

```
Comando . . . . . : CRTRPGMOD
  Emesso da . . . . . : QPGMR

Modulo . . . . . : ENCRYPTR4
  Libreria . . . . . : LIBHTTP
  Testo 'Descrizione' . . . . . : *SRCMBRTXT

Membro origine . . . . . : ENCRYPTR4
File origine . . . . . : QRPGLSRC
  Libreria . . . . . : LIBHTTP
  CCSID . . . . . : 37
  Testo 'Descrizione' . . . . . : Crypto Routines
  Ultima modifica . . . . . : 12/01/15 12:29:33

Livello di gravit{ generazione . : 10
Opzioni compilatore . . . . . : *XREF *GEN *NOSECLVL *SHOWCPY
  *EXPDDS *EXT *NOSHOWSKP *NOSRCSTMT
  *DEBUGIO *NOEVENTF

Visualizzazione debug . . . . . : *LIST
Emissione . . . . . : *PRINT
Livello di ottimizzazione . . . . : *NONE
Rientro listato origine . . . . . : *NONE
Opzioni conversione tipo . . . . . : *NONE
Sequenza di ordinamento . . . . . : *HEX
Identificativo lingua . . . . . : *JOB RUN
Sostituzione modulo . . . . . : *YES
Autorizzazione . . . . . : *LIBCRTAUT
Troncamento numerico . . . . . : *YES
Correzione campo numerico . . . . . : *NONE
Release di destinazione . . . . . : *CURRENT
Consentire valori nulli . . . . . : *NO
Indirizzario bind . . . . . : *NONE
Definiz. nomi condizione . . . . . : *NONE
Abilitaz. raccolta prestazioni . : *PEP
Dati modellamento . . . . . : *NOCOL
Opzioni LIC (Licensed Internal Code):
Creazione interfaccia programma . : *NO
Includere indirizzario . . . . . :
```

Riga	<----- Specifiche origine ----->	Commenti	Es	Pag.	Modif.	OrigSeq.
Numero1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8.....+.....9.....+.....10	Num Riga	Data	Id	Numero	

List a t o o r i g i n e

```

1 /*- + 150102 000100
2 * Copyright (c) 2012-2015 Thomas Raddatz + 150102 000200
3 * All rights reserved. + 150102 000300
4 * + 150102 000400
5 * Redistribution and use in source and binary forms, with or without + 150102 000500
6 * modification, are permitted provided that the following conditions + 150102 000600
7 * are met: + 150102 000700
8 * 1. Redistributions of source code must retain the above copyright + 150102 000800
9 * notice, this list of conditions and the following disclaimer. + 150102 000900
10 * 2. Redistributions in binary form must reproduce the above copyright + 150102 001000
11 * notice, this list of conditions and the following disclaimer in the + 150102 001100
12 * documentation and/or other materials provided with the distribution. + 150102 001200
13 * + 150102 001300
14 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS 'AS IS' AND + 150102 001400
15 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE + 150102 001500
16 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE + 150102 001600
17 * ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE + 150102 001700
18 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL + 150102 001800
19 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS + 150102 001900
20 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) + 150102 002000
21 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT + 150102 002100
22 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY + 150102 002200
23 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF + 150102 002300
24 * SUCH DAMAGE. + 150102 002400
25 * + 150102 002500
26 */ + 150102 002600
27 *====* 000000 002700
28 * NTLM: Data encryption and digest services * 000000 002800
29 *====* 000000 002900
30 * Author : Thomas Raddatz * 000000 003000
31 * Date : 28.02.2012 * 000000 003100
32 * E-mail : thomas.raddatz@Tools400.de * 000000 003200
33 * Homepage: www.tools400.de * 000000 003300
34 *====* 000000 003400
35 * History: * 000000 003500
36 * * 000000 003600
37 * Date Name Description * 000000 003700
38 * -----* 000000 003800
39 * * 000000 003900
40 *====* 000000 004000
41 /if defined(HAVE_SRCSTMT_NODEBUGIO) 000000 004100
    LINES EXCLUDED: 1
42 /endif
43 H NOMAIN 000000 004400
44 *====* 000000 004500
45 * 000000 004600
46 * ----- 000000 004700
47 * Type Definitions 000000 004800
48 * ----- 000000 004900
49 * 000000 005000
50 * ----- 000000 005100
51 * Exported prototypes 000000 005200
52 * ----- 000000 005300

```

Riga	Specifiche origine	Commenti	Es	Pag.	Modif.	OrigSeq.
Numero			Num	Riga	Data	Id Numero
53	/DEFINE RC4_INTERNAL_USE				000000	005400
54	/DEFINE MD4_INTERNAL_USE				000000	005500
55	/DEFINE MD5_INTERNAL_USE				000000	005600
56	*				000000	005700
57	/COPY QRPGLSRC,NTLM_H				000000	005800
	-----*					
	* Nome del membro RPG. . . . :	NTLM_H			*	1
	* Nome esterno :	LIBHTTP/QRPGLSRC(NTLM_H)			*	1
	* Ultima modifica. :	02/01/15 15:37:46			*	1
	* Testo 'descrizione'. . . . :	NTLM Header File			*	1
	-----*					
58+	=====*					
59+	* NTLM: Authentication Plugin - Public				*	1000100
60+	=====*					
61+	* Author : Thomas Raddatz				*	1000200
62+	* Date : 21.05.2012				*	1000300
63+	* E-mail : thomas.raddatz@tools400.de				*	1000400
64+	* Homepage: www.tools400.de				*	1000500
65+	*				*	1000600
66+	* HTTPAPI : 1.26				*	1000700
67+	=====*					
68+	/IF NOT DEFINED(NTLM_H)				*	1000800
69+	/DEFINE NTLM_H				*	1000900
70+	*				*	1001000
71+	* Returns cTrue if the server requires authentication.				*	1001100
72+	*				*	1001200
					*	1001300
					*	1001400
					*	1001500
73+D	AuthPlugin_isAuthenticationRequired...				000000	1001600
74+D	PR N				000000	1001700
75+D		extproc('NTLMR4_+			000000	1001800
76+D		AuthPlugin_isAuthenticationRequired+			000000	1001900
77+D		')			000000	1002000
78+ *					000000	1002100
79+ *	Returns the realm.				000000	1002200
80+ *					000000	1002300
81+D	AuthPlugin_getRealm...				000000	1002400
82+D	PR 124A	varying			000000	1002500
83+D		extproc('NTLMR4_+			000000	1002600
84+D		AuthPlugin_getRealm+			000000	1002700
85+D		')			000000	1002800
86+ *					000000	1002900
87+ *	Sets the NTLM authentication credentials				000000	1003000
88+ *	Called by procedure http_setAuth() of module HTTPAPIR4.				000000	1003100
89+ *					000000	1003200
90+ *	i_authType = Authentication type used to specify login credentials.				000000	1003300
91+ *	i_username = User name to use.				000000	1003400
92+ *	i_passwd = Password to use.				000000	1003500
93+ *					000000	1003600
94+D	AuthPlugin_setAuthentication...				000000	1003700
95+D	PR N				000000	1003800
96+D		extproc('NTLMR4_+			000000	1003900
97+D		AuthPlugin_setAuthentication+			000000	1004000
98+D		')			000000	1004100
99+D	i_authType	1A	const		000000	1004200
100+D	i_username	80A	const		000000	1004300

Riga	Specifiche origine	Commenti	Es	Pag.	Modif.	OrigSeq.
Numero			Num	Riga	Data	Id Numero
101+D	i_passwd	1024A const			000000	1004400
102+ *					000000	1004500
103+ *	Resets authentication parameters.				000000	1004600
104+ *					000000	1004700
105+D	AuthPlugin_resetAuthentication...				000000	1004800
106+D	PR				000000	1004900
107+D		extproc('NTLMR4_+			000000	1005000
108+D		AuthPlugin_resetAuthentication+			000000	1005100
109+D		')			000000	1005200
110+ *					000000	1005300
111+ *	Interprets a given authentication header.				000000	1005400
112+ *					000000	1005500
113+ *	i_header = Authentication header that must be interpreted.				000000	1005600
114+ *					000000	1005700
115+D	AuthPlugin_interpretAuthenticationHeader...				000000	1005800
116+D	PR				000000	1005900
117+D		extproc('NTLMR4_+			000000	1006000
118+D		AuthPlugin_+			000000	1006100
119+D		interpretAuthenticationHeader+			000000	1006200
120+D		')			000000	1006300
121+D	i_header	2048A const			000000	1006400
122+ *					000000	1006500
123+ *	Returns *ON if, HTTPAPI should receive the the 401 error page and				000000	1006600
124+ *	returns the procedure that is called to receive the error page.				000000	1006700
125+ *					000000	1006800
126+ *	io_saveProc = Procedure pointer of the procedure that is called				000000	1006900
127+ *	to receive the error page.				000000	1007000
128+ *	io_saveFD = File descriptor that is passed to io_saveProc.				000000	1007100
129+ *					000000	1007200
130+D	AuthPlugin_mustReiceAuthErrorPage...				000000	1007300
131+D	PR	N			000000	1007400
132+D		extproc('NTLMR4_+			000000	1007500
133+D		AuthPlugin_mustReiceAuthErrorPage+			000000	1007600
134+D		')			000000	1007700
135+D	io_saveProc	* procptr			000000	1007800
136+D	io_saveFD	10I 0			000000	1007900
137+ *					000000	1008000
138+ *	Negotiates the NTLM authentication parameters with the server and				000000	1008100
139+ *	produces the NTLM authentication header value (type-3) message.				000000	1008200
140+ *	Called by procedure http_persist_req() of module HTTPAPIR4.				000000	1008300
141+ *					000000	1008400
142+ *	i_comm = Pointer to persistent HTTP comm session.				000000	1008500
143+ *	i_URL = URL to GET from or POST with persistent HTTP comm.				000000	1008600
144+ *	i_timeout = Timeout is seconds when no data is received.				000000	1008700
145+ *					000000	1008800
146+D	AuthPlugin_negotiateAuthentication...				000000	1008900
147+D	PR	10I 0			000000	1009000
148+D		extproc('NTLMR4_+			000000	1009100
149+D		AuthPlugin_negotiateAuthentication+			000000	1009200
150+D		')			000000	1009300
151+D	i_comm	* const			000000	1009400
152+D	i_URL	32767A const			000000	1009500
153+D	i_timeout	10I 0 const			000000	1009600
154+ *					000000	1009700
155+ *	Produces the NTLM authentication header when negotiating				000000	1009800
156+ *	the NTLM authentication parameters with the server.				000000	1009900

Riga	Specifiche origine	Commenti	Es	Pag.	Modif.	OrigSeq.
Numero			Num	Riga	Data	Id Numero
157+	*				000000	1010000
158+	*	io_reqChain = HTTP request chain that is send to the server.			000000	1010100
159+	*				000000	1010200
160+D	AuthPlugin_produceAuthenticationHeader...				000000	1010300
161+D	PR				000000	1010400
162+D	extproc('NTLMR4_+				000000	1010500
163+D	AuthPlugin_+				000000	1010600
164+D	produceAuthenticationHeader+				000000	1010700
165+D	')				000000	1010800
166+D	io_reqChain 32767A varying				000000	1010900
167+	*				000000	1011000
168+	/ENDIF				000000	1011100
169	/COPY QRPGLESRC,NTLM_P				000000	005900
-----*						
*	Nome del membro RPG. . . . :	NTLM_P				2
*	Nome esterno :	LIBHTTP/QRPGLESRC(NTLM_P)				2
*	Ultima modifica. :	02/01/15 16:12:17				2
*	Testo 'descrizione'. . . . :	NTML Private Copybook				2
-----*						
170+/*-					150102	2000001
171+	*	Copyright (c) 2012-2015 Thomas Raddatz			150102	2000002
172+	*	All rights reserved.			150102	2000003
173+	*				150102	2000004
174+	*	Redistribution and use in source and binary forms, with or without			150102	2000005
175+	*	modification, are permitted provided that the following conditions			150102	2000006
176+	*	are met:			150102	2000007
177+	*	1. Redistributions of source code must retain the above copyright			150102	2000008
178+	*	notice, this list of conditions and the following disclaimer.			150102	2000009
179+	*	2. Redistributions in binary form must reproduce the above copyright			150102	2000010
180+	*	notice, this list of conditions and the following disclaimer in the			150102	2000011
181+	*	documentation and/or other materials provided with the distribution.			150102	2000012
182+	*				150102	2000013
183+	*	THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS 'AS IS' AND			150102	2000014
184+	*	ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE			150102	2000015
185+	*	IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE			150102	2000016
186+	*	ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE			150102	2000017
187+	*	FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL			150102	2000018
188+	*	DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS			150102	2000019
189+	*	OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)			150102	2000020
190+	*	HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT			150102	2000021
191+	*	LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY			150102	2000022
192+	*	OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF			150102	2000023
193+	*	SUCH DAMAGE.			150102	2000024
194+	*				150102	2000025
195+	*/				150102	2000026
196+	*	=====*			000000	2000100
197+	*	NTLM: Authentication Plugin - Private			000000	2000200
198+	*	=====*			000000	2000300
199+	/IF NOT DEFINED(NTLM_P)				000000	2000400
200+	/DEFINE NTLM_P				000000	2000500
201+	*				000000	2000600
202+	*	Boolean values			000000	2000700
203+D	cTrue	C *on			000000	2000800
204+D	cFalse	C *off			000000	2000900
205+	*				000000	2001000
206+	*	Domain			000000	2001100

Riga	Specifiche origine	Commenti	Es	Pag.	Modif.	OrigSeq.		
Numero	1	2	3	4	5	6		
Numero	7	8	9	10	Num Riga	Data	Id	Numero
207+D	ntlm_domain_t...					000000		2001200
208+D	S	512A	varying		based(pDummy)	000000		2001300
209+ *						000000		2001400
210+ *	Host					000000		2001500
211+D	ntlm_host_t...					000000		2001600
212+D	S	512A	varying		based(pDummy)	000000		2001700
213+ *						000000		2001800
214+ *	Workstation					000000		2001900
215+D	ntlm_workstation_t...					000000		2002000
216+D	S	512A	varying		based(pDummy)	000000		2002100
217+ *						000000		2002200
218+ *	Message					000000		2002300
219+D	ntlm_message_t...					000000		2002400
220+D	S	2048A	varying		based(pDummy)	000000		2002500
221+ *						000000		2002600
222+D	ntlm_challenge_t...					000000		2002700
223+D	S	8A			based(pDummy)	000000		2002800
224+ *						000000		2002900
225+D	ntlm_targetName_t...					000000		2003000
226+D	S	1024C	varying		based(pDummy)	000000		2003100
227+ *						000000		2003200
228+D	ntlm_targetType_t...					000000		2003300
229+D	S	10I 0			based(pDummy)	000000		2003400
230+ *						000000		2003500
231+D	ntlm_user_t...					000000		2003600
232+D	S	124A	varying		based(pDummy)	000000		2003700
233+D	ntlm_password_t...					000000		2003800
234+D	S	1024A	varying		based(pDummy)	000000		2003900
235+D	ntlm_lmResponse_t...					000000		2004000
236+D	S	24A	varying		based(pDummy)	000000		2004100
237+D	ntlm_ntResponse_t...					000000		2004200
238+D	S	24A	varying		based(pDummy)	000000		2004300
239+D	ntlm_ntlmResponse_t...					000000		2004400
240+D	S	2048A	varying		based(pDummy)	000000		2004500
241+D	ntlm_sessionKey_t...					000000		2004600
242+D	S	16A	varying		based(pDummy)	000000		2004700
243+ *						000000		2004800
244+D	errCode_t	DS	qualified		based(pDummy)	000000		2004900
245+D	bytPrv	10I 0				000000		2005000
246+D	bytAvl	10I 0				000000		2005100
247+ *						000000		2005200
248+ /IF NOT DEFINED(qJob_t)					000000		2005300
249+ /DEFINE qJob_t						000000		2005400
250+ *	Qualified job name					000000		2005500
251+D	qJob_t	DS	qualified		based(pDummy)	000000		2005600
252+D	name	10A				000000		2005700
253+D	user	10A				000000		2005800
254+D	nbr	6A				000000		2005900
255+ /ENDIF						000000		2006000
256+ *						000000		2006100
257+D	MILLISECONFS_BETWEEN_1970_AND_1601...					000000		2006200
258+D	C	116444473600000				000000		2006300
259+ *						000000		2006400
260+ *	LM Compatibility Modes:					000000		2006500
261+ *	0 -- Sends NTLMv1 response. That may also include the weak					000000		2006600
262+ *	LM response.					000000		2006700

