)
- cbc_realm
 - Gsasl, [58](#)
- cbc_service
 - Gsasl, [58](#)
- cbs_anonymous
 - Gsasl, [58](#)
- cbs_cipher
 - Gsasl, [58](#)
- cbs_cram_md5
 - Gsasl, [58](#)
- cbs_digest_md5
 - Gsasl, [58](#)
- cbs_external
 - Gsasl, [58](#)
- cbs_gssapi
 - Gsasl, [58](#)
- cbs_maxbuf
 - Gsasl, [58](#)
- cbs_qop
 - Gsasl, [59](#)
- cbs_realm
 - Gsasl, [59](#)
- cbs_retrieve
 - Gsasl, [59](#)
- cbs_securid
 - Gsasl, [59](#)
- cbs_service
 - Gsasl, [59](#)
- cbs_validate
 - Gsasl, [59](#)
- challenge
 - _Gsasl_digest_md5_client_state, [33](#)
 - _Gsasl_digest_md5_server_state, [36](#)
- challenge.c, [74](#)
 - cram_md5_challenge, [75](#)
 - DIGIT, [74](#)
 - NONCELEN, [74](#)
 - TEMPLATE, [74](#)
- challenge.h, [76](#)
 - cram_md5_challenge, [76](#)
 - CRAM_MD5_CHALLENGE_LEN, [76](#)
- CHALLENGE_ALGORITHM
 - parser.c, [224](#)
- CHALLENGE_CHARSET
 - parser.c, [224](#)
- CHALLENGE_CIPHER
 - parser.c, [224](#)
- CHALLENGE_MAXBUF
 - parser.c, [224](#)
- CHALLENGE_NONCE
 - parser.c, [224](#)
- CHALLENGE_PASSWORD
 - login/server.c, [248](#)
- CHALLENGE_QOP
 - parser.c, [224](#)
- CHALLENGE_REALM
 - parser.c, [224](#)
- CHALLENGE_STALE
 - parser.c, [224](#)
- CHALLENGE_USERNAME
 - login/server.c, [248](#)
- cipher
 - digest_md5_response, [53](#)
- CIPHER_3DES
 - parser.c, [224](#)
- CIPHER_AES_CBC
 - parser.c, [224](#)
- CIPHER_DES
 - parser.c, [224](#)
- CIPHER_RC4

- parser.c, 224
- CIPHER_RC4_40
 - parser.c, 224
- CIPHER_RC4_56
 - parser.c, 224
- ciphers
 - digest_md5_challenge, 50
- client
 - _Gssasl_gssapi_server_state, 39
 - Gssasl_mechanism, 61
- client.c, 77–79, 82, 83, 85, 88–90
- CLIENT_HELLO_LEN
 - kerberos_v5.c, 174
 - shared.h, 255
- client_mechs
 - Gssasl, 59
- clientmaxbuf
 - _Gssasl_kerberos_v5_client_state, 40
 - _Gssasl_kerberos_v5_server_state, 43
 - digest_md5_response, 53
- clientmutual
 - _Gssasl_kerberos_v5_server_state, 43
- clientp
 - Gssasl_session, 65
- clientqop
 - _Gssasl_kerberos_v5_client_state, 40
 - _Gssasl_kerberos_v5_server_state, 44
- cnonce
 - digest_md5_response, 53
- CNONCE_ENTROPY_BYTES
 - digest-md5/client.c, 79
- COLON
 - digesthmac.c, 102
- context
 - _Gssasl_gssapi_client_state, 38
 - _Gssasl_gssapi_server_state, 39
- cram-md5.h, 92
 - _gssasl_cram_md5_client_step, 92
 - _gssasl_cram_md5_server_finish, 92
 - _gssasl_cram_md5_server_start, 92
 - _gssasl_cram_md5_server_step, 92
 - gssasl_cram_md5_mechanism, 93
 - GSASL_CRAM_MD5_NAME, 92
- cram-md5/client.c
 - _gssasl_cram_md5_client_step, 78
- cram-md5/mechinfo.c
 - gssasl_cram_md5_mechanism, 184
- cram-md5/server.c
 - _gssasl_cram_md5_server_finish, 239
 - _gssasl_cram_md5_server_start, 239
 - _gssasl_cram_md5_server_step, 239
 - MD5LEN, 239
- cram_md5_challenge
 - challenge.c, 75
 - challenge.h, 76
- CRAM_MD5_CHALLENGE_LEN
 - challenge.h, 76
- cram_md5_digest
 - digest.c, 99
 - digest.h, 100
- CRAM_MD5_DIGEST_LEN
 - digest.h, 100
- cred
 - _Gssasl_gssapi_server_state, 39
- crypto.c, 94
 - gssasl_hmac_md5, 94
 - gssasl_md5, 94
 - gssasl_nonce, 94
 - gssasl_random, 95
- ctx
 - Gssasl_session, 65
- DEBUG
 - kerberos_v5.c, 174
 - shared.h, 255
- decode
 - Gssasl_mechanism_functions, 62
- DEFAULT_ALGORITHM
 - parser.c, 223
- DEFAULT_CHARSET
 - parser.c, 223
- deprecated
 - gssasl-compat.h, 148
- DERIVE_CLIENT_CONFIDENTIALITY_KEY_-STRING
 - digesthmac.c, 102
- DERIVE_CLIENT_CONFIDENTIALITY_KEY_-STRING_LEN
 - digesthmac.c, 102
- DERIVE_CLIENT_INTEGRITY_KEY_STRING
 - digesthmac.c, 102
- DERIVE_CLIENT_INTEGRITY_KEY_-STRING_LEN
 - digesthmac.c, 102
- DERIVE_SERVER_CONFIDENTIALITY_-KEY_STRING
 - digesthmac.c, 102
- DERIVE_SERVER_CONFIDENTIALITY_-KEY_STRING_LEN
 - digesthmac.c, 102
- DERIVE_SERVER_INTEGRITY_KEY_STRING
 - digesthmac.c, 102
- DERIVE_SERVER_INTEGRITY_KEY_-STRING_LEN
 - digesthmac.c, 103
- digest-md5.h, 96
 - _gssasl_digest_md5_client_decode, 96
 - _gssasl_digest_md5_client_encode, 96

- `_gsasl_digest_md5_client_finish`, 97
- `_gsasl_digest_md5_client_start`, 97
- `_gsasl_digest_md5_client_step`, 97
- `_gsasl_digest_md5_server_decode`, 97
- `_gsasl_digest_md5_server_encode`, 98
- `_gsasl_digest_md5_server_finish`, 98
- `_gsasl_digest_md5_server_start`, 98
- `_gsasl_digest_md5_server_step`, 98
- `gsasl_digest_md5_mechanism`, 98
- `GSASL_DIGEST_MD5_NAME`, 96
- `digest-md5/client.c`
 - `_Gsasl_digest_md5_client_state`, 80
 - `_gsasl_digest_md5_client_decode`, 80
 - `_gsasl_digest_md5_client_encode`, 80
 - `_gsasl_digest_md5_client_finish`, 80
 - `_gsasl_digest_md5_client_start`, 80
 - `_gsasl_digest_md5_client_step`, 80
 - `CNONCE_ENTROPY_BYTES`, 79
- `digest-md5/mechinfo.c`
 - `gsasl_digest_md5_mechanism`, 185
- `digest-md5/server.c`
 - `_Gsasl_digest_md5_server_state`, 241
 - `_gsasl_digest_md5_server_decode`, 241
 - `_gsasl_digest_md5_server_encode`, 241
 - `_gsasl_digest_md5_server_finish`, 241
 - `_gsasl_digest_md5_server_start`, 241
 - `_gsasl_digest_md5_server_step`, 241
 - `NONCE_ENTROPY_BYTES`, 240
- `digest.c`, 99
 - `cram_md5_digest`, 99
 - `HEXCHAR`, 99
- `digest.h`, 100
 - `cram_md5_digest`, 100
 - `CRAM_MD5_DIGEST_LEN`, 100
- `digest_md5_challenge`, 50
 - `ciphers`, 50
 - `nonce`, 50
 - `nrealms`, 50
 - `qops`, 50
 - `realms`, 50
 - `servermaxbuf`, 51
 - `stale`, 51
 - `tokens.h`, 264
 - `utf8`, 51
- `digest_md5_cipher`
 - `tokens.h`, 264
- `DIGEST_MD5_CIPHER_3DES`
 - `tokens.h`, 264
- `DIGEST_MD5_CIPHER_AES_CBC`
 - `tokens.h`, 264
- `DIGEST_MD5_CIPHER_DES`
 - `tokens.h`, 264
- `DIGEST_MD5_CIPHER_RC4`
 - `tokens.h`, 264
- `DIGEST_MD5_CIPHER_RC4_40`
 - `tokens.h`, 264
- `DIGEST_MD5_CIPHER_RC4_56`
 - `tokens.h`, 264
- `digest_md5_decode`
 - `session.c`, 253
 - `session.h`, 254
- `digest_md5_encode`
 - `session.c`, 253
 - `session.h`, 254
- `digest_md5_finish`, 52
 - `rspauth`, 52
 - `tokens.h`, 264
- `digest_md5_free_challenge`
 - `free.c`, 111
 - `free.h`, 112
- `digest_md5_free_finish`
 - `free.c`, 111
 - `free.h`, 112
- `digest_md5_free_response`
 - `free.c`, 111
 - `free.h`, 112
- `digest_md5_getsubopt`
 - `getsubopt.c`, 113
 - `parser.h`, 226
- `digest_md5_hmac`
 - `digesthmac.c`, 104
 - `digesthmac.h`, 105
- `DIGEST_MD5_LENGTH`
 - `tokens.h`, 263
- `digest_md5_parse_challenge`
 - `parser.c`, 225
 - `parser.h`, 226
- `digest_md5_parse_finish`
 - `parser.c`, 225
 - `parser.h`, 226
- `digest_md5_parse_response`
 - `parser.c`, 225
 - `parser.h`, 226
- `digest_md5_print_challenge`
 - `printer.c`, 229
 - `printer.h`, 230
- `digest_md5_print_finish`
 - `printer.c`, 229
 - `printer.h`, 230
- `digest_md5_print_response`
 - `printer.c`, 229
 - `printer.h`, 230
- `digest_md5_qop`
 - `tokens.h`, 264
- `DIGEST_MD5_QOP_AUTH`
 - `tokens.h`, 264
- `DIGEST_MD5_QOP_AUTH_CONF`
 - `tokens.h`, 264

