28.3.0 Komponente OpenSSL

Die Komponente *gb.openssl* implementiert mit den vier Klassen Cipher, Digest, CipherText und HMac verschiedene Verschlüsselungs- und Hash-Verfahren, die Sie unter Nutzung der OpenSSL-Krypto-Bibliothek (OpenSSL-Bibliothek) zur Kodierung und Dekodierung sowie zur Ermittlung von Hash-Werten zum Beispiel von Texten verwenden können.

28.3.0.1 Klasse Cipher

Die Klasse *Cipher* erschließt die (Block- und Stream-)Chiffre-Algorithmen des OpenSSL-Projekts für Gambas. Informieren Sie sich vor der praktischen Anwendung der Klasse über das OpenSSL-Projekt und zur Theorie von Ver- und Entschlüsselungsalgorithmen.

Die Klasse besitzt nur die eine statische Eigenschaft *Cipher.List* und die statische Methode *Cipher.IsSupported()*.

```
Cipher.List (gb.openssl)
Static Property Read List As String[]
```

Mit der Cipher.IsSupported()-Methode ermitteln Sie über den Algorithmus-Namen, ob ein bestimmter Algorithmus aus der OpenSSL-Bibliothek *auf Ihrem System* zur Verfügung steht oder nicht:

```
Print Cipher.IsSupported("AES-192-CFB")
```

Die Funktion gibt *True* zurück, wenn der genannte Algorithmus existiert. Es ist möglich, dass ein bestimmter Algorithmus auf einem System existiert und auf einem anderen nicht. Dies hängt von der installierten OpenSSL-Bibliothek auf dem System ab. Die Methode *IsSupported()* beachtet die Groß- und Kleinschreibung *nicht*.

Interessant ist die Tatsache, dass Sie die Klasse *Cipher* in Verbindung mit der Klasse '.Cipher.Method' wie ein ReadOnly-Array einsetzen können.

Die virtuelle, statische Klasse Cipher.Method (gb.openssl) besitzt diese zwei Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
IvLength	Integer	Vom Algorithmus erwartete Länge des InitVektors
KeyLength	Integer	Vom Algorithmus erwartete Schlüssel-Länge

Tabelle 28.3.0.1.1 : Übersicht zu den Eigenschaften der Klasse Cipher.Method

Die Klasse verfügt über vier Methoden:

Methode	Rückgabetyp	Beschreibung
Encrypt (Plain As String [, Key As String, InitVector As String])	CipherText	Chiffriert einen gegebenen Klartext mit einem Schlüssel und Initialisierungs-Vektor. Wird ein optionales Argument nicht übergeben, werden zufällig generierte Bytes verwendet.
Decrypt (Cipher As CipherText)	String	Entschlüsselt ein CipherText-Objekt, wie es von Encrypt() zurückgegeben wird. Es wird der Klartext zurückgegeben.
EncryptSalted (Plain As String, Password As String [, Salt As String])	String	Verschlüsselt einen Klartext anhand eines Passworts. Der optionale Salt-Parameter muss ein 8-Byte langer String sein. Wird ein Salt-Argument übergeben, dann wird durch Anfügen von Nullen oder durch Trunkieren eine Länge von 8 erreicht. Sonst wird ein zufälliger Salt-Wert benutzt.
DecryptSalted (Cipher As String, Password As String)	String	Entschlüsselt bei korrektem Passwort einen String, der von EncryptSalted() zurückgegeben wurde.

Tabelle 28.3.0.1.2 : Methoden der Klasse Cipher.Method

28.3.0.2 Klasse CipherText

Die Klasse *CipherText* verfügt über drei Eigenschaften, die nur gelesen werden können:

Eigenschaft	Datentyp	Beschreibung
Cipher	String	Die errechnete Chiffre.
InitVector	String	Der zum Verschlüsseln benutzte Initialisierungs-Vektor.
Key	String	Der verwendete Schlüssel.

Tabelle 28.3.0.2.1 : Eigenschaften der Klasse Cipher

Hinweis:

EncryptSalted() gibt, im Gegensatz zu Encrypt(), einen einzigen String zurück. Das Format dieses Strings ist kompatibel mit der Ausgabe des Konsolen-Programms "openssl":

```
hans@linux:~$ echo -n "Gambas" | openssl aes-128-cfb -k "Schrimp" -S 0001020304050607 | base64
~> U2FsdGVkX18AAQIDBAUGB0ZSV6yXHw==
```

28.3.0.3 Klasse Digest

Die Klasse *CipherText* dient als Speicher-Struktur für das Ergebnis der Encrypt()-Methode. Sie verfügt über drei Eigenschaften, die nur gelesen werden können. Sie sind Parameter des Verschlüsselungsalgorithmus und werden zum Entschlüsseln des Chiffrats benötigt:

```
Digest.List (gb.openssl)
Static Property Read List As String[]
```

Mit der Digest.IsSupported()-Methode können Sie ermitteln, ob ein bestimmter Hash-Algorithmus aus der OpenSSL-Bibliothek auf Ihrem System eingesetzt werden kann oder nicht:

```
Print Digest.IsSupported("SHA256")
```

Die Funktion gibt *True* zurück, wenn der genannte Algorithmus auf dem aktuellen System existiert. Auch diese Funktion beachtet die Groß- und Kleinschreibung *nicht*.

28.3.0.4 Klasse HMac

Die Klasse HMac implementiert hash-basierte 'Message Authentication Codes' (HMAC). Informationen dazu finden Sie u.a. bei Wikipedia unter http://de.wikipedia.org/wiki/Hashfunktion. Sie können die Klasse HMac wie eine (statische) Funktion einsetzen und zwei implementierte Konstanten nutzen:

```
HMac.RipeMD160 (gb.openssl)
Const RipeMD160 As Integer = 117 ' &H75
```

```
HMac.Sha1 (gb.openssl)
Const Sha1 As Integer = 64 ' &H40
```

Verwenden Sie diese Konstanten als Methoden-Parameter im Funktionsaufruf *HMac(konstante)*, damit der Hash-Algorithmus RIPEMD-160 oder SHA1 verwendet wird.