Riga	Specifiche origine	Commenti	Es	Pag.	Modif.	OrigSeq.
Numero	1...2...3...4...5...6...7...8...9...10	Num Riga	Data	Id	Numero	
263+	* 1 -- Sends only the NTLM response. This is more secure than		000000		2006800	
264+	* Levels 0, because it eliminates the cryptographically-weak		000000		2006900	
265+	* LM response.		000000		2007000	
266+	* 2 -- Sends only the NTLM2 response.		000000		2007100	
267+	* 3 -- Sends LMv2 and NTLMv2 data.		000000		2007200	
268+	* Session security is not yet supported.		000000		2007300	
269+	* This is the default mode.		000000		2007400	
270+D	DEFAULT_LM_COMPATIBILITY_MODE...		000000		2007500	
271+D	C 3		000000		2007600	
272+	* 3		000000		2007700	
273+D	LM_MODE_NTLM_V1...		000000		2007800	
274+D	C 0		000000		2007900	
275+D	LM_MODE_NTLM_V1_NO_LM...		000000		2008000	
276+D	C 1		000000		2008100	
277+D	LM_MODE_NTLM_V1_NTLM2_ONLY...		000000		2008200	
278+D	C 2		000000		2008300	
279+D	LM_MODE_NTLM_V2...		000000		2008400	
280+D	C 3		000000		2008500	
281+	* Message types		000000		2008600	
282+	* NEGOTIATE_MESSAGE...		000000		2008700	
283+D	C x'00000001'		000000		2008800	
284+D	C x'00000001'		000000		2008900	
285+D	CHALLENGE_MESSAGE...		000000		2009000	
286+D	C x'00000002'		000000		2009100	
287+D	AUTHENTICATE_MESSAGE...		000000		2009200	
288+D	C x'00000003'		000000		2009300	
289+	* Negotiate flags		000000		2009400	
290+	* NTLMSSP_NEGOTIATE_UNICODE...		000000		2009500	
291+D	C 1 x'00000001'		000000		2009600	
292+D	C 1 x'00000001'		000000		2009700	
293+D	NTLMSSP_NEGOTIATE_OEM...		000000		2009800	
294+D	C 2 x'00000002'		000000		2009900	
295+D	NTLMSSP_REQUEST_TARGET...		000000		2010000	
296+D	C 4 x'00000004'		000000		2010100	
297+D	NTLMSSP_UNUSED_R9...		000000		2010200	
298+D	C 8 x'00000008'		000000		2010300	
299+D	NTLMSSP_NEGOTIATE_SIGN...		000000		2010400	
300+D	C 16 x'00000010'		000000		2010500	
301+D	NTLMSSP_NEGOTIATE_SEAL...		000000		2010600	
302+D	C 32 x'00000020'		000000		2010700	
303+D	NTLMSSP_NEGOTIATE_DATAGRAM...		000000		2010800	
304+D	C 64 x'00000040'		000000		2010900	
305+D	NTLMSSP_NEGOTIATE_LM_KEY...		000000		2011000	
306+D	C 128 x'00000080'		000000		2011100	
307+D	NTLMSSP_UNUSED_R8...		000000		2011200	
308+D	C 256 x'00000100'		000000		2011300	
309+D	NTLMSSP_NEGOTIATE_NTLM...		000000		2011400	
310+D	C 512 x'00000200'		000000		2011500	
311+D	NTLMSSP_NEGOTIATE_NT_ONLY...		000000		2011600	
312+D	C 1024 x'00000400'		000000		2011700	
313+D	NTLMSSP_NEGOTIATE_CONNECTION_ANONYMOUS...		000000		2011800	
314+D	C 2048 x'00000800'		000000		2011900	
315+D	NTLMSSP_NEGOTIATE_OEM_DOMAIN_SUPPLIED...		000000		2012000	
316+D	C 4096 x'00001000'		000000		2012100	
317+D	NTLMSSP_NEGOTIATE_OEM_WORKSTATION_SUPPLIED...		000000		2012200	
318+D	C 8192 x'00002000'		000000		2012300	

Riga	Specifiche origine	Commenti	Es	Pag.	Modif.	OrigSeq.
Numero1.....2.....3.....4.....5.....6.....7.....8.....9.....10	Num Riga	Data	Id	Numero	
319+D	NTLMSSP_UNUSED_R6...				000000	2012400
320+D	C	16384	x'00004000'		000000	2012500
321+D	NTLMSSP_NEGOTIATE_ALWAYS_SIGN...				000000	2012600
322+D	C	32768	x'00008000'		000000	2012700
323+D	NTLMSSP_TARGET_TYPE_DOMAIN...				000000	2012800
324+D	C	65536	x'00010000'		000000	2012900
325+D	NTLMSSP_TARGET_TYPE_SERVER...				000000	2013000
326+D	C	131072	x'00020000'		000000	2013100
327+D	NTLMSSP_TARGET_TYPE_SHARE...				000000	2013200
328+D	C	262144	x'00040000'		000000	2013300
329+D	NTLMSSP_NEGOTIATE_NTLM2...				000000	2013400
330+D	C	524288	x'00080000'		000000	2013500
331+D	NTLMSSP_NEGOTIATE_IDENTIFY...				000000	2013600
332+D	C	1048576	x'00100000'		000000	2013700
333+D	NTLMSSP_UNUSED_R5...				000000	2013800
334+D	C	2097152	x'00200000'		000000	2013900
335+D	NTLMSSP_REQUEST_NON_NT_SESSION_KEY...				000000	2014000
336+D	C	4194304	x'00400000'		000000	2014100
337+D	NTLMSSP_NEGOTIATE_TARGET_INFO...				000000	2014200
338+D	C	8388608	x'00800000'		000000	2014300
339+D	NTLMSSP_UNUSED_R4...				000000	2014400
340+D	C	33554432	x'01000000'		000000	2014500
341+ *	The VERSION structure contains Windows version information that SHOULD					
342+ *	be ignored. This structure is used for debugging purposes only and its					
343+ *	value does not affect NTLM message.					
344+D	NTLMSSP_NEGOTIATE_VERSION...				000000	2014800
345+D	C	33554432	x'02000000'		000000	2014900
346+D	NTLMSSP_UNUSED_R3...				000000	2015000
347+D	C	67108864	x'04000000'		000000	2015100
348+D	NTLMSSP_UNUSED_R2...				000000	2015200
349+D	C	134217728	x'08000000'		000000	2015300
350+ *					000000	2015400
351+D	NTLMSSP_UNUSED_R1...				000000	2015500
352+D	C	536870912	x'10000000'		000000	2015600
353+D	NTLMSSP_NEGOTIATE_128...				000000	2015700
354+D	C	536870912	x'20000000'		000000	2015800
355+D	NTLMSSP_NEGOTIATE_KEY_EXCH...				000000	2015900
356+D	C	1073741824	x'40000000'		000000	2016000
357+D	NTLMSSP_NEGOTIATE_56...				000000	2016100
358+D	C	2147483648	x'80000000'		000000	2016200
359+ *					000000	2016300
360+ *	Message structures					
361+D	NtLmMessage_t...				000000	2016400
362+D	DS	qualified		based (pDummy)	000000	2016500
363+D	signature	8A			000000	2016600
364+D	type	10U 0			000000	2016700
365+ *					000000	2016800
366+ *	Message structures					
367+D	NtLmNegotiate_t...				000000	2016900
368+D	DS	qualified		based (pDummy)	000000	2017000
369+D	signature	8A			000000	2017100
370+D	type	10U 0			000000	2017200
371+D	flags	10U 0			000000	2017300
372+D	domain		likeds(ntlm_securityBuffer_t)		000000	2017400
373+D	workstation		likeds(ntlm_securityBuffer_t)		000000	2017500
374+D	os_version		likeds(os_version_t)		000000	2017600

Riga	Specifiche origine	Commenti	Es	Pag.	Modif.	OrigSeq.
Numero			Num	Riga	Data	Id Numero
375+	* domain	char(*)			000000	2018000
376+	* workstation	char(*)			000000	2018100
377+	*				000000	2018200
378+D	ntlm_securityBuffer_t...				000000	2018300
379+D	DS	qualified		based(pDummy)	000000	2018400
380+D	length	5U 0			000000	2018500
381+D	maxLen	5U 0			000000	2018600
382+D	offset	10U 0			000000	2018700
383+	*				000000	2018800
384+D	NtLmChallenge_t...				000000	2018900
385+D	DS	qualified		based(pDummy)	000000	2019000
386+D	signature	8A			000000	2019100
387+D	type	10U 0			000000	2019200
388+D	targetName	likeds(ntlm_securityBuffer_t)			000000	2019300
389+D	flags	10U 0			000000	2019400
390+D	challenge	like(ntlm_challenge_t)			000000	2019500
391+D	reserved	8A			000000	2019600
392+D	targetInfo	likeds(ntlm_securityBuffer_t)			000000	2019700
393+D	os_version	likeds(os_version_t)			000000	2019800
394+	* targetInfoFields	byte(8)			000000	2019900
395+	* version	byte(8)			000000	2020000
396+	* payload	char(*)			000000	2020100
397+	*				000000	2020200
398+D	targetNameChars_t...				000000	2020300
399+D	DS	qualified		based(pDummy)	000000	2020400
400+D	unicode	1024C			000000	2020500
401+D	value	2048A		overlay(unicode)	000000	2020600
402+	*				000000	2020700
403+D	targetInfo_t...				000000	2020800
404+D	DS	qualified		based(pDummy)	000000	2020900
405+D	type	5U 0			000000	2021000
406+D	length	5U 0			000000	2021100
407+D	unicode	1024C			000000	2021200
408+D	value	2048A		overlay(unicode)	000000	2021300
409+	*				000000	2021400
410+D	os_version_t...				000000	2021500
411+D	DS	qualified		based(pDummy)	000000	2021600
412+D	major	3U 0			000000	2021700
413+D	minor	3U 0			000000	2021800
414+D	build	5U 0			000000	2021900
415+D	reserved_1	3A			000000	2022000
416+D	NtLM_rev	3U 0			000000	2022100
417+	*				000000	2022200
418+	* Target information types				000000	2022300
419+D	NtLM_TARGET_TYPE_NONE...				000000	2022400
420+D	C	0			000000	2022500
421+D	NtLM_TARGET_TYPE_NB_COMPUTER_NAME...				000000	2022600
422+D	C	1			000000	2022700
423+D	NtLM_TARGET_TYPE_NB_DOMAIN_NAME...				000000	2022800
424+D	C	2			000000	2022900
425+D	NtLM_TARGET_TYPE_DNS_COMPUTER_NAME...				000000	2023000
426+D	C	3			000000	2023100
427+D	NtLM_TARGET_TYPE_DNS_DOMAIN_NAME...				000000	2023200
428+D	C	4			000000	2023300
429+D	NtLM_TARGET_TYPE_DNS_TREE_NAME...				000000	2023400
430+D	C	5			000000	2023500

Riga	Specifiche origine	Commenti	Es	Pag.	Modif.	OrigSeq.
Numero			Num	Riga	Data	Id Numero
431+D	NTLM_TARGET_TYPE_FLAGS...				000000	2023600
432+D	C	6			000000	2023700
433+D	NTLM_TARGET_TYPE_TIMESTAMP...				000000	2023800
434+D	C	7			000000	2023900
435+D	NTLM_TARGET_TYPE_RESTRICTIONS...				000000	2024000
436+D	C	8			000000	2024100
437+ *					000000	2024200
438+D	NtLmAuthenticate_t...				000000	2024300
439+D	DS	qualified		based (pDummy)	000000	2024400
440+D	signature	8A			000000	2024500
441+D	type	10U 0			000000	2024600
442+D	LM_resp	likeds(ntlm_securityBuffer_t)			000000	2024700
443+D	NTLM_resp	likeds(ntlm_securityBuffer_t)			000000	2024800
444+D	targetName	likeds(ntlm_securityBuffer_t)			000000	2024900
445+D	userName	likeds(ntlm_securityBuffer_t)			000000	2025000
446+D	workstation	likeds(ntlm_securityBuffer_t)			000000	2025100
447+D	sessionKey	likeds(ntlm_securityBuffer_t)			000000	2025200
448+D	flags	10U 0			000000	2025300
449+ *	os_version	like(os_version_t)			R 000000	2025400
450+ *	Target Name Data ("DOMAIN")				000000	2025500
451+ *	User Name Data ("user")				000000	2025600
452+ *	Workstation Name Data ("WORKSTATION")				000000	2025700
453+ *	LM Response Data				000000	2025800
454+ *	NTLM Response Data				000000	2025900
455+ *					000000	2026000
456+ *	Negotiation status				000000	2026100
457+D	NTLM_NONE	C	0		000000	2026200
458+D	NTLM_NEGOTIATE...				000000	2026300
459+D	C	1			000000	2026400
460+D	NTLM_AUTHENTICATE...				000000	2026500
461+D	C	2			000000	2026600
462+ *					000000	2026700
463+ *	Error codes				000000	2026800
464+D	NTLM_EINV_TYPE1_MSG...				000000	2026900
465+D	C	1			000000	2027000
466+D	NTLM_EINV_TYPE2_MSG...				000000	2027100
467+D	C	2			000000	2027200
468+D	NTLM_EINV_TYPE3_MSG...				000000	2027300
469+D	C	3			000000	2027400
470+D	NTLM_EINV_ENCODING...				000000	2027500
471+D	C	4			000000	2027600
472+D	NTLM_ENSUP_ENCODING...				000000	2027700
473+D	C	5			000000	2027800
474+ *					000000	2027900
475+ *	- - - - -				000000	2028000
476+ *	Protected Prototypes of NTLM				000000	2028100
477+ *	- - - - -				000000	2028200
478+ *					000000	2028300
479+ *	Transcoder handle.				000000	2028400
480+D	hTranscoder_t	S	*	based (pDummy)	000000	2028500
481+ *					000000	2028600
482+ *	Enables Test Mode for RPGUnit Test Cases				000000	2028700
483+D	NTLM_enableTestMode...				000000	2028800
484+D	PR				000000	2028900
485+D		extproc('NTLMR4_+			000000	2029000
486+D		NTLM_enableTestMode+			000000	2029100

Riga	<----- Specifiche origine ----->	<----- Commenti ----->	Es	Pag.	Modif.	OrigSeq.
Numero1.....2.....3.....4.....5.....6.....7.....8.....9.....10	Num Riga	Data	Id	Numero	
487+D)	000000		2029200	
488+D	i_mode	N const	000000		2029300	
489+ *			000000		2029400	
490+ *	Sets the LM compatibility mode.		000000		2029500	
491+ *			000000		2029600	
492+ *	0,1 -- Sends LM and NTLM responses.		000000		2029700	
493+ *	2 -- Sends only the NTLM response.		000000		2029800	
494+ *	This is more secure than Levels 0 and 1, because it		000000		2029900	
495+ *	eliminates the cryptographically-weak LM response.		000000		2030000	
496+ *	3,4,5 -- Sends LMv2 and NTLMv2 data.		000000		2030100	
497+ *	NTLMv2 session security is not yet supported.		000000		2030200	
498+D	NTLM_setLMCompatibility...		000000		2030300	
499+D	PR		000000		2030400	
500+D		extproc('NTLMR4_+	000000		2030500	
501+D		NTLM_setLMCompatibility+	000000		2030600	
502+D)	000000		2030700	
503+D	i_mode	10I 0 const options(*nopass)	000000		2030800	
504+ *			000000		2030900	
505+ *	Produces a Type-1 message		000000		2031000	
506+ *			000000		2031100	
507+ *	i_flags	Message flags. 0=use default flags.	000000		2031200	
508+ *	i_workstation	Workstation name of the client.	000000		2031300	
509+ *	i_domain	Name of the domain in which the workstation has	000000		2031400	
510+ *		membership.	000000		2031500	
511+D	Message_newType1...		000000		2031600	
512+D	PR	like(ntlm_message_t)	000000		2031700	
513+D		extproc('NTLMR4_+	000000		2031800	
514+D		Message_newType1+	000000		2031900	
515+D)	000000		2032000	
516+D	i_flags	10U 0 const options(*nopass: *omit)	000000		2032100	
517+D	i_workstation	const like(ntlm_workstation_t)	000000		2032200	
518+D		options(*varsize:	000000		2032300	
519+D		*nopass: *omit)	000000		2032400	
520+D	i_domain	const like(ntlm_domain_t)	000000		2032500	
521+D		options(*varsize:	000000		2032600	
522+D		*nopass: *omit)	000000		2032700	
523+ *			000000		2032800	
524+ *	Validates a Type-2 message.		000000		2032900	
525+ *			000000		2033000	
526+ *	i_type1Msg	Type-1 message that was sent to the server.	000000		2033100	
527+ *	i_type2Msg	Type-2 message responded by the server.	000000		2033200	
528+D	Message_validateType2...		000000		2033300	
529+D	PR	N	000000		2033400	
530+D		extproc('NTLMR4_+	000000		2033500	
531+D		Message_validateType2+	000000		2033600	
532+D)	000000		2033700	
533+D	i_type1Msg	like(ntlm_message_t)	000000		2033800	
534+D		options(*varsize)	000000		2033900	
535+D	i_type2Msg	like(ntlm_message_t)	000000		2034000	
536+D		options(*varsize)	000000		2034100	
537+D	o_rc	10I 0 options(*nopass)	000000		2034200	
538+ *			000000		2034300	
539+ *	Produces a Type-3 message:	NtLmResponse	000000		2034400	
540+ *			000000		2034500	
541+ *	i_type2Msg	Type-2 message responded by the server.	000000		2034600	
542+ *	i_user	The username for the authenticating user.	000000		2034700	