- DIGEST_MD5_QOP_AUTH_INT
 - tokens.h, 264
- digest_md5_response, 53
 - authzid, 53
 - cipher, 53
 - clientmaxbuf, 53
 - cnonce, 53
 - digesturi, 53
 - nc, 54
 - nonce, 54
 - qop, 54
 - realm, 54
 - response, 54
 - tokens.h, 264
 - username, 54
 - utf8, 54
- DIGEST_MD5_RESPONSE_LENGTH
 - tokens.h, 263
- digest_md5_validate
 - validate.c, 265
 - validate.h, 266
- digest_md5_validate_challenge
 - validate.c, 265
 - validate.h, 266
- digest_md5_validate_finish
 - validate.c, 265
 - validate.h, 266
- digest_md5_validate_response
 - validate.c, 265
 - validate.h, 266
- digesthmac.c, 101
 - A2_POST, 101
 - A2_PRE, 101
 - COLON, 102
 - DERIVE_CLIENT_CONFIDENTIALITY_-KEY_STRING, 102
 - DERIVE_CLIENT_CONFIDENTIALITY_-KEY_STRING_LEN, 102
 - DERIVE_CLIENT_INTEGRITY_KEY_-STRING, 102
 - DERIVE_CLIENT_INTEGRITY_KEY_-STRING_LEN, 102
 - DERIVE_SERVER_CONFIDENTIALITY_-KEY_STRING, 102
 - DERIVE_SERVER_CONFIDENTIALITY_-KEY_STRING_LEN, 102
 - DERIVE_SERVER_INTEGRITY_KEY_-STRING, 102
 - DERIVE_SERVER_INTEGRITY_KEY_-STRING_LEN, 103
 - digest_md5_hmac, 104
 - HEXCHAR, 103
 - MD5LEN, 103
 - QOP_AUTH, 103
 - QOP_AUTH_CONF, 103
 - QOP_AUTH_INT, 103
 - RESPONSE_LENGTH, 103
 - RSPAUTH_LENGTH, 103
- digesthmac.h, 105
 - digest_md5_hmac, 105
- digesturi
 - digest_md5_response, 53
- DIGIT
 - challenge.c, 74
- done
 - Gsasl_mechanism_functions, 62
- done.c, 106
 - gsasl_done, 106
- doxygen.c, 107
- encode
 - Gsasl_mechanism_functions, 62
- error.c, 108
 - _, 108
 - gettext_noop, 108
 - gsasl_strerror, 108
 - N_, 108
- external.h, 110
 - _gsasl_external_client_step, 110
 - _gsasl_external_server_step, 110
 - gsasl_external_mechanism, 110
 - GSASL_EXTERNAL_NAME, 110
- external/client.c
 - _gsasl_external_client_step, 82
- external/mechinfo.c
 - gsasl_external_mechanism, 186
- external/server.c
 - _gsasl_external_server_step, 243
- finish
 - _Gsasl_digest_md5_client_state, 33
 - _Gsasl_digest_md5_server_state, 36
 - Gsasl_mechanism_functions, 62
- firststep
 - _Gsasl_kerberos_v5_server_state, 44
- free.c, 111
 - digest_md5_free_challenge, 111
 - digest_md5_free_finish, 111
 - digest_md5_free_response, 111
- free.h, 112
 - digest_md5_free_challenge, 112
 - digest_md5_free_finish, 112
 - digest_md5_free_response, 112
- getsubopt.c, 113
 - digest_md5_getsubopt, 113
- gettext_noop
 - error.c, 108

- Gsasl, 56
 - application_hook, 56
 - cb, 56
 - cbc_anonymous, 57
 - cbc_authentication_id, 57
 - cbc_authorization_id, 57
 - cbc_maxbuf, 57
 - cbc_passcode, 57
 - cbc_password, 57
 - cbc_pin, 57
 - cbc_qop, 57
 - cbc_realm, 58
 - cbc_service, 58
 - cbs_anonymous, 58
 - cbs_cipher, 58
 - cbs_cram_md5, 58
 - cbs_digest_md5, 58
 - cbs_external, 58
 - cbs_gssapi, 58
 - cbs_maxbuf, 58
 - cbs_qop, 59
 - cbs_realm, 59
 - cbs_retrieve, 59
 - cbs_securid, 59
 - cbs_service, 59
 - cbs_validate, 59
 - client_mechs, 59
 - gsasl.h, 153
 - n_client_mechs, 59
 - n_server_mechs, 60
 - server_mechs, 60
- gsasl-compat.h, 114
 - __attribute__, 118
 - deprecated, 148
 - gsasl_appinfo_get, 121
 - gsasl_appinfo_set, 121
 - gsasl_application_data_get, 122
 - gsasl_application_data_set, 122
 - gsasl_base64_decode, 122
 - gsasl_base64_encode, 123
 - GSASL_CANNOT_GET_CTX, 121
 - gsasl_client_application_data_get, 123
 - gsasl_client_application_data_set, 123
 - Gsasl_client_callback_anonymous, 118
 - gsasl_client_callback_anonymous_get, 124
 - gsasl_client_callback_anonymous_set, 124
 - Gsasl_client_callback_authentication_id, 118
 - gsasl_client_callback_authentication_id_get, 124
 - gsasl_client_callback_authentication_id_set, 125
 - Gsasl_client_callback_authorization_id, 118
 - gsasl_client_callback_authorization_id_get, 125
 - gsasl_client_callback_authorization_id_set, 126
 - Gsasl_client_callback_maxbuf, 118
 - gsasl_client_callback_maxbuf_get, 126
 - gsasl_client_callback_maxbuf_set, 126
 - Gsasl_client_callback_passcode, 118
 - gsasl_client_callback_passcode_get, 127
 - gsasl_client_callback_passcode_set, 127
 - Gsasl_client_callback_password, 118
 - gsasl_client_callback_password_get, 127
 - gsasl_client_callback_password_set, 128
 - Gsasl_client_callback_pin, 119
 - gsasl_client_callback_pin_get, 128
 - gsasl_client_callback_pin_set, 128
 - Gsasl_client_callback_qop, 119
 - gsasl_client_callback_qop_get, 129
 - gsasl_client_callback_qop_set, 129
 - Gsasl_client_callback_realm, 119
 - gsasl_client_callback_realm_get, 129
 - gsasl_client_callback_realm_set, 130
 - Gsasl_client_callback_service, 119
 - gsasl_client_callback_service_get, 130
 - gsasl_client_callback_service_set, 130
 - gsasl_client_ctx_get, 131
 - gsasl_client_finish, 131
 - gsasl_client_listmech, 131
 - gsasl_client_step, 132
 - gsasl_client_step_base64, 132
 - gsasl_ctx_get, 133
 - gsasl_decode_inline, 133
 - gsasl_encode_inline, 133
 - GSASL_FCLOSE_ERROR, 120
 - GSASL_FOPEN_ERROR, 120
 - GSASL_GCRYPT_ERROR, 121
 - GSASL_INVALID_HANDLE, 121
 - gsasl_md5pwd_get_password, 134
 - GSASL_NEED_CLIENT_ANONYMOUS_-CALLBACK, 121
 - GSASL_NEED_CLIENT_-AUTHENTICATION_ID_CALLBACK, 121
 - GSASL_NEED_CLIENT_-AUTHORIZATION_ID_CALLBACK, 121
 - GSASL_NEED_CLIENT_PASSCODE_-CALLBACK, 121
 - GSASL_NEED_CLIENT_PASSWORD_-CALLBACK, 121
 - GSASL_NEED_CLIENT_PIN_CALLBACK, 121
 - GSASL_NEED_CLIENT_SERVICE_-CALLBACK, 121
 - GSASL_NEED_SERVER_ANONYMOUS_-CALLBACK, 121

