Documentation for db2-hash-routines

Helmut K. C. Tessarek $14^{\rm th}~{\rm July},~2016$

db2-hash-routines is a package which provides User Defined Functions and Stored Procedures for $\rm IBM^{\circledR}$ DB2 $^{\circledR}$ to generate and validate hashes.

Contents

1.	db2-hash-routines	1
	1.1. Building the library and registering the UDFs and SPs	1
	1.2. Description of the UDFs and SPs	2
Α.	. UDF and SP reference	3
	A.1. bcrypt	3
	A.2. sha256	4
	A.3. sha512	5
	A.4. php_md5	6
	A.5. apr_md5	7
	A.6. apr_crypt	8
	A.7. apr_sha1	9
	A.8. apr_sha256	10
	A.9. validate_pw	11

1. db2-hash-routines

1.1. Building the library and registering the UDFs and SPs

Login as the instance user and run the script

Linux and AIX ./makertn
Win32 makertn.bat

The makertn script detects the DB2 instance directory and locates apr-1-config and apu-1-config automatically. If for some reason the script cannot set either one of the necessary variables, they have to be set manually. Uncomment and change the following variables in the makertn script.

DB2PATH=

APRPATH=

APUPATH=

Set DB2PATH to the directory where DB2 is accessed. This is usually the instance home directory.

Set APRPATH to where apr-1-config is located.

Set APUPATH to where apu-1-config is located.

The UDFs and SPs are written in ANSI C and should compile on all platforms.

The only requirements are APR and APR-util. You can get APR and APR-util at http://apr.apache.org/

To register the UDFs and SPs, connect to your database and run the script:

db2 -tvf register.ddl

1.2. Description of the UDFs and SPs

This library delivers the following routines¹:

bcrypt sha256 sha512 php_md5 apr_md5 apr_crypt apr_sha1 apr_sha256 validate_pw

The php_md5 routine is compatible to the PHP md5 function.

The apr_md5, apr_crypt, apr_sha1 and bcrypt routines are compatible to the functions used in Apache's htpasswd utility.

The apr_sha256 routine returns the identifier {SHA256} plus the base64 encoded sha256 hash.

The sha256 and sha512 functions return glib2's crypt hashes (if supported).

validate_pw can be used to validate a password against a hash.

On systems with glibc2, the validate_pw routine will also validate hashes of the form \$id\$salt\$encrypted. The following values of id are supported:

ID	Method
1	MD5
2a	Blowfish (not in mainline glibc; added in some Linux distributions)
5	SHA-256 (since glibc 2.7)
6	SHA-512 (since glibc 2.7)

Note: In win32 environments apr_crypt returns the output of bcrypt, if available. If bcrypt is not available, the output of apr_md5 is returned.

¹see Appendix A for a reference of the UDFs and SPs

A. UDF and SP reference

A.1. bcrypt

bcrypt algorithm. The bcrypt routine is compatible to the function used in Apache's htpasswd utility.

The argument can be a character string that is either a CHAR or VARCHAR not exceeding 4096 bytes.

The result of the function is CHAR(60). The result can be null; if the argument is null, the result is the null value.

A.2. sha256

```
>>-SHA256--(--expression--)------><
>>-SHA256--(--expression--,--hash--)------><
```

SHA256 algorithm. The sha256 routine returns a glibc2's crypt hash. If the system's crypt does not support sha-256, an SQLSTATE 39702 is returned.

The argument can be a character string that is either a CHAR or VARCHAR not exceeding 4096 bytes.

The result of the function is CHAR(55). The result can be null; if the argument is null, the result is the null value.

A.3. sha512

```
>>-SHA512--(--expression--)------><
>>-SHA512--(--expression--,--hash--)------><
```

SHA512 algorithm. The sha512 routine returns a glibc2's crypt hash. If the system's crypt does not support sha-512, an SQLSTATE 39702 is returned.

The argument can be a character string that is either a CHAR or VARCHAR not exceeding 4096 bytes.

The result of the function is CHAR(98). The result can be null; if the argument is null, the result is the null value.

Examples:

Return Status = 0

A.4. php_md5

```
>>-PHP_MD5--(--expression--)------><
>>-PHP_MD5--(--expression--,--hash--)------><
```

MD5 hash. The php_md5 routine is compatible to the PHP md5 function.

The argument can be a character string that is either a CHAR or VARCHAR not exceeding 4096 bytes.

The result of the function is CHAR(32). The result can be null; if the argument is null, the result is the null value.

A.5. apr_md5

```
>>-APR_MD5--(--expression--)------><
>>-APR_MD5--(--expression--,--hash--)------><
```

Seeded MD5 hash. The apr_md5 routine is compatible to the function used in Apache's htpasswd utility.

The argument can be a character string that is either a CHAR or VARCHAR not exceeding 4096 bytes.

The result of the function is CHAR(37). The result can be null; if the argument is null, the result is the null value.

A.6. apr_crypt

Unix crypt. The apr_crypt routine is compatible to the function used in Apache's htpasswd utility.

The argument can be a character string that is either a CHAR or VARCHAR not exceeding 4096 bytes.

The result of the function is CHAR(13). The result can be null; if the argument is null, the result is the null value.

A.7. apr_sha1

SHA1 algorithm. The apr_sha1 routine is compatible to the function used in Apache's htpasswd utility.

The argument can be a character string that is either a CHAR or VARCHAR not exceeding 4096 bytes.

The result of the function is CHAR(33). The result can be null; if the argument is null, the result is the null value.

A.8. apr_sha256

SHA256 algorithm. The apr_sha256 routine returns the identifier {SHA256} plus the base64 encoded sha256 hash.

The argument can be a character string that is either a CHAR or VARCHAR not exceeding 4096 bytes.

The result of the function is CHAR(52). The result can be null; if the argument is null, the result is the null value.

A.9. validate_pw

This routine can be used to validate a password against a hash.

The two input arguments can be character strings that are either a CHAR or VARCHAR not exceeding 4096 bytes (password) and 120 bytes (hash). The second parameter (hash) must not be empty, otherwise an SQLSTATE 39701 is returned.

The result of the routine is an INTEGER. If the password is valid, 1 is returned. If the password is not valid, 0 is returned. The result can be null; if the argument is null, the result is the null value.

```
1)
   SELECT validate_pw('testpwd', 'cqs7u0vz8KBlk') FROM SYSIBM.SYSDUMMY1"
   1
   _____
             1
     1 record(s) selected.
2)
   CALL validate_pw('testpwd', 'cqs7u0vz8KBlk', ?)
     Value of output parameters
     _____
     Parameter Name : IS_VALID
     Parameter Value : 1
     Return Status = 0
3)
   CALL validate_pw('testpwd', '0123456789abcdef', ?)
     Value of output parameters
     Parameter Name : IS_VALID
```

Parameter Value : 0

Return Status = 0

 $\hbox{ Date: } 2016\text{-}07\text{-}14\ 21\text{:}47\text{:}14\ \text{-}0400 \qquad \hbox{Id: a} 7\text{c}178 \\ \hbox{dfd1afd7e5338a04ca36f5d04642faeda2}$