Riga	<----- Specifiche origine ----->	<----- Commenti ----->	Es	Pag.	Modif.	OrigSeq.
Numero1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8.....+.....9.....+.....10	Num Riga	Data	Id	Numero	
543+	* i_password	The password to use when constructing the response.	000000		2034800	
544+	* i_domain	The domain in which the user has an account.	000000		2034900	
545+D	Message_newType3...		000000		2035000	
546+D	PR	like(ntlm_message_t)	000000		2035100	
547+D		extproc('NTLMR4_+	000000		2035200	
548+D		Message_newType3+	000000		2035300	
549+D		')	000000		2035400	
550+D	i_type2Msg	like(ntlm_message_t)	000000		2035500	
551+D		options(*varsize)	000000		2035600	
552+D	i_user	const like(ntlm_user_t)	000000		2035700	
553+D		options(*varsize)	000000		2035800	
554+D	i_password	const like(ntlm_password_t)	000000		2035900	
555+D		options(*varsize)	000000		2036000	
556+D	i_domain	const like(ntlm_domain_t)	000000		2036100	
557+D		options(*varsize	000000		2036200	
558+D		: *omit: *nopass)	000000		2036300	
559+	*		000000		2036400	
560+	* Returns cTrue if the specified message is a Type 1 message.		000000		2036500	
561+	*		000000		2036600	
562+	* i_message	Message, that is tested for a Type-1 message.	000000		2036700	
563+D	Message_isType1...		000000		2036800	
564+D	PR	N	000000		2036900	
565+D		extproc('NTLMR4_+	000000		2037000	
566+D		Message_isType1+	000000		2037100	
567+D		')	000000		2037200	
568+D	i_message	const like(ntlm_message_t)	000000		2037300	
569+D		options(*varsize)	000000		2037400	
570+	*		000000		2037500	
571+	* Returns cTrue if the specified message is a Type 2 message.		000000		2037600	
572+	*		000000		2037700	
573+	* i_message	Message, that is tested for a Type-2 message.	000000		2037800	
574+D	Message_isType2...		000000		2037900	
575+D	PR	N	000000		2038000	
576+D		extproc('NTLMR4_+	000000		2038100	
577+D		Message_isType2+	000000		2038200	
578+D		')	000000		2038300	
579+D	i_message	const like(ntlm_message_t)	000000		2038400	
580+D		options(*varsize)	000000		2038500	
581+	*		000000		2038600	
582+	* Returns cTrue if the specified message is a Type 3 message.		000000		2038700	
583+	*		000000		2038800	
584+	* i_message	Message, that is tested for a Type-3 message.	000000		2038900	
585+D	Message_isType3...		000000		2039000	
586+D	PR	N	000000		2039100	
587+D		extproc('NTLMR4_+	000000		2039200	
588+D		Message_isType3+	000000		2039300	
589+D		')	000000		2039400	
590+D	i_message	const like(ntlm_message_t)	000000		2039500	
591+D		options(*varsize)	000000		2039600	
592+	*		000000		2039700	
593+	* Returns the challenge for a given Type-2 message.		000000		2039800	
594+	*		000000		2039900	
595+	* i_type2Msg	Message, whose challenge is returned.	000000		2040000	
596+D	Message_getChallenge...		000000		2040100	
597+D	PR	like(ntlm_challenge_t)	000000		2040200	
598+D		extproc('NTLMR4_+	000000		2040300	

Riga	<----- Specifiche origine ----->	<----- Commenti ----->	Es	Pag.	Modif.	OrigSeq.
Numero1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8.....+.....9.....+.....10	Num Riga	Data	Id	Numero	
599+D	Message_getChallenge+		000000		2040400	
600+D	')		000000		2040500	
601+D	i_message	like(ntlm_message_t)	000000		2040600	
602+D		options(*varsize)	000000		2040700	
603+ *			000000		2040800	
604+ *	Returns cTrue if the specified flag is set in a given message.		000000		2040900	
605+ *			000000		2041000	
606+ *	i_message	Message, that is tested for a given flag.	000000		2041100	
607+ *	i_flag	Flag, the message is tested for.	000000		2041200	
608+D	Message_hasFlag...		000000		2041300	
609+D	PR	N	000000		2041400	
610+D		extproc('NTLMR4_+	000000		2041500	
611+D	Message_hasFlag+		000000		2041600	
612+D		')	000000		2041700	
613+D	i_message	const like(ntlm_message_t)	000000		2041800	
614+D		options(*varsize)	000000		2041900	
615+D	i_flag	10U 0 const	000000		2042000	
616+ *			000000		2042100	
617+ *	Returns the server's NetBIOS computer name.		000000		2042200	
618+ *			000000		2042300	
619+ *	i_type2Msg	Message, the NetBIOS computer name is	000000		2042400	
620+ *		retrieved from.	000000		2042500	
621+D	Message_getTargetNBComputerName...		000000		2042600	
622+D	PR	like(ntlm_targetName_t)	000000		2042700	
623+D		extproc('NTLMR4_+	000000		2042800	
624+D		Message_getTargetNBComputerName+	000000		2042900	
625+D		')	000000		2043000	
626+D	i_type2Msg	like(ntlm_message_t)	000000		2043100	
627+D		options(*varsize)	000000		2043200	
628+ *			000000		2043300	
629+ *	Returns the server's NetBIOS domain name.		000000		2043400	
630+ *			000000		2043500	
631+ *	i_type2Msg	Message, the NetBIOS domain name is	000000		2043600	
632+ *		retrieved from.	000000		2043700	
633+D	Message_getTargetNBDomainName...		000000		2043800	
634+D	PR	like(ntlm_targetName_t)	000000		2043900	
635+D		extproc('NTLMR4_+	000000		2044000	
636+D		Message_getTargetNBDomainName+	000000		2044100	
637+D		')	000000		2044200	
638+D	i_type2Msg	like(ntlm_message_t)	000000		2044300	
639+D		options(*varsize)	000000		2044400	
640+ *			000000		2044500	
641+ *	Returns the server's Active Directory DNS computer name.		000000		2044600	
642+ *			000000		2044700	
643+ *	i_type2Msg	Message, the Active Directory DNS computer name	000000		2044800	
644+ *		retrieved from.	000000		2044900	
645+D	Message_getTargetDNSComputerName...		000000		2045000	
646+D	PR	like(ntlm_targetName_t)	000000		2045100	
647+D		extproc('NTLMR4_+	000000		2045200	
648+D		Message_getTargetDNSComputerName+	000000		2045300	
649+D		')	000000		2045400	
650+D	i_type2Msg	like(ntlm_message_t)	000000		2045500	
651+D		options(*varsize)	000000		2045600	
652+ *			000000		2045700	
653+ *	Returns the server's Active Directory DNS domain name.		000000		2045800	
654+ *			000000		2045900	

Riga	<----- Specifiche origine ----->	<----- Commenti ----->	Es	Pag.	Modif.	OrigSeq.
Numero1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8.....+.....9.....+.....10	Num Riga	Data	Id	Numero	
655+	* i_type2Msg	Message, the Active Directory DNS domain name			000000	2046000
656+	*	retrieved from.			000000	2046100
657+D	Message_getTargetDNSDomainName...				000000	2046200
658+D	PR	like(ntlm_targetName_t)			000000	2046300
659+D		extproc('NTLMR4_+			000000	2046400
660+D		Message_getTargetDNSDomainName+			000000	2046500
661+D		')			000000	2046600
662+D	i_type2Msg	like(ntlm_message_t)			000000	2046700
663+D		options(*varsize)			000000	2046800
664+	*				000000	2046900
665+	*	Returns server's Active Directory (AD) DNS forest tree name.			000000	2047000
666+	*				000000	2047100
667+	* i_type2Msg	Message, the Active Directory DNS forest tree name			000000	2047200
668+	*	retrieved from.			000000	2047300
669+D	Message_getTargetDNSTreeName...				000000	2047400
670+D	PR	like(ntlm_targetName_t)			000000	2047500
671+D		extproc('NTLMR4_+			000000	2047600
672+D		Message_getTargetDNSTreeName+			000000	2047700
673+D		')			000000	2047800
674+D	i_type2Msg	like(ntlm_message_t)			000000	2047900
675+D		options(*varsize)			000000	2048000
676+	*				000000	2048100
677+	*	Encodes a given message to Base64.			000000	2048200
678+	*				000000	2048300
679+	* i_message	Message material that is encoded to Base64.			000000	2048400
680+D	Message_encodeBase64...				000000	2048500
681+D	PR	like(ntlm_message_t)			000000	2048600
682+D		extproc('NTLMR4_+			000000	2048700
683+D		Message_encodeBase64+			000000	2048800
684+D		')			000000	2048900
685+D	i_message	const like(ntlm_message_t)			000000	2049000
686+D		options(*varsize)			000000	2049100
687+	*				000000	2049200
688+	*	Decodes a given message from Base64.			000000	2049300
689+	*				000000	2049400
690+	* i_message	Base64 encoded message that is decoded to			000000	2049500
691+	*	its binary form.			000000	2049600
692+D	Message_decodeBase64...				000000	2049700
693+D	PR	like(ntlm_message_t)			000000	2049800
694+D		extproc('NTLMR4_+			000000	2049900
695+D		Message_decodeBase64+			000000	2050000
696+D		')			000000	2050100
697+D	i_message	const like(ntlm_message_t)			000000	2050200
698+D		options(*varsize)			000000	2050300
699+	*				000000	2050400
700+	*	Removes user and password from a given URL.			000000	2050500
701+D	removeAuthFromUrl...				000000	2050600
702+D	PR	32767A varying			000000	2050700
703+D		extproc('NTLMR4_+			000000	2050800
704+D		removeAuthFromUrl+			000000	2050900
705+D		')			000000	2051000
706+D	i_URL	32767A const varying options(*varsize)			000000	2051100
707+	*				000000	2051200
708+	*	Constructs the LanManager response to the given Type-2 message			000000	2051300
709+	*	using the supplied password and server challenge.			000000	2051400
710+D	getLMResponse...				000000	2051500

Riga	Specifiche origine	Commenti	Es	Pag.	Modif.	OrigSeq.
Numero			Num	Riga	Data	Id Numero
711+D	PR	like(ntlm_lmResponse_t)			000000	2051600
712+D		extproc('NTLMR4_+			000000	2051700
713+D		getLMResponse+			000000	2051800
714+D		')			000000	2051900
715+D	i_NtLmChallenge...				000000	2052000
716+D		const likeds(NtLmChallenge_t)			000000	2052100
717+D	i_password	const like(ntlm_password_t)			000000	2052200
718+ *					000000	2052300
719+ *	Constructs the NT response to the given Type-2 message				000000	2052400
720+ *	using the supplied password and server challenge.				000000	2052500
721+D	getNTResponse...				000000	2052600
722+D	PR	like(ntlm_ntResponse_t)			000000	2052700
723+D		extproc('NTLMR4_+			000000	2052800
724+D		getNTResponse+			000000	2052900
725+D		')			000000	2053000
726+D	i_NtLmChallenge...				000000	2053100
727+D		const likeds(NtLmChallenge_t)			000000	2053200
728+D	i_password	const like(ntlm_password_t)			000000	2053300
729+ *					000000	2053400
730+ *	Constructs the NTLM2 response to the given Type-2 message				000000	2053500
731+ *	using the supplied NT response key (NTOWFv1) and challenges.				000000	2053600
732+D	getNTLM2Response...				000000	2053700
733+D	PR	like(ntlm_ntResponse_t)			000000	2053800
734+D		extproc('NTLMR4_+			000000	2053900
735+D		getNTLM2Response+			000000	2054000
736+D		')			000000	2054100
737+D	i_password	const like(ntlm_password_t)			000000	2054200
738+D		options(*varsize)			000000	2054300
739+D	i_serverChallenge...				000000	2054400
740+D		const like(ntlm_challenge_t)			000000	2054500
741+D	i_clientChallenge...				000000	2054600
742+D		const like(ntlm_challenge_t)			000000	2054700
743+ *					000000	2054800
744+ *	Constructs the LMv2 response to the given Type-2 message				000000	2054900
745+ *	using the supplied information.				000000	2055000
746+D	getLMv2Response...				000000	2055100
747+D	PR	like(ntlm_lmResponse_t)			000000	2055200
748+D		extproc('NTLMR4_+			000000	2055300
749+D		getLMv2Response+			000000	2055400
750+D		')			000000	2055500
751+D	i_NtLmChallenge...				000000	2055600
752+D		likeds(NtLmChallenge_t)			000000	2055700
753+D	i_domain	const like(ntlm_domain_t)			000000	2055800
754+D	i_user	const like(ntlm_user_t)			000000	2055900
755+D	i_password	const like(ntlm_password_t)			000000	2056000
756+D	i_clientChallenge...				000000	2056100
757+D		const like(ntlm_challenge_t)			000000	2056200
758+ *					000000	2056300
759+ *	Constructs the NTLMv2 response to the given Type-2 message				000000	2056400
760+ *	using the supplied information.				000000	2056500
761+D	getNTLMv2Response...				000000	2056600
762+D	PR	like(ntlm_ntlmResponse_t)			000000	2056700
763+D		extproc('NTLMR4_+			000000	2056800
764+D		getNTLMv2Response+			000000	2056900
765+D		')			000000	2057000
766+D	i_NtLmChallenge...				000000	2057100

Riga	Specifiche origine	Commenti	Es	Pag.	Modif.	OrigSeq.
Numero	1	2	3	4	5	6
Numero	1	2	3	4	5	6
767+D						000000
768+D	i_respKeyNT		const	like(MD5_digest_t)		000000
769+D	i_clientChallenge...		const	like(ntlm_challenge_t)		000000
770+D			const	like(ntlm_challenge_t)		000000
771+ *						000000
772+ *	Creates the NTLMv2 response for the supplied information.					000000
773+D	computeResponse...					000000
774+D	PR			like(ntlm_ntlmResponse_t)		000000
775+D				extproc('NTLMR4_+		000000
776+D				computeResponse+		000000
777+D				')		000000
778+D	i_respKeyNT		const	like(MD5_digest_t)		000000
779+D	i_serverChallenge...					000000
780+D			const	like(ntlm_challenge_t)		000000
781+D	i_clientData	2048A	const			000000
782+D	i_length	10I 0	const			000000
783+ *						000000
784+ *	Calculate the key used in NTLM v1 authentication.					000000
785+D	NTOWFv1...					000000
786+D	PR			like(MD4_digest_t)		000000
787+D				extproc('NTLMR4_+		000000
788+D				NTOWFv1+		000000
789+D				')		000000
790+D	i_password		const	like(ntlm_password_t)		000000
791+D				options(*varsize)		000000
792+ *						000000
793+ *	Calculate the key used in NTLM v2 authentication.					000000
794+D	NTOWFv2...					000000
795+D	PR			like(MD5_digest_t)		000000
796+D				extproc('NTLMR4_+		000000
797+D				NTOWFv2+		000000
798+D				')		000000
799+D	i_password		const	like(ntlm_password_t)		000000
800+D				options(*varsize)		000000
801+D	i_user		const	like(ntlm_user_t)		000000
802+D				options(*varsize)		000000
803+D	i_domain		const	like(ntlm_domain_t)		000000
804+D				options(*varsize)		000000
805+ *						000000
806+ *	Converts a given EBCDIC string to ASCII.					000000
807+D	toAscii...					000000
808+D	PR	2048A		varying		000000
809+D				extproc('NTLMR4_+		000000
810+D				toAscii+		000000
811+D				')		000000
812+D	i_ebcdic	2048A	const	varying		000000
813+ *						000000
814+ *	Frees the resources allocated by the ASCII transcoder.					000000
815+D	freeAsciiTranscoder...					000000
816+D	PR					000000
817+ *						000000
818+ *	Converts a given EBCDIC string to UNICODE.					000000
819+D	toUniCode...					000000
820+D	PR	4096A		varying		000000
821+D				extproc('NTLMR4_+		000000
822+D				toUnicode+		000000

Riga	-----	Specifiche origine	-----><-----	Commenti	---->	Es	Pag.	Modif.	OrigSeq.	
Numero1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9....+...10					Num	Riga	Data	Id	Numero
823+D)				000000		2062800
824+D	i_ebcdic	2048A	const	varying				000000		2062900
825+ *								000000		2063000
826+ *	Frees the resources allocated by the UNICODE transcoder.							000000		2063100
827+D	freeUnicodeTranscoder...							000000		2063200
828+D	PR							000000		2063300
829+ *								000000		2063400
830+ *	Returns cTrue if the specified bit is set.							000000		2063500
831+D	isBit...							000000		2063600
832+D	PR	N						000000		2063700
833+D				extproc('NTLMR4_+				000000		2063800
834+D				isBit+				000000		2063900
835+D)				000000		2064000
836+D	i_value	20U 0	const					000000		2064100
837+D	i_bit	20U 0	const					000000		2064200
838+ *								000000		2064300
839+ *	Returns the bit-wise ANDing of the bits of all the arguments.							000000		2064400
840+D	bitand...							000000		2064500
841+D	PR	20U 0						000000		2064600
842+D				extproc('NTLMR4_+				000000		2064700
843+D				bitand+				000000		2064800
844+D)				000000		2064900
845+D	i_source1	20U 0	value					000000		2065000
846+D	i_source2	20U 0	value					000000		2065100
847+ *								000000		2065200
848+ *	Returns the bit-wise ANDing of the bits of all the arguments.							000000		2065300
849+D	byteand...							000000		2065400
850+D	PR	1A						000000		2065500
851+D				extproc('NTLMR4_+				000000		2065600
852+D				byteand+				000000		2065700
853+D)				000000		2065800
854+D	i_source1	1A	value					000000		2065900
855+D	i_source2	1A	value					000000		2066000
856+ *								000000		2066100
857+ *	Returns the bit-wise ORing of the bits of all the arguments.							000000		2066200
858+D	bitor...							000000		2066300
859+D	PR	20U 0						000000		2066400
860+D				extproc('NTLMR4_+				000000		2066500
861+D				bitor+				000000		2066600
862+D)				000000		2066700
863+D	i_source1	20U 0	value					000000		2066800
864+D	i_source2	20U 0	value					000000		2066900
865+ *								000000		2067000
866+ *	Returns the bit-wise ORing of the bits of all the arguments.							000000		2067100
867+D	byteor...							000000		2067200
868+D	PR	1A						000000		2067300
869+D				extproc('NTLMR4_+				000000		2067400
870+D				byteor+				000000		2067500
871+D)				000000		2067600
872+D	i_source1	1A	value					000000		2067700
873+D	i_source2	1A	value					000000		2067800
874+ *								000000		2067900
875+ *	Converts a 2-byte integer to little-endian format.							000000		2068000
876+D	uint16LE...							000000		2068100
877+D	PR	5U 0						000000		2068200
878+D				extproc('NTLMR4_+				000000		2068300