- GSASL_NEED_SERVER_CRAM_MD5_-CALLBACK, 121
- GSASL_NEED_SERVER_DIGEST_MD5_-CALLBACK, 121
- GSASL_NEED_SERVER_EXTERNAL_-CALLBACK, 121
- GSASL_NEED_SERVER_GSSAPI_-CALLBACK, 121
- GSASL_NEED_SERVER_REALM_-CALLBACK, 121
- GSASL_NEED_SERVER_RETRIEVE_-CALLBACK, 121
- GSASL_NEED_SERVER_SECURID_-CALLBACK, 121
- GSASL_NEED_SERVER_SERVICE_-CALLBACK, 121
- GSASL_NEED_SERVER_VALIDATE_-CALLBACK, 121
- GSASL_NO_MORE_REALMS, 121
- gsasl_randomize, 134
- gsasl_server_application_data_get, 135
- gsasl_server_application_data_set, 135
- Gsasl_server_callback_anonymous, 119
- gsasl_server_callback_anonymous_get, 135
- gsasl_server_callback_anonymous_set, 136
- Gsasl_server_callback_cipher, 119
- gsasl_server_callback_cipher_get, 136
- gsasl_server_callback_cipher_set, 136
- Gsasl_server_callback_cram_md5, 119
- gsasl_server_callback_cram_md5_get, 137
- gsasl_server_callback_cram_md5_set, 137
- Gsasl_server_callback_digest_md5, 119
- gsasl_server_callback_digest_md5_get, 137
- gsasl_server_callback_digest_md5_set, 138
- Gsasl_server_callback_external, 119
- gsasl_server_callback_external_get, 138
- gsasl_server_callback_external_set, 138
- Gsasl_server_callback_gssapi, 119
- gsasl_server_callback_gssapi_get, 139
- gsasl_server_callback_gssapi_set, 139
- Gsasl_server_callback_maxbuf, 120
- gsasl_server_callback_maxbuf_get, 139
- gsasl_server_callback_maxbuf_set, 140
- Gsasl_server_callback_qop, 120
- gsasl_server_callback_qop_get, 140
- gsasl_server_callback_qop_set, 140
- Gsasl_server_callback_realm, 120
- gsasl_server_callback_realm_get, 141
- gsasl_server_callback_realm_set, 141
- Gsasl_server_callback_retrieve, 120
- gsasl_server_callback_retrieve_get, 142
- gsasl_server_callback_retrieve_set, 142
- Gsasl_server_callback_securid, 120
- gsasl_server_callback_securid_get, 142
- gsasl_server_callback_securid_set, 143
- Gsasl_server_callback_service, 120
- gsasl_server_callback_service_get, 143
- gsasl_server_callback_service_set, 143
- Gsasl_server_callback_validate, 120
- gsasl_server_callback_validate_get, 144
- gsasl_server_callback_validate_set, 144
- gsasl_server_ctx_get, 144
- gsasl_server_finish, 145
- gsasl_server_listmech, 145
- gsasl_server_step, 145
- gsasl_server_step_base64, 146
- gsasl_server_suggest_mechanism, 146
- gsasl_stringprep_nfkc, 146
- gsasl_stringprep_saslprep, 147
- gsasl_stringprep_trace, 147
- GSASL_TOO_SMALL_BUFFER, 120
- GSASL_UNICODE_NORMALIZATION_-ERROR, 121
- gsasl-mech.h, 149
 - Gsasl_code_function, 149
 - Gsasl_done_function, 149
 - Gsasl_finish_function, 149
 - Gsasl_init_function, 149
 - Gsasl_mechanism, 149
 - Gsasl_mechanism_functions, 149
 - gsasl_register, 150
 - Gsasl_start_function, 150
 - Gsasl_step_function, 150
- gsasl.h, 151
 - Gsasl, 153
 - GSASL_ALLOW_UNASSIGNED, 156
 - GSASL_ANONYMOUS_TOKEN, 154
 - GSASL_AUTHENTICATION_ERROR, 155
 - GSASL_AUTHID, 154
 - GSASL_AUTHZID, 154
 - GSASL_BASE64_ERROR, 155
 - gsasl_base64_from, 156
 - gsasl_base64_to, 156
 - gsasl_callback, 157
 - Gsasl_callback_function, 153
 - gsasl_callback_hook_get, 157
 - gsasl_callback_hook_set, 158
 - gsasl_callback_set, 158
 - gsasl_check_version, 158
 - Gsasl_cipher, 154
 - GSASL_CIPHER_3DES, 154
 - GSASL_CIPHER_AES, 154
 - GSASL_CIPHER_DES, 154
 - GSASL_CIPHER_RC4, 154
 - GSASL_CIPHER_RC4_40, 154
 - GSASL_CIPHER_RC4_56, 154
 - gsasl_client_mechlist, 159
 - gsasl_client_start, 159

- gsasl_client_suggest_mechanism, 159
- gsasl_client_support_p, 160
- GSASL_CRYPTO_ERROR, 155
- gsasl_decode, 160
- gsasl_done, 161
- gsasl_encode, 161
- gsasl_finish, 161
- GSASL_GSSAPI_ACCEPT_SEC_-
CONTEXT_ERROR, 155
- GSASL_GSSAPI_ACQUIRE_CRED_-
ERROR, 155
- GSASL_GSSAPI_DISPLAY_NAME, 154
- GSASL_GSSAPI_DISPLAY_NAME_-
ERROR, 155
- GSASL_GSSAPI_IMPORT_NAME_-
ERROR, 155
- GSASL_GSSAPI_INIT_SEC_CONTEXT_-
ERROR, 155
- GSASL_GSSAPI_RELEASE_BUFFER_-
ERROR, 155
- GSASL_GSSAPI_UNSUPPORTED_-
PROTECTION_ERROR, 155
- GSASL_GSSAPI_UNWRAP_ERROR, 155
- GSASL_GSSAPI_WRAP_ERROR, 155
- gsasl_hmac_md5, 162
- GSASL_HOSTNAME, 154
- gsasl_init, 162
- GSASL_INTEGRITY_ERROR, 155
- GSASL_KERBEROS_V5_INIT_ERROR,
156
- GSASL_KERBEROS_V5_INTERNAL_-
ERROR, 156
- GSASL_MALLOC_ERROR, 155
- GSASL_MAX_MECHANISM_SIZE, 154
- gsasl_md5, 162
- GSASL_MECHANISM_CALLED_TOO_-
MANY_TIMES, 155
- GSASL_MECHANISM_PARSE_ERROR,
155
- GSASL_MIN_MECHANISM_SIZE, 154
- GSASL_NEEDS_MORE, 155
- GSASL_NO_ANONYMOUS_TOKEN, 155
- GSASL_NO_AUTHID, 155
- GSASL_NO_AUTHZID, 155
- GSASL_NO_CALLBACK, 155
- GSASL_NO_CLIENT_CODE, 155
- GSASL_NO_HOSTNAME, 155
- GSASL_NO_PASSCODE, 155
- GSASL_NO_PASSWORD, 155
- GSASL_NO_PIN, 155
- GSASL_NO_SERVER_CODE, 155
- GSASL_NO_SERVICE, 155
- gsasl_nonce, 163
- GSASL_OK, 155
- GSASL_PASSCODE, 154
- GSASL_PASSWORD, 154
- GSASL_PIN, 154
- Gsasl_property, 154
- gsasl_property_fast, 163
- gsasl_property_get, 163
- gsasl_property_set, 164
- gsasl_property_set_raw, 165
- Gsasl_qop, 154
- GSASL_QOP_AUTH, 155
- GSASL_QOP_AUTH_CONF, 155
- GSASL_QOP_AUTH_INT, 155
- gsasl_random, 165
- Gsasl_rc, 155
- GSASL_REALM, 154
- gsasl_saslprep, 165
- GSASL_SASLPREP_ERROR, 155
- Gsasl_saslprep_flags, 156
- GSASL_SECURID_SERVER_NEED_-
ADDITIONAL_PASSCODE, 156
- GSASL_SECURID_SERVER_NEED_-
NEW_PIN, 156
- gsasl_server_mechlist, 166
- gsasl_server_start, 166
- gsasl_server_support_p, 166
- GSASL_SERVICE, 154
- Gsasl_session, 153
- gsasl_session_hook_get, 167
- gsasl_session_hook_set, 167
- GSASL_SHISHI_ERROR, 156
- gsasl_simple_getpass, 167
- gsasl_step, 168
- gsasl_step64, 168
- gsasl_strerror, 169
- GSASL_SUGGESTED_PIN, 154
- GSASL_UNKNOWN_MECHANISM, 155
- GSASL_VALID_MECHANISM_-
CHARACTERS, 170
- GSASL_VALIDATE_ANONYMOUS, 154
- GSASL_VALIDATE_EXTERNAL, 154
- GSASL_VALIDATE_GSSAPI, 154
- GSASL_VALIDATE_SECURID, 154
- GSASL_VALIDATE_SIMPLE, 154
- GSASL_VERSION, 153
- GSASL_ALLOW_UNASSIGNED
gsasl.h, 156
- gsasl_anonymous_mechanism
anonymous.h, 68
anonymous/mechinfo.c, 183
- GSASL_ANONYMOUS_NAME
anonymous.h, 67
- GSASL_ANONYMOUS_TOKEN
gsasl.h, 154
- gsasl_appinfo_get

