Introduction
The following is a database version of the “Hello World” program using Empress Database with Encryption option. The objective of the program is to show in the most simple and direct way that the Empress Database with Encryption does two things. First, it stores and prints the phrase “Hello World”. Second, it does indeed encrypt the phrase “Hello World” by examining the database file with the Linux utility “od” (octal dump).

In this example, the Linux system is Ubuntu Linux and the encryption algorithms are part of the “libgcrypt” library. This “libgcrypt” library is a standard part of most 2.6 Linux systems such as Red Hat Enterprise Linux, Novell Linux, Ubuntu, etc, and implements encryption algorithms such as AES (Advanced Encryption Standard) using key sizes of 128, 192 or 256 bits.

Preparing the Encryption Environment and Database
Log onto a Linux system that has the Empress Database with encryption option installed. Type in “pwd” to find out what directory you are in. In this case it is “/home/alex”.

alex@knopit:~$ pwd
/home/alex

Before making a new database called “testdb” in “/home/alex”, use the Empress environment variable “MSCIPHERKEYINFO” to associate the new database name “testdb” with a short hexadecimal encryption key “74657374696e6731”. You can choose the most appropriate database name and encryption key.

alex@knopit:~$ MSCIPHERKEYINFO="/home/alex/testdb":74657374696e6731
alex@knopit:~$ export MSCIPHERKEYINFO
alex@knopit:~$ echo $MSCIPHERKEYINFO
/home/alex/testdb:74657374696e6731

There are a number of different more secure ways, other than using environment variables, to associate an encryption key with a database. This will be explored in further Technical Notes.
Now create the database "testdb" in "/home/alex" using the “empmkdb” command with the “cipher” option set to “AES256”. Encryption and decryption on this database will be performed using the Advanced Encryption Standard (AES) algorithm with a key size of 256 bits (32 bytes or 64 hexadecimal digits) through the “libgcrypt” library.

For illustrative purposes only, we use a shorter less secure key of 16 hex digits. This is automatically zero padded on the right to make up the full 64 hex digits. “74657374696e6731000000000000000000000000000000000000000000000000”

alex@knopit:~$ empmkdb -cipher AES256 testdb

Start the Database, Create and Examine the “Hello” Table
Start up Empress Interactive SQL using the “empsql” command on the database “testdb”. The Interactive SQL prompt should appear.

alex@knopit:~$ empsql testdb
EMPRESS V8.62
(c) Copyright Empress Software Inc. 1983, 2008
1*

Create a new table called “Hello” with two fields called “Name” and “Message”. “Name” will store 10 characters and “Message” will store 15 characters. The “Message” field is also specified as “encrypted”.

1* create table Hello (Name character (10), Message character (15) encrypted);

Show all properties of the “Hello” table using the “display table” command with the option “all”. Note that the “Hello” table is shown as Table #  6 by the “display table all” command. In Empress, all tables are stored as files named “nnnn.rel”, so the “Hello” table would be a file in the database named “0006.rel”.

2* display table Hello all;
*** Table: Hello ***

Attributes:
Name character(10,1)
Message character(15,1) Encrypted

Creator: alex
Lock Level: RECORD

Table #: 6
Records: 0
Record size: 27
Store and Print “Hello World”
Store a value for “Name” and “Message” into the “Hello” table using the “insert” command. Use “Alex” for “Name” and “Hello World” for “Message”.

3* insert into Hello values ("Alex", "Hello World");

Select all the data from the “hello” table. The “Name”, “Alex”, and the Message “Hello World” is printed on the terminal. This is the Database version of the “Hello World” program with encryption!!

4* select * from Hello;
Name    Message
Alex    Hello World

Verify that the “Hello World” Data is Encrypted
Is the data “Alex” and “Hello World” in the database with “Hello World” encrypted? Verify that the data has been encrypted by using the “od” command.

First exit interactive SQL using “stop”.
5* stop

Then change into the “testdb” directory.
alex@knopit:~$ cd testdb

From the comments about the “display table all” above, the file name for the “Hello” table is “0006.rel”. Use the command “od –c” on file “0006.rel” to see what characters make up the “Hello” table.

alex@knopit:~/testdb$ od -c 0006.rel
0000000 231 227 233 031  \t  \0  \0  \b  \0  \
0000020  \0  \0  \0  \0 020  \0  \0  \0  \0  \0  \
0000040  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \
0000060  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \
0000080  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \
00000a0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \
00000c0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \
00000e0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \
0000100  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \
0000120  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \
0000140  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \
0000160  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \
*    00002000 252   A   l   e   x  \0  \0  \0  \0  \0  \0   | 231 006 005 321
00002020  t 303   P 334 250  1 033 371  _  -  \n00002033

The first part of the output, from 0000000 to 0000160 shows header information for the table. At item “0002000”, the “Name” “Alex” is evident, but “Hello World” is not recognizable. “Hello World” has been stored in the Empress Database just
after “Alex” as encrypted data using the AES256 algorithm from the “libgcrypt” library.

In contrast, by re-running the commands above without the “encrypted” keyword in the “create table Hello” command, you would get the following results:

```
0002000 252   A   l   e   x  \0  \0  \0  \0  \0  \0   H   e   l   l   o 
0002020       W   o   r   l   d  \0  \0  \0  \0
0002032
```

If someone uses Empress to “select * from Hello” without the correct key, they will only get an error message such as “failed to verify cipher key info”.

**Appendix 1: A List of Commands Used in This Technical Note**
The following may be cut and pasted to repeat the example. You will need to change the database path location to suit your computer login. You may want to change the database name and the 16-digit hexadecimal encryption key as well.

```
pwd
MSCIPHERKEYINFO="/home/alex/testdb":74657374696e6731
export MSCIPHERKEYINFO
echo $MSCIPHERKEYINFO
empmkdb -cipher AES256 testdb
empsql testdb
create table Hello (Name character (10), Message character (15) encrypted);
display table Hello all;
insert into Hello values ("Alex", "Hello World");
select * from Hello;
stop
cd testdb
od -c 0006.rel
```

Empress Software Inc.
www.empress.com