Riga	<----- Specifiche origine ----->	<----- Commenti ----->	Es	Pag.	Modif.	OrigSeq.
Numero1.....2.....3.....4.....5.....6.....7.....8.....9.....10	Num Riga	Data	Id	Numero	
879+D		uint16LE+	000000		2068400	
880+D		')	000000		2068500	
881+D	i_int2	5U 0 const	000000		2068600	
882+ *			000000		2068700	
883+ *	Converts a 4-byte integer to little-endian format.		000000		2068800	
884+D		uint32LE...	000000		2068900	
885+D		PR 10U 0	000000		2069000	
886+D		extproc('NTLMR4_+	000000		2069100	
887+D		uint32LE+	000000		2069200	
888+D		')	000000		2069300	
889+D	i_int4	10U 0 const	000000		2069400	
890+ *			000000		2069500	
891+ *	Converts a 8-byte integer to little-endian format.		000000		2069600	
892+D		uint64LE...	000000		2069700	
893+D		PR 20U 0	000000		2069800	
894+D		extproc('NTLMR4_+	000000		2069900	
895+D		uint64LE+	000000		2070000	
896+D		')	000000		2070100	
897+D	i_int8	20U 0 const	000000		2070200	
898+ *			000000		2070300	
899+ *	Converts a string value to little-endian format.		000000		2070400	
900+D		stringLE...	000000		2070500	
901+D		PR 1024A varying	000000		2070600	
902+D		extproc('NTLMR4_+	000000		2070700	
903+D		stringLE+	000000		2070800	
904+D		')	000000		2070900	
905+D	i_string	1024A options(*varsize)	000000		2071000	
906+D	i_length	10I 0 const	000000		2071100	
907+ *			000000		2071200	
908+ *	Converts a string value to little-endian format.		000000		2071300	
909+D		varstringLE...	000000		2071400	
910+D		PR 1024A varying	000000		2071500	
911+D		extproc('NTLMR4_+	000000		2071600	
912+D		varstringLE+	000000		2071700	
913+D		')	000000		2071800	
914+D	i_string	1024A const varying options(*varsize)	000000		2071900	
915+ *			000000		2072000	
916+ *	Returns a random 8-byte challenge.		000000		2072100	
917+D		random...	000000		2072200	
918+D		PR opdesc	000000		2072300	
919+D		extproc('NTLMR4_+	000000		2072400	
920+D		random+	000000		2072500	
921+D		')	000000		2072600	
922+D	o_random	128A options(*varsize)	000000		2072700	
923+D	i_isTestMode	N const options(*nopass)	000000		2072800	
924+ *			000000		2072900	
925+ *	Converts a given string to upper case.		000000		2073000	
926+D		toUpper...	000000		2073100	
927+D		PR 1024A varying	000000		2073200	
928+D		extproc('NTLMR4_+	000000		2073300	
929+D		toUpper+	000000		2073400	
930+D		')	000000		2073500	
931+D	i_string	1024A const varying	000000		2073600	
932+ *			000000		2073700	
933+ *	Converts a given string to lower case.		000000		2073800	
934+D		toLower...	000000		2073900	

Riga	Specifiche origine	Commenti	Es	Pag.	Modif.	OrigSeq.
Numero	1	2	3	4	5	6
	10	Num	Riga	Data	Id	Numero
935+D	PR	1024A	varying			000000 2074000
936+D			extproc('NTLMR4_+			000000 2074100
937+D			toLower+			000000 2074200
938+D			')			000000 2074300
939+D	i_string	1024A	const varying			000000 2074400
940+ *						000000 2074500
941+ *			Returns the next token from a list of tokens.			000000 2074600
942+D			getToken...			000000 2074700
943+D	PR	2048A	varying			000000 2074800
944+D			extproc('NTLMR4_+			000000 2074900
945+D			getToken+			000000 2075000
946+D			')			000000 2075100
947+D	i_tokens	2048A	const varying options(*omit)			000000 2075200
948+D	i_delimiters	16A	const varying			000000 2075300
949+ *						000000 2075400
950+ *						000000 2075500
951+ *			Protected Prototypes of RC4 encryption			000000 2075600
952+ *						000000 2075700
953+D	RC4_key_t	S	16A varying			000000 2075800
954+ *						000000 2075900
955+ *			Returns the RC4 encrypted value of a given string.			000000 2076000
956+D	RC4...					000000 2076100
957+D	PR	4096A	varying			000000 2076200
958+D			extproc('ENCRYPTR4_+			000000 2076300
959+D			RC4+			000000 2076400
960+D			')			000000 2076500
961+D	i_key		const like(RC4_key_t)			000000 2076600
962+D	i_string	4096A	varying options(*varsize)			000000 2076700
963+ /IF NOT DEFINED(RC4_INTERNAL_USE)						000000 2076800
			LINES EXCLUDED: 1			
964+ /ENDIF						000000 2077000
965+ *						000000 2077100
966+ *						000000 2077200
967+ *			Protected Prototypes of DES encryption			000000 2077300
968+ *						000000 2077400
969+D	DES_key_t	S	8A			000000 2077500
970+ *						000000 2077600
971+ *			Returns the DES digest of a given string.			000000 2077700
972+D	DES...					000000 2077800
973+D	PR	4096A	varying			000000 2077900
974+D			extproc('ENCRYPTR4_+			000000 2078000
975+D			DES+			000000 2078100
976+D			')			000000 2078200
977+D	i_string	4096A	const varying options(*varsize)			000000 2078300
978+D	i_challenge		const like(ntlm_challenge_t)			000000 2078400
979+ *						000000 2078500
980+ *			Produces a DES key for a given 7-byte value.			000000 2078600
981+D	DES_produceKey...					000000 2078700
982+D	PR		like(DES_key_t)			000000 2078800
983+D			extproc('ENCRYPTR4_+			000000 2078900
984+D			DES_produceKey+			000000 2079000
985+D			')			000000 2079100
986+D	i_value	7A	const			000000 2079200
987+ *						000000 2079300
988+ *						000000 2079400
989+ *			Protected Prototypes of MD4 encryption			000000 2079500

Riga	Specifiche origine	Commenti	Es	Pag.	Modif.	OrigSeq.
Numero1.....2.....3.....4.....5.....6.....7.....8.....9.....10	Num Riga	Data	Id	Numero	
990+	*		000000		2079600	
991+	/COPY QRPGLSRC,MD4_H		000000		2079700	

	* Nome del membro RPG. : MD4_H				3	
	* Nome esterno : LIBHTTP/QRPGLSRC(MD4_H)				3	
	* Ultima modifica. : 02/01/15 16:10:21				3	
	* Testo 'descrizione'. : Prototypes for MD4 Digest				3	

992+/*-		+	150102		3000001	
993+*	Copyright (c) 2012-2015 Thomas Raddatz	+	150102		3000002	
994+*	All rights reserved.	+	150102		3000003	
995+*		+	150102		3000004	
996+*	Redistribution and use in source and binary forms, with or without	+	150102		3000005	
997+*	modification, are permitted provided that the following conditions	+	150102		3000006	
998+*	are met:	+	150102		3000007	
999+*	1. Redistributions of source code must retain the above copyright	+	150102		3000008	
1000+*	notice, this list of conditions and the following disclaimer.	+	150102		3000009	
1001+*	2. Redistributions in binary form must reproduce the above copyright	+	150102		3000010	
1002+*	notice, this list of conditions and the following disclaimer in the	+	150102		3000011	
1003+*	documentation and/or other materials provided with the distribution.	+	150102		3000012	
1004+*		+	150102		3000013	
1005+*	THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS 'AS IS' AND	+	150102		3000014	
1006+*	ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE	+	150102		3000015	
1007+*	IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE	+	150102		3000016	
1008+*	ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE	+	150102		3000017	
1009+*	FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL	+	150102		3000018	
1010+*	DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS	+	150102		3000019	
1011+*	OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)	+	150102		3000020	
1012+*	HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT	+	150102		3000021	
1013+*	LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY	+	150102		3000022	
1014+*	OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF	+	150102		3000023	
1015+*	SUCH DAMAGE.	+	150102		3000024	
1016+*		+	150102		3000025	
1017+*/		+	150102		3000026	
1018+*	=====*		000000		3000100	
1019+*	MD4 message-digest algorithm	*	000000		3000200	
1020+*	=====*		000000		3000300	
1021+*	Author : Thomas Raddatz	*	000000		3000400	
1022+*	Date : 22.05.2012	*	000000		3000500	
1023+*	E-mail : thomas.raddatz@tools400.de	*	000000		3000600	
1024+*	Homepage: www.tools400.de	*	000000		3000700	
1025+*	=====*		000000		3001000	
1026+*	/IF NOT DEFINED(MD4_H)		000000		3001100	
1027+*	/DEFINE MD4_H		000000		3001200	
1028+*			000000		3001300	
1029+D	MD4_digest_t S 16A based(pDummy)		000000		3001400	
1030+*			000000		3001500	
1031+D	MD4_CTX_t...		000000		3001600	
1032+D	DS qualified based(pDummy)		000000		3001700	
1033+D	state 1 16U 0 dim(4)		000000		3001800	
1034+D	stateA 1 16A		000000		3001900	
1035+D	count 17 24U 0 dim(2)		000000		3002000	
1036+D	countA 17 24A		000000		3002100	
1037+D	buffer 25 88A		000000		3002200	
1038+*			000000		3002300	
1039+*	MD4 initialization. Begins an MD4 operation, writing a new context.		000000		3002400	

Riga	-----	Specifiche origine	-----><-----	Commenti	---->	Es	Pag.	Modif.	OrigSeq.
Numero1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9....+...10	Num	Riga	Data	Id	Numero			
1040+D	MD4Init_r...					000000			3002500
1041+D		PR				000000			3002600
1042+D				extproc('MD4R4_+		000000			3002700
1043+D				MD4Init_r+		000000			3002800
1044+D				'		000000			3002900
1045+D	context			likeds(MD4_CTX_t)		000000			3003000
1046+ *						000000			3003100
1047+ *	MD4 block update operation. Continues an MD4 message-digest					000000			3003200
1048+ *	operation, processing another message block, and updating the					000000			3003300
1049+ *	context.					000000			3003400
1050+D	MD4Update_r...					000000			3003500
1051+D		PR				000000			3003600
1052+D				extproc('MD4R4_+		000000			3003700
1053+D				MD4Update_r+		000000			3003800
1054+D				'		000000			3003900
1055+D	context			likeds(MD4_CTX_t)		000000			3004000
1056+D	input			* value		000000			3004100
1057+D	inputLen			10U 0 value		000000			3004200
1058+ *						000000			3004300
1059+ *	MD4 finalization. Ends an MD4 message-digest operation, writing					000000			3004400
1060+ *	the message digest and zeroizing the context.					000000			3004500
1061+D	MD4Final_r...					000000			3004600
1062+D		PR				000000			3004700
1063+D				extproc('MD4R4_+		000000			3004800
1064+D				MD4Final_r+		000000			3004900
1065+D				'		000000			3005000
1066+D	digest			like(MD4_digest_t)		000000			3005100
1067+D	context			likeds(MD4_CTX_t)		000000			3005200
1068+ *						000000			3005300
1069+ *	MD4 operation.					000000			3005400
1070+D	MD4Only_r...					000000			3005500
1071+D		PR				000000			3005600
1072+D				extproc('MD4R4_+		000000			3005700
1073+D				MD4Only_r+		000000			3005800
1074+D				'		000000			3005900
1075+D	digest			like(MD4_digest_t)		000000			3006000
1076+D	input			* value		000000			3006100
1077+D	inputLen			10U 0 value		000000			3006200
1078+ *						000000			3006300
1079+ *	MD4 initialization. Begins an MD4 operation, writing a new context.					000000			3006400
1080+D	MD4Init_c...					000000			3006500
1081+D		PR				000000			3006600
1082+D				extproc('+		000000			3006700
1083+D				MD4Init_c+		000000			3006800
1084+D				'		000000			3006900
1085+D	context			likeds(MD4_CTX_t)		000000			3007000
1086+ *						000000			3007100
1087+ *	MD4 block update operation. Continues an MD4 message-digest					000000			3007200
1088+ *	operation, processing another message block, and updating the					000000			3007300
1089+ *	context.					000000			3007400
1090+D	MD4Update_c...					000000			3007500
1091+D		PR				000000			3007600
1092+D				extproc('+		000000			3007700
1093+D				MD4Update_c+		000000			3007800
1094+D				'		000000			3007900
1095+D	context			likeds(MD4_CTX_t)		000000			3008000

Riga	Specifiche origine	Commenti	Es	Pag.	Modif.	OrigSeq.
Numero	1...10	1...10	Num	Riga	Data	Id Numero
1096+D	input	* value			000000	3008100
1097+D	inputLen	10U 0 value			000000	3008200
1098+ *					000000	3008300
1099+ *	MD4 finalization. Ends an MD4 message-digest operation, writing				000000	3008400
1100+ *	the message digest and zeroizing the context.				000000	3008500
1101+D	MD4Final_c...				000000	3008600
1102+D	PR				000000	3008700
1103+D		extproc('+			000000	3008800
1104+D		MD4Final_c+			000000	3008900
1105+D		')			000000	3009000
1106+D	digest	like(MD4_digest_t)			000000	3009100
1107+D	context	likeds(MD4_CTX_t)			000000	3009200
1108+ *					000000	3009300
1109+ *	MD4 operation.				000000	3009400
1110+D	MD4Only_c...				000000	3009500
1111+D	PR				000000	3009600
1112+D		extproc('+			000000	3009700
1113+D		MD4Only_c+			000000	3009800
1114+D		')			000000	3009900
1115+D	digest	like(MD4_digest_t)			000000	3010000
1116+D	input	* value			000000	3010100
1117+D	inputLen	10U 0 value			000000	3010200
1118+ *					000000	3010300
1119+ /ENDIF					000000	3010400
1120+ *					000000	2079800
1121+ *	Returns the MD4 digest of a given string.				000000	2079900
1122+D	MD4...				000000	2080000
1123+D	PR	like(MD4_digest_t)			000000	2080100
1124+D		extproc('ENCRYPTR4_+			000000	2080200
1125+D		MD4+			000000	2080300
1126+D		')			000000	2080400
1127+D	i_string	4096A	varying options(*varsize)		000000	2080500
1128+ /IF NOT DEFINED(MD4_INTERNAL_USE)					000000	2080600
	LINES EXCLUDED: 1					
1129+ /ENDIF					000000	2080800
1130+ *					000000	2080900
1131+ *	-----				000000	2081000
1132+ *	Protected Prototypes of MD5 hmac and digest				000000	2081100
1133+ *	-----				000000	2081200
1134+D	MD5_digest_t	S	16A	based(pDummy)	000000	2081300
1135+ *					000000	2081400
1136+ *	Returns the MD5 digest of a given string.				000000	2081500
1137+D	MD5Hmac...				000000	2081600
1138+D	PR	like(MD5_digest_t)			000000	2081700
1139+D		extproc('ENCRYPTR4_+			000000	2081800
1140+D		MD5Hmac+			000000	2081900
1141+D		')			000000	2082000
1142+D	i_hmacKey	const	like(MD5_digest_t)		000000	2082100
1143+D	i_string	4096A	varying options(*varsize)		000000	2082200
1144+ /IF NOT DEFINED(MD5_INTERNAL_USE)					000000	2082300
	LINES EXCLUDED: 1					
1145+ /ENDIF					000000	2082500
1146+ *					000000	2082600
1147+ *	Returns the MD5 digest of a given string.				000000	2082700
1148+D	MD5Digest...				000000	2082800
1149+D	PR	like(MD5_digest_t)			000000	2082900