- gsasl-compat.h, 121
- obsolete.c, 196
- gsasl_appinfo_set
 - gsasl-compat.h, 121
 - obsolete.c, 196
- gsasl_application_data_get
 - gsasl-compat.h, 122
 - obsolete.c, 196
- gsasl_application_data_set
 - gsasl-compat.h, 122
 - obsolete.c, 197
- GSASL_AUTHENTICATION_ERROR
 - gsasl.h, 155
- GSASL_AUTHID
 - gsasl.h, 154
- GSASL_AUTHZID
 - gsasl.h, 154
- gsasl_base64_decode
 - gsasl-compat.h, 122
 - obsolete.c, 197
- gsasl_base64_encode
 - gsasl-compat.h, 123
 - obsolete.c, 197
- GSASL_BASE64_ERROR
 - gsasl.h, 155
- gsasl_base64_from
 - base64.c, 69
 - gsasl.h, 156
- gsasl_base64_to
 - base64.c, 69
 - gsasl.h, 156
- gsasl_callback
 - callback.c, 71
 - gsasl.h, 157
- Gsasl_callback_function
 - gsasl.h, 153
- gsasl_callback_hook_get
 - callback.c, 71
 - gsasl.h, 157
- gsasl_callback_hook_set
 - callback.c, 72
 - gsasl.h, 158
- gsasl_callback_set
 - callback.c, 72
 - gsasl.h, 158
- GSASL_CANNOT_GET_CTX
 - gsasl-compat.h, 121
- gsasl_check_version
 - gsasl.h, 158
 - version.c, 267
- Gsasl_cipher
 - gsasl.h, 154
- GSASL_CIPHER_3DES
 - gsasl.h, 154
- GSASL_CIPHER_AES
 - gsasl.h, 154
- GSASL_CIPHER_DES
 - gsasl.h, 154
- GSASL_CIPHER_RC4
 - gsasl.h, 154
- GSASL_CIPHER_RC4_40
 - gsasl.h, 154
- GSASL_CIPHER_RC4_56
 - gsasl.h, 154
- gsasl_client_application_data_get
 - gsasl-compat.h, 123
 - obsolete.c, 198
- gsasl_client_application_data_set
 - gsasl-compat.h, 123
 - obsolete.c, 198
- Gsasl_client_callback_anonymous
 - gsasl-compat.h, 118
- gsasl_client_callback_anonymous_get
 - gsasl-compat.h, 124
 - obsolete.c, 198
- gsasl_client_callback_anonymous_set
 - gsasl-compat.h, 124
 - obsolete.c, 199
- Gsasl_client_callback_authentication_id
 - gsasl-compat.h, 118
- gsasl_client_callback_authentication_id_get
 - gsasl-compat.h, 124
 - obsolete.c, 199
- gsasl_client_callback_authentication_id_set
 - gsasl-compat.h, 125
 - obsolete.c, 199
- Gsasl_client_callback_authorization_id
 - gsasl-compat.h, 118
- gsasl_client_callback_authorization_id_get
 - gsasl-compat.h, 125
 - obsolete.c, 200
- gsasl_client_callback_authorization_id_set
 - gsasl-compat.h, 126
 - obsolete.c, 200
- Gsasl_client_callback_maxbuf
 - gsasl-compat.h, 118
- gsasl_client_callback_maxbuf_get
 - gsasl-compat.h, 126
 - obsolete.c, 201
- gsasl_client_callback_maxbuf_set
 - gsasl-compat.h, 126
 - obsolete.c, 201
- Gsasl_client_callback_passcode
 - gsasl-compat.h, 118
- gsasl_client_callback_passcode_get
 - gsasl-compat.h, 127
 - obsolete.c, 201
- gsasl_client_callback_passcode_set

- gsasl-compat.h, 127
- obsolete.c, 202
- Gsasl_client_callback_password
 - gsasl-compat.h, 118
- gsasl_client_callback_password_get
 - gsasl-compat.h, 127
 - obsolete.c, 202
- gsasl_client_callback_password_set
 - gsasl-compat.h, 128
 - obsolete.c, 202
- Gsasl_client_callback_pin
 - gsasl-compat.h, 119
- gsasl_client_callback_pin_get
 - gsasl-compat.h, 128
 - obsolete.c, 203
- gsasl_client_callback_pin_set
 - gsasl-compat.h, 128
 - obsolete.c, 203
- Gsasl_client_callback_qop
 - gsasl-compat.h, 119
- gsasl_client_callback_qop_get
 - gsasl-compat.h, 129
 - obsolete.c, 203
- gsasl_client_callback_qop_set
 - gsasl-compat.h, 129
 - obsolete.c, 204
- Gsasl_client_callback_realm
 - gsasl-compat.h, 119
- gsasl_client_callback_realm_get
 - gsasl-compat.h, 129
 - obsolete.c, 204
- gsasl_client_callback_realm_set
 - gsasl-compat.h, 130
 - obsolete.c, 204
- Gsasl_client_callback_service
 - gsasl-compat.h, 119
- gsasl_client_callback_service_get
 - gsasl-compat.h, 130
 - obsolete.c, 205
- gsasl_client_callback_service_set
 - gsasl-compat.h, 130
 - obsolete.c, 205
- gsasl_client_ctx_get
 - gsasl-compat.h, 131
 - obsolete.c, 205
- gsasl_client_finish
 - gsasl-compat.h, 131
 - obsolete.c, 206
- gsasl_client_listmech
 - gsasl-compat.h, 131
 - obsolete.c, 206
- gsasl_client_mechlist
 - gsasl.h, 159
 - listmech.c, 179
- gsasl_client_start
 - gsasl.h, 159
 - xstart.c, 276
- gsasl_client_step
 - gsasl-compat.h, 132
 - obsolete.c, 206
- gsasl_client_step_base64
 - gsasl-compat.h, 132
 - obsolete.c, 207
- gsasl_client_suggest_mechanism
 - gsasl.h, 159
 - suggest.c, 257
- gsasl_client_support_p
 - gsasl.h, 160
 - supportp.c, 258
- Gsasl_code_function
 - gsasl-mech.h, 149
- gsasl_cram_md5_mechanism
 - cram-md5.h, 93
 - cram-md5/mechinfo.c, 184
- GSASL_CRAM_MD5_NAME
 - cram-md5.h, 92
- GSASL_CRYPTO_ERROR
 - gsasl.h, 155
- gsasl_ctx_get
 - gsasl-compat.h, 133
 - obsolete.c, 207
- gsasl_decode
 - gsasl.h, 160
 - xcode.c, 273
- gsasl_decode_inline
 - gsasl-compat.h, 133
 - obsolete.c, 207
- gsasl_digest_md5_mechanism
 - digest-md5.h, 98
 - digest-md5/mechinfo.c, 185
- GSASL_DIGEST_MD5_NAME
 - digest-md5.h, 96
- gsasl_done
 - done.c, 106
 - gsasl.h, 161
- Gsasl_done_function
 - gsasl-mech.h, 149
- gsasl_encode
 - gsasl.h, 161
 - xcode.c, 273
- gsasl_encode_inline
 - gsasl-compat.h, 133
 - obsolete.c, 208
- gsasl_external_mechanism
 - external.h, 110
 - external/mechinfo.c, 186
- GSASL_EXTERNAL_NAME
 - external.h, 110

- GSASL_FCLOSE_ERROR
 - gsasl-compat.h, 120
- gsasl_finish
 - gsasl.h, 161
 - xfinish.c, 275
- Gsasl_finish_function
 - gsasl-mech.h, 149
- GSASL_FOPEN_ERROR
 - gsasl-compat.h, 120
- GSASL_GCRYPT_ERROR
 - gsasl-compat.h, 121
- GSASL_GSSAPI_ACCEPT_SEC_CONTEXT_-
ERROR
 - gsasl.h, 155
- GSASL_GSSAPI_ACQUIRE_CRED_ERROR
 - gsasl.h, 155
- GSASL_GSSAPI_DISPLAY_NAME
 - gsasl.h, 154
- GSASL_GSSAPI_DISPLAY_NAME_ERROR
 - gsasl.h, 155
- GSASL_GSSAPI_IMPORT_NAME_ERROR
 - gsasl.h, 155
- GSASL_GSSAPI_INIT_SEC_CONTEXT_-
ERROR
 - gsasl.h, 155
- gsasl_gssapi_mechanism
 - gssapi/mechinfo.c, 187
 - x-gssapi.h, 270
- GSASL_GSSAPI_NAME
 - x-gssapi.h, 268
- GSASL_GSSAPI_RELEASE_BUFFER_ERROR
 - gsasl.h, 155
- GSASL_GSSAPI_UNSUPPORTED_-
PROTECTION_ERROR
 - gsasl.h, 155
- GSASL_GSSAPI_UNWRAP_ERROR
 - gsasl.h, 155
- GSASL_GSSAPI_WRAP_ERROR
 - gsasl.h, 155
- gsasl_hmac_md5
 - crypto.c, 94
 - gsasl.h, 162
- GSASL_HOSTNAME
 - gsasl.h, 154
- gsasl_init
 - gsasl.h, 162
 - init.c, 171
- Gsasl_init_function
 - gsasl-mech.h, 149
- GSASL_INTEGRITY_ERROR
 - gsasl.h, 155
- GSASL_INVALID_HANDLE
 - gsasl-compat.h, 121
- GSASL_KERBEROS_V5_INIT_ERROR
 - gsasl.h, 156
- GSASL_KERBEROS_V5_INTERNAL_ERROR
 - gsasl.h, 156
- gsasl_login_mechanism
 - login.h, 181
 - login/mechinfo.c, 188
- GSASL_LOGIN_NAME
 - login.h, 180
- GSASL_MALLOC_ERROR
 - gsasl.h, 155
- GSASL_MAX_MECHANISM_SIZE
 - gsasl.h, 154
- gsasl_md5
 - crypto.c, 94
 - gsasl.h, 162
- gsasl_md5pwd_get_password
 - gsasl-compat.h, 134
 - obsolete.c, 208
- Gsasl_mechanism, 61
 - client, 61
 - gsasl-mech.h, 149
 - name, 61
 - server, 61
- GSASL_MECHANISM_CALLED_TOO_-
MANY_TIMES
 - gsasl.h, 155
- Gsasl_mechanism_functions, 62
 - decode, 62
 - done, 62
 - encode, 62
 - finish, 62
 - gsasl-mech.h, 149
 - init, 62
 - start, 62
 - step, 63
- GSASL_MECHANISM_PARSE_ERROR
 - gsasl.h, 155
- GSASL_MIN_MECHANISM_SIZE
 - gsasl.h, 154
- GSASL_NEED_CLIENT_ANONYMOUS_-
CALLBACK
 - gsasl-compat.h, 121
- GSASL_NEED_CLIENT_AUTHENTICATION_-
ID_CALLBACK
 - gsasl-compat.h, 121
- GSASL_NEED_CLIENT_AUTHORIZATION_-
ID_CALLBACK
 - gsasl-compat.h, 121
- GSASL_NEED_CLIENT_PASSCODE_-
CALLBACK
 - gsasl-compat.h, 121
- GSASL_NEED_CLIENT_PASSWORD_-
CALLBACK
 - gsasl-compat.h, 121

