Barista Encryptor Object

Description

The Barista Encryptor Object, available in version 12.0, provides the ability to encrypt and decrypt application information based on an existing encryption record residing in the Barista Configuration Records file (ADS_CONFIGURE). Barista uses a specific encryption algorithm where the resulting data is stored as the HTA of the encrypted value. Using this algorithm allows developers to disregard potential delimiter characters generated with the standard encryption.

Encryptor Object Methods

Method	Comments
setConfiguration(str <cfg>)	Sets the encrypt/decrypt configuration record.
encryptData(str <src>)	Returns the encrypted source data.
encryptData(str <src>, str <cfg>)	Returns the encrypted source data using the specified configuration key.
decryptData(str <src>)	Returns the decrypted source data.
decryptData(str <src>, str <cfg>)	Returns the decrypted source data using the specified configuration key.

- <cfg> - Specifies the encrypt/decrypt configuration record to use for the subsequent actions. Must be a key to a valid record in the Barista Configuration Records file (ADS_CONFIGURE).
- <src> - The source text to encrypt or decrypt

Sample Program

```
use ::sys/prog/bao_encryptor.bbj::Encryptor
declare Encryptor Encryptor!
encryptor! = new Encryptor()

config_id$ = "BAR_CREDIT_CARD"
encryptor!.setConfiguration(config_id$)

src_text$ = "4234567890123456"
enc_text$ = encryptor!.encryptData(src_text$)
print enc_text$

dec_text$ = encryptor!.decryptData(enc_text$)
print dec_text$

input "Press enter to release",*
release
```