Riga	Specifiche origine	Commenti	Es	Pag.	Modif.	OrigSeq.
Numero			Num	Riga	Data	Id Numero
1150+D					000000	2083000
1151+D	extproc('ENCRYPTR4_+				000000	2083100
1152+D	MD5Digest+				000000	2083200
1153+D	i_string 4096A	varying options(*varsize)			000000	2083300
1154+	/IF NOT DEFINED(MD5_INTERNAL_USE) LINES EXCLUDED: 1				000000	2083400
1155+	/ENDIF				000000	2083600
1156+	*				000000	2083700
1157+	/ENDIF				000000	2083800
1158	/COPY QRPGLSRC,MD4_H				000000	006000
	-----*					
	* Nome del membro RPG.	MD4_H				4
	* Nome esterno	LIBHTTP/QRPGLSRC(MD4_H)				4
	* Ultima modifica.	02/01/15 16:10:21				4
	* Testo 'descrizione'.	Prototypes for MD4 Digest				4
	-----*					
1159+/*-					150102	4000001
1160+	* Copyright (c) 2012-2015 Thomas Raddatz				150102	4000002
1161+	* All rights reserved.				150102	4000003
1162+	*				150102	4000004
1163+	* Redistribution and use in source and binary forms, with or without				150102	4000005
1164+	* modification, are permitted provided that the following conditions				150102	4000006
1165+	* are met:				150102	4000007
1166+	* 1. Redistributions of source code must retain the above copyright				150102	4000008
1167+	* notice, this list of conditions and the following disclaimer.				150102	4000009
1168+	* 2. Redistributions in binary form must reproduce the above copyright				150102	4000010
1169+	* notice, this list of conditions and the following disclaimer in the				150102	4000011
1170+	* documentation and/or other materials provided with the distribution.				150102	4000012
1171+	*				150102	4000013
1172+	* THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS 'AS IS' AND				150102	4000014
1173+	* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE				150102	4000015
1174+	* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE				150102	4000016
1175+	* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE				150102	4000017
1176+	* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL				150102	4000018
1177+	* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS				150102	4000019
1178+	* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)				150102	4000020
1179+	* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT				150102	4000021
1180+	* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY				150102	4000022
1181+	* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF				150102	4000023
1182+	* SUCH DAMAGE.				150102	4000024
1183+	*				150102	4000025
1184+	*/				150102	4000026
1185+	=====*					
1186+	* MD4 message-digest algorithm				000000	4000200
1187+	=====*					
1188+	* Author : Thomas Raddatz				000000	4000300
1189+	* Date : 22.05.2012				000000	4000400
1190+	* E-mail : thomas.raddatz@tools400.de				000000	4000500
1191+	* Homepage: www.tools400.de				000000	4000600
1192+	=====*					
1193+	/IF NOT DEFINED(MD4_H) LINES EXCLUDED: 92				000000	4001100
1194+	/ENDIF				000000	4010400
1195	*				000000	006100
1196	-----				000000	006200
1197	* Imported prototypes				000000	006300

Riga	Specifiche origine	Commenti	Es	Pag.	Modif.	OrigSeq.
Numero1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8.....+.....9.....+.....10	Num Riga	Data	Id	Numero	
1198	*		000000		006400	
1199	*		000000		006500	
1200	*		000000		006600	
1201	*	Internal prototypes	000000		006700	
1202	*		000000		006800	
1203	*		000000		006900	
1204	*	Calculates the odd parity bit for each byte	000000		007000	
1205	*	of a given value.	000000		007100	
1206	D	setParityBit...	000000		007200	
1207	D	PR 1A	000000		007300	
1208	D	extproc('setParityBit')	000000		007400	
1209	D	i_char 1A const	000000		007500	
1210	*		000000		007600	
1211	*	QtqIconvOpen()--Code Conversion Allocation API	000000		007700	
1212	D	QtqIconv_open...	000000		007800	
1213	D	PR extproc('QtqIconvOpen')	000000		007900	
1214	D	likeds(iconv_t)	000000		008000	
1215	D	i_toCode const likeds(QtqCode_t)	000000		008100	
1216	D	i_fromCode const likeds(QtqCode_t)	000000		008200	
1217	*		000000		008300	
1218	D	iconv_t DS qualified based(pDummy) align	000000		008400	
1219	D	return_value 10I 0	000000		008500	
1220	D	cd 10I 0 dim(12)	000000		008600	
1221	*		000000		008700	
1222	D	QtqCode_t...	000000		008800	
1223	D	DS qualified based(pDummy)	000000		008900	
1224	D	ccsid 10I 0	000000		009000	
1225	D	conversionA 10I 0	000000		009100	
1226	D	substitutionA 10I 0	000000		009200	
1227	D	shiftStateA 10I 0	000000		009300	
1228	D	inplenOpt 10I 0	000000		009400	
1229	D	errOptMxdDta 10I 0	000000		009500	
1230	D	reserved 12A	000000		009600	
1231	*		000000		009700	
1232	*	iconv()--Code Conversion API	000000		009800	
1233	D	iconv...	000000		009900	
1234	D	PR 10U 0 extproc('iconv')	000000		010000	
1235	D	i_cd value likeds(iconv_t)	000000		010100	
1236	D	i_pInBuf *	000000		010200	
1237	D	i_inBytLeft 10U 0	000000		010300	
1238	D	i_pOutBuf *	000000		010400	
1239	D	i_outBytLeft 10U 0	000000		010500	
1240	*		000000		010600	
1241	D	ICONV_ERROR C const(4294967295)	000000		010700	
1242	D	E2BIG_C C const(3491)	000000		010800	
1243	*		000000		010900	Argument list
1244	*	iconv_close()--Code Conversion Deallocation API	000000		011000	
1245	D	iconv_close...	000000		011100	
1246	D	PR 10I 0 extproc('iconv_close')	000000		011200	
1247	D	i_cd value likeds(iconv_t)	000000		011300	
1248	*		000000		011400	
1249	*	Cipher (CIPHER)	000000		011500	
1250	D	cipher...	000000		011600	
1251	D	PR extproc('_CIPHER')	000000		011700	
1252	D	* const	000000		011800	
1253	D	* value	000000		011900	

Riga	Numero	Specifiche origine	Commenti	Es	Pag.	Modif.	OrigSeq.	
1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8.....+.....9.....+.....10			Num	Riga	Data	Id	Numero
1254	D	* const				000000		012000
1255	*					000000		012100
1256	D	cipherCtrls_0005_t...				000000		012200
1257	D	DS	qualified based(pDummy) align			000000		012300
1258	D	function	1 2A			000000		012400
1259	D	hashAlg	3 3A			000000		012500
1260	D	sequence	4 4A			000000		012600
1261	D	dataLength	5 8U 0			000000		012700
1262	D	output	9 9A			000000		012800
1263	D	reserved_1	10 16A			000000		012900
1264	D	hashContext	17 32*			000000		013000
1265	D	HMACKey	33 48*			000000		013100
1266	D	HMACKeyLength	49 52U 0			000000		013200
1267	D	reserved_2	53 96A			000000		013300
1268	*					000000		013400
1269	D	cipherCtrls_0013_t...				000000		013500
1270	D	DS	qualified based(pDummy) align			000000		013600
1271	D	function	1 2A			000000		013700
1272	D	dataLength	3 4U 0			000000		013800
1273	D	operation	5 5A			000000		013900
1274	D	reserved	6 16A			000000		014000
1275	D	keyCtxPtr	17 32*			000000		014100
1276	*					000000		014200
1277	D	cCIPHER_ENCRYPT...				000000		014300
1278	D	C	const(x'00')			000000		014400
1279	D	cCIPHER_DECRYPT...				000000		014500
1280	D	C	const(x'01')			000000		014600
1281	*					000000		014700
1282	D	cCIPHER_MD5...				000000		014800
1283	D	C	const(x'00')			000000		014900
1284	D	cCIPHER_HASH...				000000		015000
1285	D	C	const(x'00')			000000		015100
1286	D	cCIPHER_HMAC...				000000		015200
1287	D	C	const(x'01')			000000		015300
1288	D	cCIPHER_ONLY...				000000		015400
1289	D	C	const(x'00')			000000		015500
1290	*					000000		015600
1291	D	MD5_CTX_t	DS	qualified	based(pDummy)	000000		015700
1292	D	key	16A			000000		015800
1293	D	context	160A			000000		015900
1294	D	state	1A			000000		016000
1295	D	digest	16A			000000		016100
1296	*					000000		016200
1297	D	rc4_ctx_t	ds	qualified	based(pDummy)	000000		016300
1298	D	stream	256A			000000		016400
1299	D	length	5U 0			000000		016500
1300	D	reserved	6A			000000		016600
1301	*					000000		016700
1302	*	Encrypt Data (QC3ENCDT, Qc3EncryptData) API				000000		016800
1303	D	Qc3EncryptData...				000000		016900
1304	D	PR	extproc('Qc3EncryptData')			000000		017000
1305	D	i_clearData	65535A	const	options(*varsize)	000000		017100
1306	D	i_length	10I 0	const		000000		017200
1307	D	i_dataFormat	8A	const		000000		017300
1308	D	i_algDesc	65535A	const	options(*varsize)	000000		017400
1309	D	i_algFormat	8A	const		000000		017500

Riga	----- Specifiche origine -----				-----><----- Commenti ----->	Es	Pag.	Modif.	OrigSeq.	
Numero	1	2	3	4	5	6	7	Data	Id	Numero
1310	D	i_keyDesc	65535A	const	options(*varsize)			000000		017600
1311	D	i_keyFormat	8A	const				000000		017700
1312	D	i_cryptSrvPrv	1A	const				000000		017800
1313	D	i_cryptDevNme	10A	const				000000		017900
1314	D	o_encrypted	65535A		options(*varsize)			000000		018000
1315	D	i_encLenPrv	10I	0	const			000000		018100
1316	D	o_encLenRet	10I	0				000000		018200
1317	D	io_ErrCode	32767A		options(*nopass: *varsize)			000000		018300
1318	*							000000		018400
1319	D	algd0200_t	DS		qualified		based(pDummy)	000000		018500
1320	D	algorithm	10I	0				000000		018600
1321	D	blockLen	10I	0				000000		018700
1322	D	mode	1A					000000		018800
1323	D	padOption	1A					000000		018900
1324	D	padChar	1A					000000		019000
1325	D	reserved_1	3A					000000		019100
1326	D	macLen	10I	0				000000		019200
1327	D	keySize	10I	0				000000		019300
1328	D	initVector	32A					000000		019400
1329	*							000000		019500
1330	D	keyd0200_t	DS		qualified		based(pDummy)	000000		019600
1331	D	type	10I	0				000000		019700
1332	D	length	10I	0				000000		019800
1333	D	format	1A					000000		019900
1334	D	reserved_1	3A					000000		020000
1335	D	value	8A					000000		020100
1336	*							000000		020200
1337	*							000000		020300
1338	*	Global fields						000000		020400
1339	*							000000		020500
1340	*							000000		020600
1341	*							000000		020700
1342	*	*** Exported, because internally used by NTLMR4, RPGUNIT tests ***						R 000000		020800
1343	*	Encrypts a given string using the RC4 algorithm						000000		020900
1344	*							000000		021000
1345	P	RC4...						000000		021100
1346	P		B		export			000000		021200
1347	*							000000		021300
1348	D	RC4...						000000		021400
1349	D		PI	4096A	varying			000000		021500
1350	D	i_key		const	like(RC4_key_t)			000000		021600
1351	D	i_string	4096A		varying options(*varsize)			000000		021700
1352	*							000000		021800
1353	*	Return value						000000		021900
1354	D	digest	S	4096A	varying inz			000000		022000
1355	*							000000		022100
1356	*	Local fields						000000		022200
1357	D	controls	DS		likeds(cipherCtrls_0013_t) inz			000000		022300
1358	D	rc4_ctx	DS		likeds(rc4_ctx_t) inz			000000		022400
1359	*							000000		022500
1360		/FREE						000000		022600
1361								000000		022700

Riga	Specifiche origine	Commenti	Es	Pag.	Modif.	OrigSeq.
Numero			Num	Riga	Data	Id Numero
1362	if (i_string = '');		E01		000000	022800
1363	return '';		01		000000	022900
1364	endif;		E01		000000	023000
1365					000000	023100
1366	rc4_ctx = *ALLx'00';				000000	023200
1367	%subst(rc4_ctx.stream: 1: %len(i_key)) = i_key;				000000	023300
1368	rc4_ctx.length = %len(i_key);				000000	023400
1369	rc4_ctx.reserved = *ALLx'00';				000000	023500
1370					000000	023600
1371	controls = *ALLx'00';				000000	023700
1372					000000	023800
1373	controls.function = x'0013'; // RC4				000000	023900
1374	controls.dataLength = %len(i_string);				000000	024000
1375	controls.operation = cCIPHER_ENCRYPT; // Hex 00 = Encrypt				000000	024100
1376	// Hex 01 = Decrypt				000000	024200
1377	controls.reserved = *ALLx'00';				000000	024300
1378	controls.keyCtxPtr = %addr(rc4_ctx);				000000	024400
1379					000000	024500
1380	%len(digest) = %len(i_string);				000000	024600
1381	cipher(%addr(digest)+2: %addr(controls): %addr(i_string)+2);				000000	024700
1382					000000	024800
1383	return digest;				000000	024900
1384					000000	025000
1385	/END-FREE				000000	025100
1386	*				000000	025200
1387	P RC4...				000000	025300
1388	P E				000000	025400
1389	*				000000	025500
1390	*=====				000000	025600
1391	* *** Exported because, internally used by NTLMR4, RPGUNIT tests ***				R 000000	025700
1392	* Encrypts a given string using the DES algorithm				000000	025800
1393	*=====				000000	025900
1394	P DES...				000000	026000
1395	P B export				000000	026100
1396	*				000000	026200
1397	D DES...				000000	026300
1398	D PI 4096A varying				000000	026400
1399	D i_string 4096A const varying options(*varsize)				000000	026500
1400	D i_challenge const like(ntlm_challenge_t)				000000	026600
1401	*				000000	026700
1402	* Return value				000000	026800
1403	D encrypted S 4096A varying inz				000000	026900
1404	*				000000	027000
1405	* Local fields				000000	027100
1406	D tmpEncrypted S 4096A inz				000000	027200
1407	D encryptedLen S 10I 0 inz				000000	027300
1408	*				000000	027400
1409	D algd0200 DS likeds(algd0200_t) inz				000000	027500
1410	D keyd0200 DS likeds(keyd0200_t) inz				000000	027600
1411	D errCode DS likeds(errCode_t) inz				000000	027700
1412	* - - - - -				000000	027800
1413	/FREE				000000	027900
1414					000000	028000

Riga	Specifiche origine	Commenti	Es	Pag.	Modif.	OrigSeq.
Numero			Num	Riga	Data	Id Numero
1415	if (i_string = '');		E01		000000	028100
1416	return i_string;		01		000000	028200
1417	endif;		E01		000000	028300
1418					000000	028400
1419	algd0200 = *ALLx'00';				000000	028500
1420	algd0200.algorithm = 20;				000000	028600
1421	algd0200.blockLen = 8;				000000	028700
1422	algd0200.mode = '0';				000000	028800
1423	algd0200.padOption = '1';				000000	028900
1424	algd0200.padChar = x'00';				000000	029000
1425	algd0200.reserved_1 = *ALLx'00';				000000	029100
1426	algd0200.macLen = 0;				000000	029200
1427	algd0200.keySize = 0;				000000	029300
1428	algd0200.initVector = *ALLx'00';				000000	029400
1429					000000	029500
1430	keyd0200 = *ALLx'00';				000000	029600
1431	keyd0200.type = 20;				000000	029700
1432	keyd0200.length = 8;				000000	029800
1433	keyd0200.format = '0';				000000	029900
1434	keyd0200.reserved_1 = *ALLx'00';				000000	030000
1435	keyd0200.value = i_challenge;				000000	030100
1436					000000	030200
1437	clear errCode;				000000	030300
1438					000000	030400
1439	// Qc3EncryptData(i_string	// Clear data			150112	030500
=====>aa						
=====>bb						
*RNF5505 30 a	030500	Nelle specifiche di calcolo a formato libero le posizioni 6-7 non sono in bianco.				
*RNF0274 30 b	030500	Direttiva di compilazione sconosciuta; direttiva ignorata.				
1440 //		: %len(i_string) // Length of clear data			150112	030600
=====>aa						
=====>bb						
*RNF5505 30 a	030600	Nelle specifiche di calcolo a formato libero le posizioni 6-7 non sono in bianco.				
*RNF0274 30 b	030600	Direttiva di compilazione sconosciuta; direttiva ignorata.				
1441 //		: 'DATA0100' // Clear data format name			150112	030700
=====>aa						
=====>bb						
*RNF5505 30 a	030700	Nelle specifiche di calcolo a formato libero le posizioni 6-7 non sono in bianco.				
*RNF0274 30 b	030700	Direttiva di compilazione sconosciuta; direttiva ignorata.				
1442 //		: algd0200 // Algorithm description			150112	030800
=====>aa						
=====>bb						
*RNF5505 30 a	030800	Nelle specifiche di calcolo a formato libero le posizioni 6-7 non sono in bianco.				
*RNF0274 30 b	030800	Direttiva di compilazione sconosciuta; direttiva ignorata.				
1443 //		: 'ALGD0200' // Algorithm description format name			150112	030900
=====>aa						
=====>bb						
*RNF5505 30 a	030900	Nelle specifiche di calcolo a formato libero le posizioni 6-7 non sono in bianco.				
*RNF0274 30 b	030900	Direttiva di compilazione sconosciuta; direttiva ignorata.				
1444 //		: keyd0200 // Key description			150112	031000
=====>aa						