- GSASL_NEED_CLIENT_PIN_CALLBACK
 - gsasl-compat.h, 121
- GSASL_NEED_CLIENT_SERVICE_-CALLBACK
 - gsasl-compat.h, 121
- GSASL_NEED_SERVER_ANONYMOUS_-CALLBACK
 - gsasl-compat.h, 121
- GSASL_NEED_SERVER_CRAM_MD5_-CALLBACK
 - gsasl-compat.h, 121
- GSASL_NEED_SERVER_DIGEST_MD5_-CALLBACK
 - gsasl-compat.h, 121
- GSASL_NEED_SERVER_EXTERNAL_-CALLBACK
 - gsasl-compat.h, 121
- GSASL_NEED_SERVER_GSSAPI_CALLBACK
 - gsasl-compat.h, 121
- GSASL_NEED_SERVER_REALM_CALLBACK
 - gsasl-compat.h, 121
- GSASL_NEED_SERVER_RETRIEVE_-CALLBACK
 - gsasl-compat.h, 121
- GSASL_NEED_SERVER_SECURID_-CALLBACK
 - gsasl-compat.h, 121
- GSASL_NEED_SERVER_SERVICE_-CALLBACK
 - gsasl-compat.h, 121
- GSASL_NEED_SERVER_VALIDATE_-CALLBACK
 - gsasl-compat.h, 121
- GSASL_NEEDS_MORE
 - gsasl.h, 155
- GSASL_NO_ANONYMOUS_TOKEN
 - gsasl.h, 155
- GSASL_NO_AUTHID
 - gsasl.h, 155
- GSASL_NO_AUTHZID
 - gsasl.h, 155
- GSASL_NO_CALLBACK
 - gsasl.h, 155
- GSASL_NO_CLIENT_CODE
 - gsasl.h, 155
- GSASL_NO_HOSTNAME
 - gsasl.h, 155
- GSASL_NO_MORE_REALMS
 - gsasl-compat.h, 121
- GSASL_NO_PASSCODE
 - gsasl.h, 155
- GSASL_NO_PASSWORD
 - gsasl.h, 155
- GSASL_NO_PIN
 - gsasl.h, 155
- GSASL_NO_SERVER_CODE
 - gsasl.h, 155
- GSASL_NO_SERVICE
 - gsasl.h, 155
- gsasl_nonce
 - crypto.c, 94
 - gsasl.h, 163
- gsasl_ntlm_mechanism
 - ntlm/mechinfo.c, 189
 - x-ntlm.h, 272
- GSASL_NTLM_NAME
 - x-ntlm.h, 271
- GSASL_OK
 - gsasl.h, 155
- GSASL_PASSCODE
 - gsasl.h, 154
- GSASL_PASSWORD
 - gsasl.h, 154
- GSASL_PIN
 - gsasl.h, 154
- gsasl_plain_mechanism
 - plain.h, 228
 - plain/mechinfo.c, 190
- GSASL_PLAIN_NAME
 - plain.h, 227
- Gsasl_property
 - gsasl.h, 154
- gsasl_property_fast
 - gsasl.h, 163
 - property.c, 231
- gsasl_property_get
 - gsasl.h, 163
 - property.c, 231
- gsasl_property_set
 - gsasl.h, 164
 - property.c, 232
- gsasl_property_set_raw
 - gsasl.h, 165
 - property.c, 232
- Gsasl_qop
 - gsasl.h, 154
- GSASL_QOP_AUTH
 - gsasl.h, 155
- GSASL_QOP_AUTH_CONF
 - gsasl.h, 155
- GSASL_QOP_AUTH_INT
 - gsasl.h, 155
- gsasl_random
 - crypto.c, 95
 - gsasl.h, 165
- gsasl_randomize
 - gsasl-compat.h, 134
 - obsolete.c, 209

- Gsasl_rc
 - gsasl.h, 155
- GSASL_REALM
 - gsasl.h, 154
- gsasl_register
 - gsasl-mech.h, 150
 - register.c, 234
- gsasl_saslprep
 - gsasl.h, 165
 - saslprep.c, 235
- GSASL_SASLPREP_ERROR
 - gsasl.h, 155
- Gsasl_saslprep_flags
 - gsasl.h, 156
- gsasl_securid_mechanism
 - securid.h, 237
 - securid/mechinfo.c, 191
- GSASL_SECURID_NAME
 - securid.h, 236
- GSASL_SECURID_SERVER_NEED_-
ADDITIONAL_PASSCODE
 - gsasl.h, 156
- GSASL_SECURID_SERVER_NEED_NEW_PIN
 - gsasl.h, 156
- gsasl_server_application_data_get
 - gsasl-compat.h, 135
 - obsolete.c, 209
- gsasl_server_application_data_set
 - gsasl-compat.h, 135
 - obsolete.c, 210
- Gsasl_server_callback_anonymous
 - gsasl-compat.h, 119
- gsasl_server_callback_anonymous_get
 - gsasl-compat.h, 135
 - obsolete.c, 210
- gsasl_server_callback_anonymous_set
 - gsasl-compat.h, 136
 - obsolete.c, 210
- Gsasl_server_callback_cipher
 - gsasl-compat.h, 119
- gsasl_server_callback_cipher_get
 - gsasl-compat.h, 136
 - obsolete.c, 211
- gsasl_server_callback_cipher_set
 - gsasl-compat.h, 136
 - obsolete.c, 211
- Gsasl_server_callback_cram_md5
 - gsasl-compat.h, 119
- gsasl_server_callback_cram_md5_get
 - gsasl-compat.h, 137
 - obsolete.c, 211
- gsasl_server_callback_cram_md5_set
 - gsasl-compat.h, 137
 - obsolete.c, 212
- Gsasl_server_callback_digest_md5
 - gsasl-compat.h, 119
- gsasl_server_callback_digest_md5_get
 - gsasl-compat.h, 137
 - obsolete.c, 212
- gsasl_server_callback_digest_md5_set
 - gsasl-compat.h, 138
 - obsolete.c, 212
- Gsasl_server_callback_external
 - gsasl-compat.h, 119
- gsasl_server_callback_external_get
 - gsasl-compat.h, 138
 - obsolete.c, 213
- gsasl_server_callback_external_set
 - gsasl-compat.h, 138
 - obsolete.c, 213
- Gsasl_server_callback_gssapi
 - gsasl-compat.h, 119
- gsasl_server_callback_gssapi_get
 - gsasl-compat.h, 139
 - obsolete.c, 213
- gsasl_server_callback_gssapi_set
 - gsasl-compat.h, 139
 - obsolete.c, 214
- Gsasl_server_callback_maxbuf
 - gsasl-compat.h, 120
- gsasl_server_callback_maxbuf_get
 - gsasl-compat.h, 139
 - obsolete.c, 214
- gsasl_server_callback_maxbuf_set
 - gsasl-compat.h, 140
 - obsolete.c, 214
- Gsasl_server_callback_qop
 - gsasl-compat.h, 120
- gsasl_server_callback_qop_get
 - gsasl-compat.h, 140
 - obsolete.c, 215
- gsasl_server_callback_qop_set
 - gsasl-compat.h, 140
 - obsolete.c, 215
- Gsasl_server_callback_realm
 - gsasl-compat.h, 120
- gsasl_server_callback_realm_get
 - gsasl-compat.h, 141
 - obsolete.c, 215
- gsasl_server_callback_realm_set
 - gsasl-compat.h, 141
 - obsolete.c, 216
- Gsasl_server_callback_retrieve
 - gsasl-compat.h, 120
- gsasl_server_callback_retrieve_get
 - gsasl-compat.h, 142
 - obsolete.c, 216
- gsasl_server_callback_retrieve_set