Riga	Numero	Specifiche origine	Commenti	Es	Pag.	Modif.	OrigSeq.
1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8.....+.....9.....+.....10			Num	Riga	Data	Id
=====>	bb						
*RNF5505	30 a	031000	Nelle specifiche di calcolo a formato libero le posizioni 6-7 non sono in bianco.				
*RNF0274	30 b	031000	Direttiva di compilazione sconosciuta; direttiva ignorata.				
1445	//	:	'KEYD0200' // Key description format name			150112	031100
=====>	aa						
=====>	bb						
*RNF5505	30 a	031100	Nelle specifiche di calcolo a formato libero le posizioni 6-7 non sono in bianco.				
*RNF0274	30 b	031100	Direttiva di compilazione sconosciuta; direttiva ignorata.				
1446	//	:	'0' // Cryptographic service provider			150112	031200
=====>	aa						
=====>	bb						
*RNF5505	30 a	031200	Nelle specifiche di calcolo a formato libero le posizioni 6-7 non sono in bianco.				
*RNF0274	30 b	031200	Direttiva di compilazione sconosciuta; direttiva ignorata.				
1447	//	:	' ' // Cryptographic device name			150112	031300
=====>	aa						
=====>	bb						
*RNF5505	30 a	031300	Nelle specifiche di calcolo a formato libero le posizioni 6-7 non sono in bianco.				
*RNF0274	30 b	031300	Direttiva di compilazione sconosciuta; direttiva ignorata.				
1448	//	:	tmpEncrypted // Encrypted data			150112	031400
=====>	aa						
=====>	bb						
*RNF5505	30 a	031400	Nelle specifiche di calcolo a formato libero le posizioni 6-7 non sono in bianco.				
*RNF0274	30 b	031400	Direttiva di compilazione sconosciuta; direttiva ignorata.				
1449	//	:	%size(tmpEncrypted) // Length of area provided for encrypted data			150112	031500
=====>	aa						
=====>	bb						
*RNF5505	30 a	031500	Nelle specifiche di calcolo a formato libero le posizioni 6-7 non sono in bianco.				
*RNF0274	30 b	031500	Direttiva di compilazione sconosciuta; direttiva ignorata.				
1450	//	:	encryptedLen // Length of encrypted data returned			150112	031600
=====>	aa						
=====>	bb						
*RNF5505	30 a	031600	Nelle specifiche di calcolo a formato libero le posizioni 6-7 non sono in bianco.				
*RNF0274	30 b	031600	Direttiva di compilazione sconosciuta; direttiva ignorata.				
1451	//	:	errCode); // Error code			150112	031700
=====>	aa						
=====>	bb						
*RNF5505	30 a	031700	Nelle specifiche di calcolo a formato libero le posizioni 6-7 non sono in bianco.				
*RNF0274	30 b	031700	Direttiva di compilazione sconosciuta; direttiva ignorata.				
1452						000000	031800
1453			encrypted = %subst(tmpEncrypted: 1: encryptedLen);			000000	031900
1454						000000	032000
1455			return encrypted;			000000	032100
1456						000000	032200
1457			/END-FREE			000000	032300
1458	*					000000	032400
1459	P DES...					000000	032500
1460	P	E				000000	032600
1461	*					000000	032700

Riga	----- Specifiche origine -----><----- Commenti ----->	Es	Pag.	Modif.	OrigSeq.	
Numero1.....2.....3.....4.....5.....6.....7.....8.....9.....10	Num	Riga	Data	Id	Numero
1462	*=====*			000000		032800
1463	* *** Exported because, internally used by NTLMR4, RPGUNIT tests ***			000000	R	032900
1464	* Calculates a DES key from a given key value.			000000		033000
1465	*=====*			000000		033100
1466	P DES_produceKey...			000000		033200
1467	P B export			000000		033300
1468	*			000000		033400
1469	D DES_produceKey...			000000		033500
1470	D PI like(DES_key_t)			000000		033600
1471	D i_value 7A const			000000		033700
1472	*			000000		033800
1473	* Return value			000000		033900
1474	D desKey S like(DES_key_t) inz(x'00')			000000		034000
1475	*			000000		034100
1476	* Local fields			000000		034200
1477	D x S 10I 0 inz			000000		034300
1478	D i S 10I 0 inz			000000		034400
1479	D char S 1A inz(x'00')			000000		034500
1480	*			000000		034600
1481	D inp DS qualified			000000		034700
1482	D lm_byte 1 1A			000000		034800
1483	D lm_int4 1 4U 0			000000		034900
1484	D lm_int4_4 4 4A			000000		035000
1485	D rm_int4_1 5 5A			000000		035100
1486	D rm_int4 5 8U 0			000000		035200
1487	D value 1 7A			000000		035300
1488	D rm_byte 8 8A inz(x'00')			000000		035400
1489	*			000000		035500
1490	D tmp DS qualified			000000		035600
1491	D int4 1 4I 0 inz			000000		035700
1492	D byte 4 4 inz(x'00')			000000		035800
1493	* - - - - -			000000		035900
1494	/FREE			000000		036000
1495				000000		036100
1496	inp = *ALLx'00';			000000		036200
1497	inp.value = i_value;			000000		036300
1498				000000		036400
1499	for x = 1 to 8;		B01	000000		036500
1500				000000		036600
1501	// get the 7 left most bits of the left most byte			000000		036700
1502	char = byteand(inp.lm_byte: x'FE');		01	000000		036800
1503				000000		036900
1504	// set parity bit			000000		037000
1505	char = setParityBit(char);		01	000000		037100
1506				000000		037200
1507	// put result into the DES key value			000000		037300
1508	i = i + 1;		01	000000		037400
1509	%subst(desKey: i: 1) = char;		01	000000		037500
1510				000000		037600
1511	// shift the 4 left most bytes 7 bits to the left			000000		037700
1512	inp.lm_int4 = bitand(inp.lm_int4: x'01FFFFFF');		01	000000		037800
1513	inp.lm_int4 = inp.lm_int4 * 128;		01	000000		037900

Riga	Specifiche origine	Commenti	Es	Pag.	Modif.	Orig	Seq.
Numero1.....2.....3.....4.....5.....6.....7.....8.....9.....10		Num	Riga	Data	Id	Numero
1514					000000		038000
1515	// shift the 7 left most bits of byte 5 to byte 4				000000		038100
1516	tmp.byte = byteand(inp.rm_int4_1: x'FE');		01		000000		038200
1517	tmp.int4 = tmp.int4 / 2;		01		000000		038300
1518	inp.lm_int4_4 = byteor(inp.lm_int4_4: tmp.byte);		01		000000		038400
1519					000000		038500
1520	// shift the 4 right most bytes 7 bits to the left				000000		038600
1521	inp.rm_int4 = bitand(inp.rm_int4: x'01FFFFFF');		01		000000		038700
1522	inp.rm_int4 = inp.rm_int4 * 128;		01		000000		038800
1523					000000		038900
1524	endfor;			E01	000000		039000
1525					000000		039100
1526	return desKey;				000000		039200
1527					000000		039300
1528	/END-FREE				000000		039400
1529	*				000000		039500
1530	P DES_produceKey...				000000		039600
1531	P E				000000		039700
1532	*				000000		039800
1533	*=====*				000000		039900
1534	* *** Private ***			R	000000		040000
1535	* Calculates the odd parity bit for each byte				000000		040100
1536	* of a given value.				000000		040200
1537	*=====*				000000		040300
1538	P setParityBit...				000000		040400
1539	P B				000000		040500
1540	*				000000		040600
1541	D setParityBit...				000000		040700
1542	D PI 1A				000000		040800
1543	D i_char 1A const				000000		040900
1544	*				000000		041000
1545	* Return value				000000		041100
1546	D char S 1A inz				000000		041200
1547	*				000000		041300
1548	* Local fields				000000		041400
1549	D x S 10I 0 inz				000000		041500
1550	D numlBits S 10I 0 inz				000000		041600
1551	*				000000		041700
1552	D bit S 1A dim(7)				000000		041800
1553	* - - - - -				000000		041900
1554	/FREE				000000		042000
1555					000000		042100
1556	bit(7) = byteand(i_char: x'80') = x'80';				000000		042200
1557	bit(6) = byteand(i_char: x'40') = x'40';				000000		042300
1558	bit(5) = byteand(i_char: x'20') = x'20';				000000		042400
1559	bit(4) = byteand(i_char: x'10') = x'10';				000000		042500
1560	bit(3) = byteand(i_char: x'08') = x'08';				000000		042600
1561	bit(2) = byteand(i_char: x'04') = x'04';				000000		042700
1562	bit(1) = byteand(i_char: x'02') = x'02';				000000		042800
1563					000000		042900
1564	for x = 1 to %elem(bit);		B01		000000		043000
1565	if (bit(x)) = '1';		B02		000000		043100

Riga	<----- Specifiche origine ----->	<----- Commenti ----->	Es	Pag.	Modif.	OrigSeq.
Numero1.....2.....3.....4.....5.....6.....7.....8.....9.....10		Num	Riga	Data	Id
1566	numlBits = numlBits + 1;		02		000000	043200
1567	endif;		E02		000000	043300
1568	endfor;		E01		000000	043400
1569					000000	043500
1570	if (%rem(numlBits: 2) = 0);		B01		000000	043600
1571	char = byteor(i_char: x'01');		01		000000	043700
1572	else;		X01		000000	043800
1573	char = byteand(i_char: x'FE');		01		000000	043900
1574	endif;		E01		000000	044000
1575					000000	044100
1576	return char;				000000	044200
1577					000000	044300
1578	/END-FREE				000000	044400
1579	*				000000	044500
1580	P setParityBit...				000000	044600
1581	P E				000000	044700
1582	*				000000	044800
1583	*=====*				000000	044900
1584	* *** Exported because, internally used by NTLMR4, RPGUNIT tests ***			R	000000	045000
1585	* MD4 operation.				000000	045100
1586	*=====*				000000	045200
1587	P MD4...				000000	045300
1588	P B export				000000	045400
1589	*				000000	045500
1590	D MD4...				000000	045600
1591	D PI like(MD4_digest_t)				000000	045700
1592	D i_string 4096A varying options(*varsize)				000000	045800
1593	*				000000	045900
1594	* Return value				000000	046000
1595	D digest S like(MD4_digest_t) inz				000000	046100
1596	* - - - - -				000000	046200
1597	/FREE				000000	046300
1598					000000	046400
1599	MD4Only_r(digest: %addr(i_string)+2: %len(i_string));				000000	046500
1600					000000	046600
1601	return digest;				000000	046700
1602					000000	046800
1603	/END-FREE				000000	046900
1604	*				000000	047000
1605	P MD4...				000000	047100
1606	P E				000000	047200
1607	*				000000	047300
1608	*=====*				000000	047400
1609	* *** Exported because, internally used by NTLMR4, RPGUNIT tests ***			R	000000	047500
1610	* Returns the MD5 digest of a given string.				000000	047600
1611	*=====*				000000	047700
1612	P MD5Hmac...				000000	047800
1613	P B export				000000	047900
1614	*				000000	048000
1615	D MD5Hmac...				000000	048100

Riga	Specifiche origine	Commenti	Es	Pag.	Modif.	Orig	Seq.
Numero1.....2.....3.....4.....5.....6.....7.....8.....9.....10	Num Riga	Data	Id	Numero		
1616	D	PI	like(MD5_digest_t)		000000		048200
1617	D	i_hmacKey	const like(MD5_digest_t)		000000		048300
1618	D	i_string	4096A varying options(*varsize)		000000		048400
1619	*				000000		048500
1620	*	Return value			000000		048600
1621	D	digest	S like(MD5_digest_t)		000000		048700
1622	*				000000		048800
1623	*	Local fields			000000		048900
1624	D	controls	DS likeds(cipherCtrls_0005_t) inz		000000		049000
1625	D	MD5_CTX	DS likeds(MD5_CTX_t) inz		000000		049100
1626	*	- - - - -			000000		049200
1627		/FREE			000000		049300
1628					000000		049400
1629		if (i_string = '');			E01		049500
1630		return '';			01		049600
1631		endif;			E01		049700
1632							049800
1633		MD5_CTX.key = i_hmacKey;					049900
1634		MD5_CTX.context = *ALLx'00';					050000
1635		MD5_CTX.state = cCIPHER_ONLY;					050100
1636		MD5_CTX.digest = *ALLx'00';					050200
1637							050300
1638		controls = *ALLx'00';					050400
1639							050500
1640		controls.function = x'0005'; // MD5 or SHA-1					050600
1641		controls.hashAlg = cCIPHER_MD5; // Hex 00 = MD5					050700
1642		// Hex 01 = SHA-1					050800
1643		controls.sequence = MD5_CTX.state; // Hex 00 = Only					050900
1644		// Hex 01 = First					051000
1645		// Hex 02 = Middle					051100
1646		// Hex 03 = Final					051200
1647		controls.dataLength = %len(i_string);					051300
1648		controls.output = cCIPHER_HMAC; // Hex 00 = Hash					051400
1649		// Hex 01 = HMAC					051500
1650		controls.hashContext = %addr(MD5_CTX.context);					051600
1651		controls.HMACKey = %addr(MD5_CTX.key);					051700
1652		controls.HMACKeyLength = %size(MD5_CTX.key);					051800
1653							051900
1654		cipher(%addr(MD5_CTX.digest): %addr(controls): %addr(i_string)+2);					052000
1655							052100
1656		digest = MD5_CTX.digest;					052200
1657							052300
1658		return digest;					052400
1659							052500
1660		/END-FREE					052600
1661	*						052700
1662	P	MD5Hmac...					052800
1663	P	E					052900
1664	*						053000
1665	*	*****					053100
1666	*	*** Exported because, internally used by NTLMR4, RPGUNIT tests ***			R		053200
1667	*	Returns the MD5 digest of a given string.					053300
1668	*	*****					053400
1669	P	MD5Digest...					053500

Riga	Specifiche origine	Commenti	Es	Pag.	Modif.	OrigSeq.	
Numero	1...10	1...10	Num	Riga	Data	Id	Numero
1670	P	B			000000		053600
1671	*	export			000000		053700
1672	D	MD5Digest...			000000		053800
1673	D	PI	like(MD5_digest_t)		000000		053900
1674	D	i_string	4096A	varying options(*varsize)	000000		054000
1675	*				000000		054100
1676	*	Return value			000000		054200
1677	D	digest	S	like(MD5_digest_t)	000000		054300
1678	*				000000		054400
1679	*	Local fields			000000		054500
1680	D	controls	DS	likeds(cipherCtrls_0005_t) inz	000000		054600
1681	D	MD5_CTX	DS	likeds(MD5_CTX_t) inz	000000		054700
1682	*				000000		054800
1683		/FREE			000000		054900
1684					000000		055000
1685		if (i_string = '');			B01	000000	055100
1686		return '';			01	000000	055200
1687		endif;			E01	000000	055300
1688						000000	055400
1689		MD5_CTX.key = *ALLx'00';				000000	055500
1690		MD5_CTX.context = *ALLx'00';				000000	055600
1691		MD5_CTX.state = cCIPHER_ONLY;				000000	055700
1692		MD5_CTX.digest = *ALLx'00';				000000	055800
1693						000000	055900
1694		controls = *ALLx'00';				000000	056000
1695						000000	056100
1696		controls.function = x'0005'; // MD5 or SHA-1				000000	056200
1697		controls.hashAlg = cCIPHER_MD5; // Hex 00 = MD5				000000	056300
1698		// Hex 01 = SHA-1				000000	056400
1699		controls.sequence = MD5_CTX.state; // Hex 00 = Only				000000	056500
1700		// Hex 01 = First				000000	056600
1701		// Hex 02 = Middle				000000	056700
1702		// Hex 03 = Final				000000	056800
1703		controls.dataLength = %len(i_string);				000000	056900
1704		controls.output = cCIPHER_HASH; // Hex 00 = Hash				000000	057000
1705		// Hex 01 = HMAC				000000	057100
1706		controls.hashContext = %addr(MD5_CTX.context);				000000	057200
1707		controls.HMACKey = *NULL;				000000	057300
1708		controls.HMACKeyLength = 0;				000000	057400
1709						000000	057500
1710		cipher(%addr(MD5_CTX.digest): %addr(controls): %addr(i_string)+2);				000000	057600
1711						000000	057700
1712		digest = MD5_CTX.digest;				000000	057800
1713						000000	057900
1714		return digest;				000000	058000
1715						000000	058100
1716		/END-FREE				000000	058200
1717	*					000000	058300
1718	P	MD5Digest...			000000		058400
1719	P	E			000000		058500
1720	*				000000		058600

* * * * * F I N E O R I G I N E * * * * *

Riga	<----- Specifiche origine ----->	<----- Commenti ----->	Es	Pag.	Modif.	OrigSeq.
Numero 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10	Num Riga	Data	Id	Numero	

M e s s a g g i d i d i a g n o s t i c a a g g i u n t i v i

Id msg Sv Numero Seq. Testo messaggio
 *RNF7226 00 1490 035600 La struttura dati TMP ha inizializzazioni di sottocampi che
 si sovrappongono.

* * * * * F I N E M E S S A G G I D I D I A G N O S T I C A A G G I U N T I V I * * * * *

/ C o p i a M e m b r o

Riga Orig Nome RPG <----- Nome esterno -----> CCSID <- Ult. modif. ->

Numero	Id	Libreria	File	Membro	Data	Ora
57	1	NTLM_H	LIBHTTP	QRPGLESRC	NTLM_H	37 02/01/15 15:37:46
169	2	NTLM_P	LIBHTTP	QRPGLESRC	NTLM_P	37 02/01/15 16:12:17
991	3	MD4_H	LIBHTTP	QRPGLESRC	MD4_H	37 02/01/15 16:10:21
1158	4	MD4_H	LIBHTTP	QRPGLESRC	MD4_H	37 02/01/15 16:10:21

* * * * * F I N E / C O P I A M E M B R I * * * * *

R i f e r i m e n t i i n c r o c i a t i

Riferimenti file e record:

File Record	Unit{	Riferimenti (D=Definito)
----------------	-------	--------------------------

Non ci sono riferimenti nell'origine.

Riferimenti campo globali:

Campo	Attributi	Riferimenti (D=Defin. M=Modific.)			
*RNF7031 ALGD0200_T	DS(54) BASED(PDUMMY)	1319D	1409		
ALGORITHM	I(10,0)	1320D			
BLOCKLEN	I(10,0)	1321D			
INITVECTOR	A(32)	1328D			
KEYSIZE	I(10,0)	1327D			
MACLEN	I(10,0)	1326D			
MODE	A(1)	1322D			
PADCHAR	A(1)	1324D			
PADOPTION	A(1)	1323D			
RESERVED_1	A(3)	1325D			
AUTHENTICATE_MESSAGE...	CONST	288D			
*RNF7031 AUTHPLUGIN_GETREALM...	A(124) VARYING PROTOTYPE	82D			
*RNF7031 AUTHPLUGIN_INTERPRETAUTHENTICATIONHEADER...	PROTOTYPE	116D			
*RNF7031 AUTHPLUGIN_ISAUTHENTICATIONREQUIRED...	N(1) PROTOTYPE	74D			
*RNF7031 AUTHPLUGIN_MUSTRECEIVEAUTHERRORPAGE...	N(1) PROTOTYPE	131D			
*RNF7031 AUTHPLUGIN_NEGOTIATEAUTHENTICATION...	I(10,0) PROTOTYPE	147D			
*RNF7031 AUTHPLUGIN_PRODUCEAUTHENTICATIONHEADER...	PROTOTYPE	161D			
*RNF7031 AUTHPLUGIN_RESETAUTHENTICATION...	PROTOTYPE	106D			
*RNF7031 AUTHPLUGIN_SETAUTHENTICATION...	N(1) PROTOTYPE	95D			
BITAND	U(20,0) PROTOTYPE	841D	1512	1521	
*RNF7031 BITOR	U(20,0) PROTOTYPE	859D			
BYTEAND	A(1) PROTOTYPE	850D	1502	1516	1556
		1557	1558	1559	1560
		1561	1562	1573	
BYTEOR	A(1) PROTOTYPE	868D	1518	1571	
CCIPHER_DECRYPT	CONST	1280D			
CCIPHER_ENCRYPT	CONST	1278D	1375		

	CCIPHER_HASH	CONST	1285D	1704		
	CCIPHER_HMAC	CONST	1287D	1648		
	CCIPHER_MD5	CONST	1283D	1641	1697	
	CCIPHER_ONLY	CONST	1289D	1635	1691	
	CFALSE	CONST	204D			
	CHALLENGE_MESSAGE	CONST	286D			
	CIPHER	PROTOTYPE	1251D	1381M	1654M	1710M
*RNF7031	CIPHERCTRLS_0005_T...					
		DS(96)	1257D	1624	1680	
		BASED(PDUMMY)				
	DATALENGTH	U(10,0)	1261D			
	FUNCTION	A(2)	1258D			
	HASHALG	A(1)	1259D			
	HASHCONTEXT	*(16)	1264D			
	HMACKEY	*(16)	1265D			
	HMACKEYLENGTH	U(10,0)	1266D			
	OUTPUT	A(1)	1262D			
*RNF7031	RESERVED_1	A(7)	1263D			
*RNF7031	RESERVED_2	A(44)	1267D			
	SEQUENCE	A(1)	1260D			
*RNF7031	CIPHERCTRLS_0013_T...					
		DS(32)	1270D	1357		
		BASED(PDUMMY)				
	DATALENGTH	U(5,0)	1272D			
	FUNCTION	A(2)	1271D			
	KEYCTXPTR	*(16)	1275D			
	OPERATION	A(1)	1273D			
	RESERVED	A(11)	1274D			
*RNF7031	COMPUTERESPONSE	A(2048)	774D			
		VARYING				
		PROTOTYPE				
	CTRUE	CONST	203D			
	DEFAULT_LM_COMPATIBILITY_MODE...					
		CONST	271D			
	DES	A(4096)	973D	1394	1398	1459
		VARYING				
		PROTOTYPE				
*RNF7031	DES_KEY_T	A(8)	969D	982	1470	1474
		BASED(PDUMMY)				
	DES_PRODUCEKEY	A(8)	982D	1466	1470	1530
		PROTOTYPE				
*RNF7031	ERRCODE_T	DS(8)	244D	1411		
		BASED(PDUMMY)				
*RNF7031	BYTAVL	I(10,0)	246D			
*RNF7031	BYTPRV	I(10,0)	245D			
	E2BIG_C	CONST	1242D			
*RNF7031	FREEASCIIITRANSCODER...					
		PROTOTYPE	816D			
*RNF7031	FREEUNICODETRANSCODER...					
		PROTOTYPE	828D			
*RNF7031	GETLMRESPONSE	A(24)	711D			
		VARYING				
		PROTOTYPE				
*RNF7031	GETLMV2RESPONSE	A(24)	747D			
		VARYING				
		PROTOTYPE				
*RNF7031	GETNTLMV2RESPONSE	A(2048)	762D			
		VARYING				

	PROTOTYPE				
*RNF7031	GETNTLM2RESPONSE	A(24)	733D		
	VARYING				
	PROTOTYPE				
*RNF7031	GETNTRESPONSE	A(24)	722D		
	VARYING				
	PROTOTYPE				
*RNF7031	GETTOKEN	A(2048)	943D		
	VARYING				
	PROTOTYPE				
*RNF7031	HTRANSCODER_T	*(16)	480D		
	BASED(PDUMMY)				
*RNF7031	ICONV	U(10,0)	1234D		
	PROTOTYPE				
*RNF7031	ICONV_CLOSE	I(10,0)	1246D		
	PROTOTYPE				
	ICONV_ERROR	CONST	1241D		
*RNF7031	ICONV_T	DS(52)	1214	1218D	1235 1247
	BASED(PDUMMY)				
*RNF7031	CD(12)	I(10,0)	1220D		
*RNF7031	RETURN_VALUE	I(10,0)	1219D		
*RNF7031	ISBIT	N(1)	832D		
	PROTOTYPE				
*RNF7031	KEYD0200_T	DS(20)	1330D	1410	
	BASED(PDUMMY)				
	FORMAT	A(1)	1333D		
	LENGTH	I(10,0)	1332D		
	RESERVED_1	A(3)	1334D		
	TYPE	I(10,0)	1331D		
	VALUE	A(8)	1335D		
	LM_MODE_NTLM_V1	CONST	274D		
	LM_MODE_NTLM_V1_NO_LM...	CONST	276D		
	LM_MODE_NTLM_V1_NTLM2_ONLY...	CONST	278D		
	LM_MODE_NTLM_V2	CONST	280D		
	MD4	A(16)	1123D	1587	1591 1605
	PROTOTYPE				
*RNF7031	MD4_CTX_T	DS(88)	1032D	1045	1055 1067
	BASED(PDUMMY)		1085	1095	1107
*RNF7031	BUFFER	A(64)	1037D		
*RNF7031	COUNT(2)	U(10,0)	1035D		
*RNF7031	COUNTA	A(8)	1036D		
*RNF7031	STATE(4)	U(10,0)	1033D		
*RNF7031	STATEA	A(16)	1034D		
*RNF7031	MD4_DIGEST_T	A(16)	786	1029D	1066 1075
	BASED(PDUMMY)		1106	1115	1123 1591
			1595		
*RNF7031	MD4FINAL_C	PROTOTYPE	1102D		
*RNF7031	MD4FINAL_R	PROTOTYPE	1062D		
*RNF7031	MD4INIT_C	PROTOTYPE	1081D		
*RNF7031	MD4INIT_R	PROTOTYPE	1041D		
*RNF7031	MD4ONLY_C	PROTOTYPE	1111D		
	MD4ONLY_R	PROTOTYPE	1071D	1599M	
*RNF7031	MD4UPDATE_C	PROTOTYPE	1091D		
*RNF7031	MD4UPDATE_R	PROTOTYPE	1051D		
*RNF7031	MD5_CTX_T	DS(193)	1291D	1625	1681
	BASED(PDUMMY)				

	CONTEXT	A(160)	1293D				
	DIGEST	A(16)	1295D				
	KEY	A(16)	1292D				
	STATE	A(1)	1294D				
*RNF7031	MD5_DIGEST_T	A(16)	768	778	795	1134D	
		BASED(PDUMMY)	1138	1142	1149	1616	
			1617	1621	1673	1677	
	MD5DIGEST	A(16)	1149D	1669	1673	1718	
		PROTOTYPE					
	MD5HMAC	A(16)	1138D	1612	1616	1662	
		PROTOTYPE					
*RNF7031	MESSAGE_DECODEBASE	64...					
		A(2048)	693D				
		VARYING					
		PROTOTYPE					
*RNF7031	MESSAGE_ENCODEBASE	64...					
		A(2048)	681D				
		VARYING					
		PROTOTYPE					
*RNF7031	MESSAGE_GETCHALLENGE	...					
		A(8)	597D				
		PROTOTYPE					
*RNF7031	MESSAGE_GETTARGETDNSCOMPUTERNAME	...					
		C(1024)	646D				
		VARYING					
		CCSID(13488)					
		PROTOTYPE					
*RNF7031	MESSAGE_GETTARGETDNSDOMAINNAME	...					
		C(1024)	658D				
		VARYING					
		CCSID(13488)					
		PROTOTYPE					
*RNF7031	MESSAGE_GETTARGETDNSSTRENAME	...					
		C(1024)	670D				
		VARYING					
		CCSID(13488)					
		PROTOTYPE					
*RNF7031	MESSAGE_GETTARGETNBCOMPUTERNAME	...					
		C(1024)	622D				
		VARYING					
		CCSID(13488)					
		PROTOTYPE					
*RNF7031	MESSAGE_GETTARGETNBDOMAINNAME	...					
		C(1024)	634D				
		VARYING					
		CCSID(13488)					
		PROTOTYPE					
*RNF7031	MESSAGE_HASFLAG	N(1)	609D				
		PROTOTYPE					
*RNF7031	MESSAGE_ISTYPE1	N(1)	564D				
		PROTOTYPE					
*RNF7031	MESSAGE_ISTYPE2	N(1)	575D				
		PROTOTYPE					
*RNF7031	MESSAGE_ISTYPE3	N(1)	586D				
		PROTOTYPE					
*RNF7031	MESSAGE_NEWTYPE1	A(2048)	512D				
		VARYING					
		PROTOTYPE					


```

*RNF7031 MESSAGE_NEWTYPE3  A(2048)          546D
                          VARYING
                          PROTOTYPE
*RNF7031 MESSAGE_VALIDATETYPE2...
                          N(1)            529D
                          PROTOTYPE
                          MILLISECONFS_BETWEEN_1970_AND_1601...
                          CONST            258D
                          NEGOTIATE_MESSAGE CONST 284D
                          NTLM_AUTHENTICATE CONST 461D
*RNF7031 NTLM_CHALLENGE_T  A(8)            223D      390      597      740
                          BASED(PDUMMY)    742      757      770      780
                          978      1400
*RNF7031 NTLM_DOMAIN_T    A(512)          208D      520      556      753
                          BASED(PDUMMY)    803
                          VARYING
                          NTLM_EINV_ENCODING...
                          CONST            471D
                          NTLM_EINV_TYPE1_MSG...
                          CONST            465D
                          NTLM_EINV_TYPE2_MSG...
                          CONST            467D
                          NTLM_EINV_TYPE3_MSG...
                          CONST            469D
*RNF7031 NTLM_ENABLETESTMODE...
                          PROTOTYPE      484D
                          NTLM_ENSUP_ENCODING...
                          CONST            473D
*RNF7031 NTLM_HOST_T      A(512)          212D
                          BASED(PDUMMY)
                          VARYING
*RNF7031 NTLM_LMRESPONSE_T A(24)          236D      711      747
                          BASED(PDUMMY)
                          VARYING
*RNF7031 NTLM_MESSAGE_T  A(2048)          220D      512      533      535
                          BASED(PDUMMY)    546      550      568      579
                          VARYING         590      601      613      626
                          638      650      662      674
                          681      685      693      697
                          NTLM_NEGOTIATE   CONST 459D
                          NTLM_NONE       CONST 457D
*RNF7031 NTLM_NTLMRESPONSE_T...
                          A(2048)          240D      762      774
                          BASED(PDUMMY)
                          VARYING
*RNF7031 NTLM_NTRESPONSE_T A(24)          238D      722      733
                          BASED(PDUMMY)
                          VARYING
*RNF7031 NTLM_PASSWORD_T  A(1024)         234D      554      717      728
                          BASED(PDUMMY)    737      755      790      799
                          VARYING
*RNF7031 NTLM_SECURITYBUFFER_T...
                          DS(8)            372      373      379D      388
                          BASED(PDUMMY)    392      442      443      444
                          445      446      447
*RNF7031 LENGTH          U(5,0)          380D
*RNF7031 MAXLEN          U(5,0)          381D
*RNF7031 OFFSET          U(10,0)         382D
    
```

*RNF7031	NTLM_SESSIONKEY_T	A(16)	242D			
		BASED(PDUMMY)				
		VARYING				
*RNF7031	NTLM_SETLMCOMPATIBILITY...	PROTOTYPE	499D			
	NTLM_TARGET_TYPE_DNS_COMPUTER_NAME...	CONST	426D			
	NTLM_TARGET_TYPE_DNS_DOMAIN_NAME...	CONST	428D			
	NTLM_TARGET_TYPE_DNS_TREE_NAME...	CONST	430D			
	NTLM_TARGET_TYPE_FLAGS...	CONST	432D			
	NTLM_TARGET_TYPE_NB_COMPUTER_NAME...	CONST	422D			
	NTLM_TARGET_TYPE_NB_DOMAIN_NAME...	CONST	424D			
	NTLM_TARGET_TYPE_NONE...	CONST	420D			
	NTLM_TARGET_TYPE_RESTRICTIONS...	CONST	436D			
	NTLM_TARGET_TYPE_TIMESTAMP...	CONST	434D			
*RNF7031	NTLM_TARGETNAME_T	C(1024)	226D	622	634	646
		BASED(PDUMMY)	658	670		
		VARYING				
		CCSID(13488)				
*RNF7031	NTLM_TARGETTYPE_T	I(10,0)	229D			
		BASED(PDUMMY)				
*RNF7031	NTLM_USER_T	A(124)	232D	552	754	801
		BASED(PDUMMY)				
		VARYING				
*RNF7031	NTLM_WORKSTATION_T...	A(512)	216D	517		
		BASED(PDUMMY)				
		VARYING				
*RNF7031	NTLMAUTHENTICATE_T...	DS(64)	439D			
		BASED(PDUMMY)				
*RNF7031	FLAGS	U(10,0)	448D			
*RNF7031	LM_RESP	DS(8)	442D			
*RNF7031	NTLM_RESP	DS(8)	443D			
*RNF7031	SESSIONKEY	DS(8)	447D			
*RNF7031	SIGNATURE	A(8)	440D			
*RNF7031	TARGETNAME	DS(8)	444D			
*RNF7031	TYPE	U(10,0)	441D			
*RNF7031	USERNAME	DS(8)	445D			
*RNF7031	WORKSTATION	DS(8)	446D			
*RNF7031	NTLMCHALLENGE_T	DS(56)	385D	716	727	752
		BASED(PDUMMY)	767			
*RNF7031	CHALLENGE	A(8)	390D			
*RNF7031	FLAGS	U(10,0)	389D			
*RNF7031	OS_VERSION	DS(8)	393D			
*RNF7031	RESERVED	A(8)	391D			
*RNF7031	SIGNATURE	A(8)	386D			
*RNF7031	TARGETINFO	DS(8)	392D			
*RNF7031	TARGETNAME	DS(8)	388D			
*RNF7031	TYPE	U(10,0)	387D			