- gsasl-compat.h, 142
- obsolete.c, 216
- Gsasl_server_callback_securid
 - gsasl-compat.h, 120
- gsasl_server_callback_securid_get
 - gsasl-compat.h, 142
 - obsolete.c, 217
- gsasl_server_callback_securid_set
 - gsasl-compat.h, 143
 - obsolete.c, 217
- Gsasl_server_callback_service
 - gsasl-compat.h, 120
- gsasl_server_callback_service_get
 - gsasl-compat.h, 143
 - obsolete.c, 218
- gsasl_server_callback_service_set
 - gsasl-compat.h, 143
 - obsolete.c, 218
- Gsasl_server_callback_validate
 - gsasl-compat.h, 120
- gsasl_server_callback_validate_get
 - gsasl-compat.h, 144
 - obsolete.c, 218
- gsasl_server_callback_validate_set
 - gsasl-compat.h, 144
 - obsolete.c, 219
- gsasl_server_ctx_get
 - gsasl-compat.h, 144
 - obsolete.c, 219
- gsasl_server_finish
 - gsasl-compat.h, 145
 - obsolete.c, 219
- gsasl_server_listmech
 - gsasl-compat.h, 145
 - obsolete.c, 220
- gsasl_server_mechlist
 - gsasl.h, 166
 - listmech.c, 179
- gsasl_server_start
 - gsasl.h, 166
 - xstart.c, 276
- gsasl_server_step
 - gsasl-compat.h, 145
 - obsolete.c, 220
- gsasl_server_step_base64
 - gsasl-compat.h, 146
 - obsolete.c, 220
- gsasl_server_suggest_mechanism
 - gsasl-compat.h, 146
 - obsolete.c, 221
- gsasl_server_support_p
 - gsasl.h, 166
 - supportp.c, 258
- GSASL_SERVICE
 - gsasl.h, 154
- Gsasl_session, 64
 - anonymous_token, 64
 - application_data, 64
 - application_hook, 64
 - authid, 64
 - authzid, 65
 - clientp, 65
 - ctx, 65
 - gsasl.h, 153
 - gssapi_display_name, 65
 - hostname, 65
 - mech, 65
 - mech_data, 65
 - passcode, 65
 - password, 66
 - pin, 66
 - realm, 66
 - service, 66
 - suggestedpin, 66
- gsasl_session_hook_get
 - callback.c, 73
 - gsasl.h, 167
- gsasl_session_hook_set
 - callback.c, 73
 - gsasl.h, 167
- GSASL_SHISHI_ERROR
 - gsasl.h, 156
- gsasl_simple_getpass
 - gsasl.h, 167
 - md5pwd.c, 182
- Gsasl_start_function
 - gsasl-mech.h, 150
- gsasl_step
 - gsasl.h, 168
 - xstep.c, 277
- gsasl_step64
 - gsasl.h, 168
 - xstep.c, 277
- Gsasl_step_function
 - gsasl-mech.h, 150
- gsasl_strerror
 - error.c, 108
 - gsasl.h, 169
- gsasl_stringprep_nfkc
 - gsasl-compat.h, 146
 - obsolete.c, 221
- gsasl_stringprep_saslprep
 - gsasl-compat.h, 147
 - obsolete.c, 222
- gsasl_stringprep_trace
 - gsasl-compat.h, 147
 - obsolete.c, 222
- GSASL_SUGGESTED_PIN

- gsasl.h, 154
- GSASL_TOO_SMALL_BUFFER
 - gsasl-compat.h, 120
- GSASL_UNICODE_NORMALIZATION_ERROR
 - gsasl-compat.h, 121
- GSASL_UNKNOWN_MECHANISM
 - gsasl.h, 155
- GSASL_VALID_MECHANISM_CHARACTERS
 - gsasl.h, 170
 - init.c, 171
- GSASL_VALIDATE_ANONYMOUS
 - gsasl.h, 154
- GSASL_VALIDATE_EXTERNAL
 - gsasl.h, 154
- GSASL_VALIDATE_GSSAPI
 - gsasl.h, 154
- GSASL_VALIDATE_SECURID
 - gsasl.h, 154
- GSASL_VALIDATE_SIMPLE
 - gsasl.h, 154
- GSASL_VERSION
 - gsasl.h, 153
- gssapi/client.c
 - _Gsasl_gssapi_client_state, 83
 - _gsasl_gssapi_client_decode, 83
 - _gsasl_gssapi_client_encode, 83
 - _gsasl_gssapi_client_finish, 84
 - _gsasl_gssapi_client_start, 84
 - _gsasl_gssapi_client_step, 84
- gssapi/mechinfo.c
 - gsasl_gssapi_mechanism, 187
- gssapi/server.c
 - _Gsasl_gssapi_server_state, 244
 - _gsasl_gssapi_server_finish, 244
 - _gsasl_gssapi_server_start, 244
 - _gsasl_gssapi_server_step, 244
- gssapi_display_name
 - Gsasl_session, 65
- HEXCHAR
 - digest.c, 99
 - digesthmac.c, 103
- hostname
 - Gsasl_session, 65
- init
 - Gsasl_mechanism_functions, 62
- init.c, 171
 - gsasl_init, 171
 - GSASL_VALID_MECHANISM_CHARACTERS, 171
- internal.h, 173
- kcc
 - _Gsasl_digest_md5_client_state, 33
 - _Gsasl_digest_md5_server_state, 36
- kcs
 - _Gsasl_digest_md5_client_state, 34
 - _Gsasl_digest_md5_server_state, 36
- kerberos_v5.c, 174
 - BITMAP_LEN, 174
 - CLIENT_HELLO_LEN, 174
 - DEBUG, 174
 - MAXBUF_DEFAULT, 174
 - MAXBUF_LEN, 174
 - MUTUAL, 174
 - RANDOM_LEN, 174
 - SERVER_HELLO_LEN, 175
- kerberos_v5.h, 176
 - _GSASL_KERBEROS_V5_NAME, 176
 - _gsasl_kerberos_v5_client_decode, 176
 - _gsasl_kerberos_v5_client_encode, 176
 - _gsasl_kerberos_v5_client_finish, 177
 - _gsasl_kerberos_v5_client_init, 177
 - _gsasl_kerberos_v5_client_start, 177
 - _gsasl_kerberos_v5_client_step, 177
 - _gsasl_kerberos_v5_server_decode, 177
 - _gsasl_kerberos_v5_server_encode, 177
 - _gsasl_kerberos_v5_server_finish, 178
 - _gsasl_kerberos_v5_server_init, 178
 - _gsasl_kerberos_v5_server_start, 178
 - _gsasl_kerberos_v5_server_step, 178
- kerberos_v5/client.c
 - _gsasl_kerberos_v5_client_decode, 86
 - _gsasl_kerberos_v5_client_encode, 86
 - _gsasl_kerberos_v5_client_finish, 86
 - _gsasl_kerberos_v5_client_init, 86
 - _gsasl_kerberos_v5_client_start, 86
 - _gsasl_kerberos_v5_client_step, 87
 - STEP_FIRST, 85
 - STEP_NONINFRA_SEND_APREQ, 85
 - STEP_NONINFRA_SEND_ASREQ, 85
 - STEP_NONINFRA_WAIT_APREP, 85
 - STEP_NONINFRA_WAIT_ASREP, 86
 - STEP_SUCCESS, 86
- kerberos_v5/server.c
 - _gsasl_kerberos_v5_server_decode, 246
 - _gsasl_kerberos_v5_server_encode, 246
 - _gsasl_kerberos_v5_server_finish, 246
 - _gsasl_kerberos_v5_server_init, 246
 - _gsasl_kerberos_v5_server_start, 247
 - _gsasl_kerberos_v5_server_step, 247
- kic
 - _Gsasl_digest_md5_client_state, 34
 - _Gsasl_digest_md5_server_state, 36
- kis
 - _Gsasl_digest_md5_client_state, 34

- `_Gsasl_digest_md5_server_state`, 36
- listmech.c, 179
 - `gsasl_client_mechlist`, 179
 - `gsasl_server_mechlist`, 179
- login.h, 180
 - `_gsasl_login_client_finish`, 180
 - `_gsasl_login_client_start`, 180
 - `_gsasl_login_client_step`, 180
 - `_gsasl_login_server_finish`, 180
 - `_gsasl_login_server_start`, 181
 - `_gsasl_login_server_step`, 181
 - `gsasl_login_mechanism`, 181
 - `GSASL_LOGIN_NAME`, 180
- login/client.c
 - `_gsasl_login_client_finish`, 88
 - `_gsasl_login_client_start`, 88
 - `_gsasl_login_client_step`, 88
- login/mechinfo.c
 - `gsasl_login_mechanism`, 188
- login/server.c
 - `_gsasl_login_server_finish`, 248
 - `_gsasl_login_server_start`, 248
 - `_gsasl_login_server_step`, 248
 - `CHALLENGE_PASSWORD`, 248
 - `CHALLENGE_USERNAME`, 248
- MAC_DATA_LEN
 - session.c, 252
- MAC_HMAC_LEN
 - session.c, 252
- MAC_MSG_TYPE
 - session.c, 252
- MAC_MSG_TYPE_LEN
 - session.c, 253
- MAC_SEQNUM_LEN
 - session.c, 253
- main
 - test-gc-hmac-md5.c, 259
 - test-gc-md5.c, 260
 - test-gc.c, 261
 - test-parser.c, 262
- MAXBUF_DEFAULT
 - kerberos_v5.c, 174
 - shared.h, 255
- MAXBUF_LEN
 - kerberos_v5.c, 174
 - shared.h, 255
- MD5LEN
 - cram-md5/server.c, 239
 - digesthmac.c, 103
 - session.c, 253
- md5pwd.c, 182
 - `gsasl_simple_getpass`, 182
- mech
 - `Gsasl_session`, 65
- mech_data
 - `Gsasl_session`, 65
- mechinfo.c, 183–191
- MUTUAL
 - kerberos_v5.c, 174
 - shared.h, 255
- N_
 - error.c, 108
- n_client_mechs
 - `Gsasl`, 59
- n_server_mechs
 - `Gsasl`, 60
- name
 - `Gsasl_mechanism`, 61
- nc
 - `digest_md5_response`, 54
- nonce
 - `digest_md5_challenge`, 50
 - `digest_md5_response`, 54
- NONCE_ENTROPY_BYTES
 - `digest-md5/server.c`, 240
- NONCELEN
 - challenge.c, 74
- nrealms
 - `digest_md5_challenge`, 50
- ntlm.c, 192
 - `_Gsasl_ntlm_state`, 192
 - `_gsasl_ntlm_client_finish`, 192
 - `_gsasl_ntlm_client_start`, 192
 - `_gsasl_ntlm_client_step`, 192
- ntlm/mechinfo.c
 - `gsasl_ntlm_mechanism`, 189
- obsolete.c, 194
 - Assert, 196
 - `gsasl_appinfo_get`, 196
 - `gsasl_appinfo_set`, 196
 - `gsasl_application_data_get`, 196
 - `gsasl_application_data_set`, 197
 - `gsasl_base64_decode`, 197
 - `gsasl_base64_encode`, 197
 - `gsasl_client_application_data_get`, 198
 - `gsasl_client_application_data_set`, 198
 - `gsasl_client_callback_anonymous_get`, 198
 - `gsasl_client_callback_anonymous_set`, 199
 - `gsasl_client_callback_authentication_id_get`, 199
 - `gsasl_client_callback_authentication_id_set`, 199
 - `gsasl_client_callback_authorization_id_get`, 200

- gsasl_client_callback_authorization_id_set, 200
- gsasl_client_callback_maxbuf_get, 201
- gsasl_client_callback_maxbuf_set, 201
- gsasl_client_callback_passcode_get, 201
- gsasl_client_callback_passcode_set, 202
- gsasl_client_callback_password_get, 202
- gsasl_client_callback_password_set, 202
- gsasl_client_callback_pin_get, 203
- gsasl_client_callback_pin_set, 203
- gsasl_client_callback_qop_get, 203
- gsasl_client_callback_qop_set, 204
- gsasl_client_callback_realm_get, 204
- gsasl_client_callback_realm_set, 204
- gsasl_client_callback_service_get, 205
- gsasl_client_callback_service_set, 205
- gsasl_client_ctx_get, 205
- gsasl_client_finish, 206
- gsasl_client_listmech, 206
- gsasl_client_step, 206
- gsasl_client_step_base64, 207
- gsasl_ctx_get, 207
- gsasl_decode_inline, 207
- gsasl_encode_inline, 208
- gsasl_md5pwd_get_password, 208
- gsasl_randomize, 209
- gsasl_server_application_data_get, 209
- gsasl_server_application_data_set, 210
- gsasl_server_callback_anonymous_get, 210
- gsasl_server_callback_anonymous_set, 210
- gsasl_server_callback_cipher_get, 211
- gsasl_server_callback_cipher_set, 211
- gsasl_server_callback_cram_md5_get, 211
- gsasl_server_callback_cram_md5_set, 212
- gsasl_server_callback_digest_md5_get, 212
- gsasl_server_callback_digest_md5_set, 212
- gsasl_server_callback_external_get, 213
- gsasl_server_callback_external_set, 213
- gsasl_server_callback_gssapi_get, 213
- gsasl_server_callback_gssapi_set, 214
- gsasl_server_callback_maxbuf_get, 214
- gsasl_server_callback_maxbuf_set, 214
- gsasl_server_callback_qop_get, 215
- gsasl_server_callback_qop_set, 215
- gsasl_server_callback_realm_get, 215
- gsasl_server_callback_realm_set, 216
- gsasl_server_callback_retrieve_get, 216
- gsasl_server_callback_retrieve_set, 216
- gsasl_server_callback_secured_get, 217
- gsasl_server_callback_secured_set, 217
- gsasl_server_callback_service_get, 218
- gsasl_server_callback_service_set, 218
- gsasl_server_callback_validate_get, 218
- gsasl_server_callback_validate_set, 219
- gsasl_server_ctx_get, 219
- gsasl_server_finish, 219
- gsasl_server_listmech, 220
- gsasl_server_step, 220
- gsasl_server_step_base64, 220
- gsasl_server_suggest_mechanism, 221
- gsasl_stringprep_nfkc, 221
- gsasl_stringprep_saslprep, 222
- gsasl_stringprep_trace, 222
- parser.c, 223
 - CHALLENGE_ALGORITHM, 224
 - CHALLENGE_CHARSET, 224
 - CHALLENGE_CIPHER, 224
 - CHALLENGE_MAXBUF, 224
 - CHALLENGE_NONCE, 224
 - CHALLENGE_QOP, 224
 - CHALLENGE_REALM, 224
 - CHALLENGE_STALE, 224
 - CIPHER_3DES, 224
 - CIPHER_AES_CBC, 224
 - CIPHER_DES, 224
 - CIPHER_RC4, 224
 - CIPHER_RC4_40, 224
 - CIPHER_RC4_56, 224
 - DEFAULT_ALGORITHM, 223
 - DEFAULT_CHARSET, 223
 - digest_md5_parse_challenge, 225
 - digest_md5_parse_finish, 225
 - digest_md5_parse_response, 225
 - QOP_AUTH, 224
 - QOP_AUTH_CONF, 224
 - QOP_AUTH_INT, 224
 - RESPONSE_AUTHZID, 225
 - RESPONSE_CHARSET, 225
 - RESPONSE_CIPHER, 225
 - RESPONSE_CNONCE, 224
 - RESPONSE_DIGEST_URI, 225
 - RESPONSE_MAXBUF, 225
 - RESPONSE_NC, 224
 - RESPONSE_NONCE, 224
 - RESPONSE_QOP, 225
 - RESPONSE_REALM, 224
 - RESPONSE_RESPONSE, 225
 - RESPONSE_USERNAME, 224
 - RESPONSEAUTH_RSPAUTH, 225
- parser.h, 226
 - digest_md5_getsubopt, 226
 - digest_md5_parse_challenge, 226
 - digest_md5_parse_finish, 226
 - digest_md5_parse_response, 226
- PASSCODE
 - secured/client.c, 90
 - secured/server.c, 251

- passcode
 - Gsasl_session, 65
- password
 - _Gsasl_kerberos_v5_server_state, 44
 - _Gsasl_login_server_state, 48
 - Gsasl_session, 66
- PIN
 - securid/client.c, 90
 - securid/server.c, 251
- pin
 - Gsasl_session, 66
- plain.h, 227
 - _gsasl_plain_client_step, 227
 - _gsasl_plain_server_step, 227
 - gsasl_plain_mechanism, 228
 - GSASL_PLAIN_NAME, 227
- plain/client.c
 - _gsasl_plain_client_step, 89
- plain/mechinfo.c
 - gsasl_plain_mechanism, 190
- plain/server.c
 - _gsasl_plain_server_step, 250
- printer.c, 229
 - digest_md5_print_challenge, 229
 - digest_md5_print_finish, 229
 - digest_md5_print_response, 229
- printer.h, 230
 - digest_md5_print_challenge, 230
 - digest_md5_print_finish, 230
 - digest_md5_print_response, 230
- property.c, 231
 - gsasl_property_fast, 231
 - gsasl_property_get, 231
 - gsasl_property_set, 232
 - gsasl_property_set_raw, 232
- qop
 - _Gsasl_gssapi_client_state, 38
 - digest_md5_response, 54
- QOP_AUTH
 - parser.c, 224
- QOP_AUTH
 - digesthmac.c, 103
- QOP_AUTH_CONF
 - parser.c, 224
- QOP_AUTH_CONF
 - digesthmac.c, 103
- QOP_AUTH_INT
 - parser.c, 224
- QOP_AUTH_INT
 - digesthmac.c, 103
- qops
 - digest_md5_challenge, 50
- random
 - _Gsasl_kerberos_v5_server_state, 44
- RANDOM_LEN
 - kerberos_v5.c, 174
 - shared.h, 255
- readseqnum
 - _Gsasl_digest_md5_client_state, 34
 - _Gsasl_digest_md5_server_state, 37
- realm
 - digest_md5_response, 54
 - Gsasl_session, 66
- realms
 - digest_md5_challenge, 50
- register.c, 234
 - gsasl_register, 234
- response
 - _Gsasl_digest_md5_client_state, 34
 - _Gsasl_digest_md5_server_state, 37
 - digest_md5_response, 54
- RESPONSE_AUTHZID
 - parser.c, 225
- RESPONSE_CHARSET
 - parser.c, 225
- RESPONSE_CIPHER
 - parser.c, 225
- RESPONSE_CNONCE
 - parser.c, 224
- RESPONSE_DIGEST_URI
 - parser.c, 225
- RESPONSE_LENGTH
 - digesthmac.c, 103
- RESPONSE_MAXBUF
 - parser.c, 225
- RESPONSE_NC
 - parser.c, 224
- RESPONSE_NONCE
 - parser.c, 224
- RESPONSE_QOP
 - parser.c, 225
- RESPONSE_REALM
 - parser.c, 224
- RESPONSE_RESPONSE
 - parser.c, 225
- RESPONSE_USERNAME
 - parser.c, 224
- RESPONSEAUTH_RSPAUTH
 - parser.c, 225
- rspauth
 - digest_md5_finish, 52
- RSPAUTH_LENGTH
 - digesthmac.c, 103
- safe
 - _Gsasl_kerberos_v5_client_state, 40