*RNF7031	NTLMMESSAGE_T	DS(12)	362D
		BASED(PDUMMY)	
*RNF7031	SIGNATURE	A(8)	363D
*RNF7031	TYPE	U(10,0)	364D
*RNF7031	NTLMNEGOTIATE_T	DS(40)	368D
		BASED(PDUMMY)	
*RNF7031	DOMAIN	DS(8)	372D
*RNF7031	FLAGS	U(10,0)	371D
*RNF7031	OS_VERSION	DS(8)	374D
*RNF7031	SIGNATURE	A(8)	369D
*RNF7031	TYPE	U(10,0)	370D
*RNF7031	WORKSTATION	DS(8)	373D
	NTLMSSP_NEGOTIATE_ALWAYS_SIGN...		
		CONST	322D
	NTLMSSP_NEGOTIATE_CONNECTION_ANONYMOUS...		
		CONST	314D
	NTLMSSP_NEGOTIATE_DATAGRAM...		
		CONST	304D
	NTLMSSP_NEGOTIATE_IDENTIFY...		
		CONST	332D
	NTLMSSP_NEGOTIATE_KEY_EXCH...		
		CONST	356D
	NTLMSSP_NEGOTIATE_LM_KEY...		
		CONST	306D
	NTLMSSP_NEGOTIATE_NT_ONLY...		
		CONST	312D
	NTLMSSP_NEGOTIATE_NTLM...		
		CONST	310D
	NTLMSSP_NEGOTIATE_NTLM2...		
		CONST	330D
	NTLMSSP_NEGOTIATE_OEM...		
		CONST	294D
	NTLMSSP_NEGOTIATE_OEM_DOMAIN_SUPPLIED...		
		CONST	316D
	NTLMSSP_NEGOTIATE_OEM_WORKSTATION_SUPPLIED...		
		CONST	318D
	NTLMSSP_NEGOTIATE_SEAL...		
		CONST	302D
	NTLMSSP_NEGOTIATE_SIGN...		
		CONST	300D
	NTLMSSP_NEGOTIATE_TARGET_INFO...		
		CONST	338D
	NTLMSSP_NEGOTIATE_UNICODE...		
		CONST	292D
	NTLMSSP_NEGOTIATE_VERSION...		
		CONST	345D
	NTLMSSP_NEGOTIATE_128...		
		CONST	354D
	NTLMSSP_NEGOTIATE_56...		
		CONST	358D
	NTLMSSP_REQUEST_NON_NT_SESSION_KEY...		
		CONST	336D
	NTLMSSP_REQUEST_TARGET...		
		CONST	296D
	NTLMSSP_TARGET_TYPE_DOMAIN...		
		CONST	324D
	NTLMSSP_TARGET_TYPE_SERVER...		
		CONST	326D

```

NTLMSSP_TARGET_TYPE_SHARE...
NTLMSSP_UNUSED_R1  CONST          328D
NTLMSSP_UNUSED_R2  CONST          352D
NTLMSSP_UNUSED_R3  CONST          349D
NTLMSSP_UNUSED_R4  CONST          347D
NTLMSSP_UNUSED_R5  CONST          340D
NTLMSSP_UNUSED_R6  CONST          334D
NTLMSSP_UNUSED_R8  CONST          320D
NTLMSSP_UNUSED_R9  CONST          308D
NTLMSSP_UNUSED_R9  CONST          298D
*RNF7031 NTOWFV1      A(16)          786D
                    PROTOTYPE
*RNF7031 NTOWFV2      A(16)          795D
                    PROTOTYPE
*RNF7031 OS_VERSION_T DS(8)          374      393      411D
                    BASED(PDUMMY)
*RNF7031 BUILD        U(5,0)          414D
*RNF7031 MAJOR        U(3,0)          412D
*RNF7031 MINOR        U(3,0)          413D
*RNF7031 NTLM_REV     U(3,0)          416D
*RNF7031 RESERVED_1  A(3)          415D
PDUMMY              *(16)          208      212      216      220
                    223      226      229      232
                    234      236      238      240
                    242      244      251      362
                    368      379      385      399
                    404      411      439      480
                    953      969      1029     1032
                    1134     1218     1223     1257
                    1270     1291     1297     1319
                    1330
*RNF7031 QC3ENCRYPTDATA PROTOTYPE     1304D
*RNF7031 QJOB_T        DS(26)          251D
                    BASED(PDUMMY)
*RNF7031 NAME          A(10)          252D
*RNF7031 NBR           A(6)          254D
*RNF7031 USER          A(10)          253D
*RNF7031 QTQCODE_T    DS(36)          1215     1216     1223D
                    BASED(PDUMMY)
*RNF7031 CCSID         I(10,0)          1224D
*RNF7031 CONVERSIONA  I(10,0)          1225D
*RNF7031 ERROPTMXDDTA I(10,0)          1229D
*RNF7031 INPLENOPT    I(10,0)          1228D
*RNF7031 RESERVED     A(12)          1230D
*RNF7031 SHIFTSTATEA I(10,0)          1227D
*RNF7031 SUBSTITUTIONA I(10,0)          1226D
*RNF7031 QTQICNV_OPEN DS(52)          1213D
                    PROTOTYPE
*RNF7031 RANDOM        PROTOTYPE          918D
RC4                  A(4096)          957D     1345     1349     1387
                    VARYING
                    PROTOTYPE
*RNF7031 RC4_CTX_T    DS(264)          1297D     1358
                    BASED(PDUMMY)
                    LENGTH
                    RESERVED     A(6)          1300D
                    STREAM       A(256)          1298D
*RNF7031 RC4_KEY_T    A(16)          953D     961      1350

```

```

        BASED(PDUMMY)
        VARYING
*RNF7031 REMOVEAUTHFROMURL A(32767)          702D
        VARYING
        PROTOTYPE
        SETPARITYBIT A(1)          1207D    1505    1538    1542
        PROTOTYPE          1580
*RNF7031 STRINGLE A(1024)          901D
        VARYING
        PROTOTYPE
*RNF7031 TARGETINFO_T DS(2052)          404D
        BASED(PDUMMY)
*RNF7031 LENGTH U(5,0)          406D
*RNF7031 TYPE U(5,0)          405D
        UNICOD E C(1024)          407D
        CCSID(13488)
*RNF7031 VALUE A(2048)          408D
*RNF7031 TARGETNAMECHARS_T DS(2048)          399D
        BASED(PDUMMY)
        UNICOD E C(1024)          400D
        CCSID(13488)
*RNF7031 VALUE A(2048)          401D
*RNF7031 TOASCII A(2048)          808D
        VARYING
        PROTOTYPE
*RNF7031 TOLOWER A(1024)          935D
        VARYING
        PROTOTYPE
*RNF7031 TOUNICOD E A(4096)          820D
        VARYING
        PROTOTYPE
*RNF7031 TOUPPER A(1024)          927D
        VARYING
        PROTOTYPE
*RNF7031 UINT16LE U(5,0)          877D
        PROTOTYPE
*RNF7031 UINT32LE U(10,0)          885D
        PROTOTYPE
*RNF7031 UINT64LE U(20,0)          893D
        PROTOTYPE
*RNF7031 VARSTRINGLE A(1024)          910D
        VARYING
        PROTOTYPE
    
```

Riferimenti di campo per la sottoprecedura RC4

Campo	Attributi	Riferimenti (D=Defin. M=Modific.)			
I_KEY	A(16) VARYING	1350D	1367	1367	1368
I_STRING	A(4096) VARYING	1351D	1362	1374	1380
DIGEST	A(4096) VARYING	1381	1380M	1381	1383
CONTROLS	DS(32)	1357D	1371M	1373M	1374M
		1375M	1377M	1378M	1381
DATALENGTH	U(5,0)	1374			
FUNCTION	A(2)	1373			
KEYCTXPTR	*(16)	1378			
OPERATION	A(1)	1375			

RESERVED	A(11)	1377			
RC4_CTX	DS(264)	1358D	1366M	1367M	1368M
		1369M	1378		
LENGTH	U(5,0)	1368			
RESERVED	A(6)	1369			
STREAM	A(256)	1367			

Riferimenti di campo per la sottoprecedura DES

Campo	Attributi	Riferimenti (D=Defin. M=Modific.)			
I_STRING	A(4096)	1399D	1415	1416	
	VARYING				
I_CHALLENGE	A(8)	1400D	1435		
ENCRYPTED	A(4096)	1403D	1453M	1455	
	VARYING				
TMPENCRYPTED	A(4096)	1406D	1453		
ENCRYPTEDLEN	I(10,0)	1407D	1453		
ALGDO200	DS(54)	1409D	1419M	1420M	1421M
		1422M	1423M	1424M	1425M
		1426M	1427M	1428M	
ALGORITHM	I(10,0)	1420			
BLOCKLEN	I(10,0)	1421			
INITVECTOR	A(32)	1428			
KEYSIZE	I(10,0)	1427			
MACLEN	I(10,0)	1426			
MODE	A(1)	1422			
PADCHAR	A(1)	1424			
PADOPTION	A(1)	1423			
RESERVED_1	A(3)	1425			
KEYDO200	DS(20)	1410D	1430M	1431M	1432M
		1433M	1434M	1435M	
FORMAT	A(1)	1433			
LENGTH	I(10,0)	1432			
RESERVED_1	A(3)	1434			
TYPE	I(10,0)	1431			
VALUE	A(8)	1435			
ERRCODE	DS(8)	1411D	1437		

Riferimenti di campo per la sottoprecedura DES_PRODUCKEY

Campo	Attributi	Riferimenti (D=Defin. M=Modific.)			
I_VALUE	A(7)	1471D	1497		
DESKEY	A(8)	1474D	1509M	1526	
X	I(10,0)	1477D	1499		
I	I(10,0)	1478D	1508M	1508	1509
CHAR	A(1)	1479D	1502M	1505M	1505
		1509			
INP	DS(8)	1481D	1496M	1497M	1502
		1512M	1512	1513M	1513
		1516	1518M	1518	1521M
		1521	1522M	1522	
LM_BYTE	A(1)	1482D	1502		
LM_INT4	U(10,0)	1483D	1512	1512	1513
		1513			
LM_INT4_4	A(1)	1484D	1518	1518	
*RNF7031 RM_BYTE	A(1)	1488D			
RM_INT4	U(10,0)	1486D	1521	1521	1522
		1522			
RM_INT4_1	A(1)	1485D	1516		
VALUE	A(7)	1487D	1497		

TMP	DS(4)	1490D	1516M	1517M	1517
		1518			
BYTE	A(1)	1492D	1516	1518	
INT4	I(10,0)	1491D	1517	1517	
Riferimenti di campo per la sottoprecedura SETPARITYBIT					
Campo	Attributi	Riferimenti (D=Defin. M=Modific.)			
I_CHAR	A(1)	1543D	1556	1557	1558
		1559	1560	1561	1562
		1571	1573		
CHAR	A(1)	1546D	1571M	1573M	1576
X	I(10,0)	1549D	1564	1565	
NUM1BITS	I(10,0)	1550D	1566M	1566	1570
BIT(7)	A(1)	1552D	1556M	1557M	1558M
		1559M	1560M	1561M	1562M
		1564	1565		
Riferimenti di campo per la sottoprecedura MD4					
Campo	Attributi	Riferimenti (D=Defin. M=Modific.)			
I_STRING	A(4096)	1592D	1599	1599	
	VARYING				
DIGEST	A(16)	1595D	1599	1601	
Riferimenti di campo per la sottoprecedura MD5HMAC					
Campo	Attributi	Riferimenti (D=Defin. M=Modific.)			
I_HMACKEY	A(16)	1617D	1633		
I_STRING	A(4096)	1618D	1629	1647	1654
	VARYING				
DIGEST	A(16)	1621D	1656M	1658	
CONTROLS	DS(96)	1624D	1638M	1640M	1641M
		1643M	1647M	1648M	1650M
		1651M	1652M	1654	
DATALENGTH	U(10,0)	1647			
FUNCTION	A(2)	1640			
HASHALG	A(1)	1641			
HASHCONTEXT	*(16)	1650			
HMACKEY	*(16)	1651			
HMACKEYLENGTH	U(10,0)	1652			
OUTPUT	A(1)	1648			
SEQUENCE	A(1)	1643			
MD5_CTX	DS(193)	1625D	1633M	1634M	1635M
		1636M	1643	1650	1651
		1652	1654	1656	
CONTEXT	A(160)	1634	1650		
DIGEST	A(16)	1636	1654	1656	
KEY	A(16)	1633	1651	1652	
STATE	A(1)	1635	1643		
Riferimenti di campo per la sottoprecedura MD5DIGEST					
Campo	Attributi	Riferimenti (D=Defin. M=Modific.)			
I_STRING	A(4096)	1674D	1685	1703	1710
	VARYING				
DIGEST	A(16)	1677D	1712M	1714	
CONTROLS	DS(96)	1680D	1694M	1696M	1697M
		1699M	1703M	1704M	1706M
		1707M	1708M	1710	
DATALENGTH	U(10,0)	1703			
FUNCTION	A(2)	1696			

HASHALG	A(1)	1697			
HASHCONTEXT	*(16)	1706			
HMACKEY	*(16)	1707			
HMACKEYLENGTH	U(10,0)	1708			
OUTPUT	A(1)	1704			
SEQUENCE	A(1)	1699			
MD5_CTX	DS(193)	1681D	1689M	1690M	1691M
		1692M	1699	1706	1710
		1712			
CONTEXT	A(160)	1690	1706		
DIGEST	A(16)	1692	1710	1712	
KEY	A(16)	1689			
STATE	A(1)	1691	1699		

Riferimenti dell'indicatore

Indicatore

Riferimenti (D=Defin. M=Modific.)

* * * * * F I N E R I F E R I M E N T I I N C R O C I A T I * * * * *

R i f e r i m e n t i e s t e r n i

Procedure di cui si } effettuato il bind in modo statico:

Procedura	Riferimenti			
iconv	1234			
_CIPHER	1251	1381	1654	1710
MD4Init_c	1082			
MD4Only_c	1112			
MD4Final_c	1103			
MD4Update_c	1092			
iconv_close	1246			
NLMR4_isBit	833			
NLMR4_bitor	860			
QtqIconvOpen	1213			
NLMR4_bitand	842	1512	1521	
NLMR4_byteor	869	1518	1571	
NLMR4_random	919			
NLMR4_NTOWFv1	787			
NLMR4_NTOWFv2	796			
NLMR4_toAscii	809			
NLMR4_byteand	851	1502	1516	1556
	1557	1558	1559	1560
	1561	1562	1573	
NLMR4_toUpper	928			
NLMR4_toLower	936			
Qc3EncryptData	1304			
NLMR4_uint16LE	878			
NLMR4_uint32LE	886			
NLMR4_uint64LE	894			
NLMR4_stringLE	902			
NLMR4_getToken	944			
MD4R4_MD4Init_r	1042			
MD4R4_MD4Only_r	1072	1599		
NLMR4_toUnicode	821			
MD4R4_MD4Final_r	1063			
MD4R4_MD4Update_r	1052			
NLMR4_varstringLE	911			
FREEASCII TRANSCODER	816			
NLMR4_getLMResponse	712			
NLMR4_getNTRResponse	723			
FREEUNICODETRANSCODER	828			
NLMR4_Message_isType1	565			
NLMR4_Message_isType2	576			
NLMR4_Message_isType3	587			
NLMR4_Message_hasFlag	610			
NLMR4_getLMv2Response	748			
NLMR4_computeResponse	775			
NLMR4_Message_newType1	513			
NLMR4_Message_newType3	547			
NLMR4_getNTLM2Response	734			
NLMR4_removeAuthFromUrl	703			
NLMR4_getNTLMv2Response	763			
NLMR4_AuthPlugin_getRealm	83			
NLMR4_NTLM_enableTestMode	485			
NLMR4_Message_getChallenge	598			
NLMR4_Message_encodeBase64	682			
NLMR4_Message_decodeBase64	694			
NLMR4_Message_validateType2	530			

NTLMR4_NTLM_setLMCompatibility	500
NTLMR4_AuthPlugin_setAuthentication	96
NTLMR4_Message_getTargetDNSTreeName	671
NTLMR4_Message_getTargetNBDomainName	635
NTLMR4_AuthPlugin_resetAuthentication	107
NTLMR4_Message_getTargetDNSDomainName	659
NTLMR4_Message_getTargetNBComputerName	623
NTLMR4_Message_getTargetDNSComputerName	647
NTLMR4_AuthPlugin_negotiateAuthentication	148
NTLMR4_AuthPlugin_isAuthenticationRequired	75
NTLMR4_AuthPlugin_mustReceiceAuthErrorPage	132
NTLMR4_AuthPlugin_produceAuthenticationHeader	162
NTLMR4_AuthPlugin_interpretAuthenticationHeader	117

Campi importati:

Campo	Attributi	Definito
Non ci sono riferimenti nell'origine.		

Campi esportati:

Campo	Attributi	Definito
Non ci sono riferimenti nell'origine.		

* * * * * F I N E R I F E R I M E N T I E S T E R N I * * * * *

S o m m a r i o m e s s a g g i o

Id msg	Sv	Numero	Testo messaggio
*RNF7031	00	156	Il nome o l'indicatore non sono referenziati.
*RNF7226	00	1	La struttura dati ha inizializzazioni di sottocampi in sovrapposizione.
*RNF0274	30	13	Direttiva di compilazione sconosciuta; direttiva ignorata.
*RNF5505	30	13	Nelle specifiche di calcolo a formato libero le posizioni 6-7 non sono in bianco.

* * * * * F I N E S O M M A R I O M E S S A G G I O * * * * *

S o m m a r i o f i n a l e

Totali messaggio:

Informativi (00)	:	157
Avvertenze (10)	:	0
Errori (20)	:	0
Errori gravi (30+)	:	26
-----		-----
Totale	:	183

Totali origine

Record	:	1720
Specifiche	:	971
Record dati	:	0
Commenti	:	613

* * * * * F I N E S O M M A R I O F I N A L E * * * * *

La compilazione si } arrestata. Nel programma sono stati trovati errori di gravita{ 30.

* * * * * E N D O F C O M P I L A T I O N * * * * *