- [_Gsasl_kerberos_v5_server_state, 44](#)
- [SASL_INTEGRITY_PREFIX_LENGTH](#)
- [session.c, 253](#)
- [saslprep.c, 235](#)
- [gsasl_saslprep, 235](#)
- [secret](#)
- [_Gsasl_digest_md5_client_state, 34](#)
 - [_Gsasl_digest_md5_server_state, 37](#)
- [securid.h, 236](#)
- [_gsasl_securid_client_finish, 236](#)
 - [_gsasl_securid_client_start, 236](#)
 - [_gsasl_securid_client_step, 236](#)
 - [_gsasl_securid_server_step, 236](#)
 - [gsasl_securid_mechanism, 237](#)
 - [GSASL_SECURID_NAME, 236](#)
- [securid/client.c](#)
- [_gsasl_securid_client_finish, 90](#)
 - [_gsasl_securid_client_start, 90](#)
 - [_gsasl_securid_client_step, 90](#)
 - [PASSCODE, 90](#)
 - [PIN, 90](#)
- [securid/mechinfo.c](#)
- [gsasl_securid_mechanism, 191](#)
- [securid/server.c](#)
- [_gsasl_securid_server_step, 251](#)
 - [PASSCODE, 251](#)
 - [PIN, 251](#)
- [sendseqnum](#)
- [_Gsasl_digest_md5_client_state, 34](#)
 - [_Gsasl_digest_md5_server_state, 37](#)
- [server](#)
- [Gsasl_mechanism, 61](#)
- [server.c, 238–240, 243, 244, 246, 248, 250, 251](#)
- [SERVER_HELLO_LEN](#)
- [kerberos_v5.c, 175](#)
 - [shared.h, 255](#)
- [server_mechs](#)
- [Gsasl, 60](#)
- [serverhello](#)
- [_Gsasl_kerberos_v5_client_state, 41](#)
 - [_Gsasl_kerberos_v5_server_state, 44](#)
- [serverhostname](#)
- [_Gsasl_kerberos_v5_server_state, 44](#)
- [servermaxbuf](#)
- [_Gsasl_kerberos_v5_client_state, 41](#)
 - [_Gsasl_kerberos_v5_server_state, 44](#)
 - [digest_md5_challenge, 51](#)
- [servermutual](#)
- [_Gsasl_kerberos_v5_client_state, 41](#)
- [serverqops](#)
- [_Gsasl_kerberos_v5_client_state, 41](#)
 - [_Gsasl_kerberos_v5_server_state, 45](#)
- [serverrealm](#)
- [_Gsasl_kerberos_v5_server_state, 45](#)
- [serverservice](#)
- [_Gsasl_kerberos_v5_server_state, 45](#)
- [service](#)
- [_Gsasl_gssapi_client_state, 38](#)
 - [Gsasl_session, 66](#)
- [session.c, 252](#)
- [C2I, 252](#)
 - [digest_md5_decode, 253](#)
 - [digest_md5_encode, 253](#)
 - [MAC_DATA_LEN, 252](#)
 - [MAC_HMAC_LEN, 252](#)
 - [MAC_MSG_TYPE, 252](#)
 - [MAC_MSG_TYPE_LEN, 253](#)
 - [MAC_SEQNUM_LEN, 253](#)
 - [MD5LEN, 253](#)
 - [SASL_INTEGRITY_PREFIX_LENGTH, 253](#)
- [session.h, 254](#)
- [digest_md5_decode, 254](#)
 - [digest_md5_encode, 254](#)
- [sessionkey](#)
- [_Gsasl_kerberos_v5_client_state, 41](#)
 - [_Gsasl_kerberos_v5_server_state, 45](#)
- [sessiontkkey](#)
- [_Gsasl_kerberos_v5_server_state, 45](#)
- [sh](#)
- [_Gsasl_kerberos_v5_client_state, 41](#)
 - [_Gsasl_kerberos_v5_server_state, 45](#)
- [shared.h, 255](#)
- [BITMAP_LEN, 255](#)
 - [CLIENT_HELLO_LEN, 255](#)
 - [DEBUG, 255](#)
 - [MAXBUF_DEFAULT, 255](#)
 - [MAXBUF_LEN, 255](#)
 - [MUTUAL, 255](#)
 - [RANDOM_LEN, 255](#)
 - [SERVER_HELLO_LEN, 255](#)
- [stale](#)
- [digest_md5_challenge, 51](#)
- [start](#)
- [Gsasl_mechanism_functions, 62](#)
- [step](#)
- [_Gsasl_digest_md5_client_state, 34](#)
 - [_Gsasl_digest_md5_server_state, 37](#)
 - [_Gsasl_gssapi_client_state, 38](#)
 - [_Gsasl_gssapi_server_state, 39](#)
 - [_Gsasl_kerberos_v5_client_state, 41](#)
 - [_Gsasl_login_client_state, 47](#)
 - [_Gsasl_login_server_state, 48](#)
 - [_Gsasl_ntlm_state, 49](#)
 - [Gsasl_mechanism_functions, 63](#)
- [STEP_FIRST](#)
- [kerberos_v5/client.c, 85](#)
- [STEP_NONINFRA_SEND_APREQ](#)
- [kerberos_v5/client.c, 85](#)

- STEP_NONINFRA_SEND_ASREQ
 - kerberos_v5/client.c, 85
- STEP_NONINFRA_WAIT_APREP
 - kerberos_v5/client.c, 85
- STEP_NONINFRA_WAIT_ASREP
 - kerberos_v5/client.c, 86
- STEP_SUCCESS
 - kerberos_v5/client.c, 86
- suggest.c, 257
 - gsasl_client_suggest_mechanism, 257
- suggestedpin
 - Gsasl_session, 66
- supportp.c, 258
 - gsasl_client_support_p, 258
 - gsasl_server_support_p, 258
- TEMPLATE
 - challenge.c, 74
- test-gc-hmac-md5.c, 259
 - main, 259
- test-gc-md5.c, 260
 - main, 260
- test-gc.c, 261
 - main, 261
- test-parser.c, 262
 - main, 262
- tkt
 - _Gsasl_kerberos_v5_client_state, 41
- tokens.h, 263
 - digest_md5_challenge, 264
 - digest_md5_cipher, 264
 - DIGEST_MD5_CIPHER_3DES, 264
 - DIGEST_MD5_CIPHER_AES_CBC, 264
 - DIGEST_MD5_CIPHER_DES, 264
 - DIGEST_MD5_CIPHER_RC4, 264
 - DIGEST_MD5_CIPHER_RC4_40, 264
 - DIGEST_MD5_CIPHER_RC4_56, 264
 - digest_md5_finish, 264
 - DIGEST_MD5_LENGTH, 263
 - digest_md5_qop, 264
 - DIGEST_MD5_QOP_AUTH, 264
 - DIGEST_MD5_QOP_AUTH_CONF, 264
 - DIGEST_MD5_QOP_AUTH_INT, 264
 - digest_md5_response, 264
 - DIGEST_MD5_RESPONSE_LENGTH, 263
- userkey
 - _Gsasl_kerberos_v5_server_state, 45
- username
 - _Gsasl_kerberos_v5_server_state, 45
 - _Gsasl_login_server_state, 48
 - digest_md5_response, 54
- userrealm
 - _Gsasl_kerberos_v5_server_state, 46
- utf8
 - digest_md5_challenge, 51
 - digest_md5_response, 54
- validate.c, 265
 - digest_md5_validate, 265
 - digest_md5_validate_challenge, 265
 - digest_md5_validate_finish, 265
 - digest_md5_validate_response, 265
- validate.h, 266
 - digest_md5_validate, 266
 - digest_md5_validate_challenge, 266
 - digest_md5_validate_finish, 266
 - digest_md5_validate_response, 266
- version.c, 267
 - gsasl_check_version, 267
- x-gssapi.h, 268
 - _gsasl_gssapi_client_decode, 268
 - _gsasl_gssapi_client_encode, 268
 - _gsasl_gssapi_client_finish, 269
 - _gsasl_gssapi_client_start, 269
 - _gsasl_gssapi_client_step, 269
 - _gsasl_gssapi_server_finish, 269
 - _gsasl_gssapi_server_start, 269
 - _gsasl_gssapi_server_step, 269
 - gsasl_gssapi_mechanism, 270
 - GSASL_GSSAPI_NAME, 268
- x-ntlm.h, 271
 - _gsasl_ntlm_client_finish, 271
 - _gsasl_ntlm_client_start, 271
 - _gsasl_ntlm_client_step, 271
 - gsasl_ntlm_mechanism, 272
 - GSASL_NTLM_NAME, 271
- xcode.c, 273
 - gsasl_decode, 273
 - gsasl_encode, 273
- xfinish.c, 275
 - gsasl_finish, 275
- xstart.c, 276
 - gsasl_client_start, 276
 - gsasl_server_start, 276
- xstep.c, 277
 - gsasl_step, 277
 - gsasl_step64, 277