

Barista Credit Card Object

Description

The Barista Credit Card Object, available in version 12.0, provides the ability to validate, mask, encrypt and decrypt application credit card information. The card validation is based on the industry standard MOD-10 ([Luhn Algorithm](#)) check digit information found in the sys/config/credit_params.txt file. The encrypt and decrypt functions use the [Barista Encryptor Object](#), and require an existing encryption record residing in the Barista Configuration Records file (ADS_CONFIGURE).

This object works in conjunction with the new Credit Card Element Subtype. The BAR_CREDIT_CARD element is supplied as part of the standard download, and may be used in application files, or as an example for other elements. The Credit Card subtype automatically handles encryption/decryption and masking on all output.

Credit Card Object Methods

Method	Comments
setCardType(str <type>)	Sets the base card type for the validation (optional). <M>astercard <V>isa <A>merican Express <C>arte Blanche/Diners Club <D>iscover <E>nRoute <J>CB
validateCard(str <card>)	Returns the result of the credit card number validation. See error codes below.
getMaskedCardValue(str <card>)	Returns the masked card value. Masking is based on the stbl("+CARD_MASK_CHAR"), or "X" if not found.
encryptData(str <src>, str <cfg>)	Returns the encrypted card number data using the specified configuration key.
decryptData(str <src>, str <cfg>)	Returns the decrypted card number 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).
- <card> - The source credit card data to encrypt or decrypt

Error Codes

PARAM_FILE_NF - Parameter file not found.

PARAM_FILE_FMT - Parameter file incorrectly formatted.

" = Card validated successfully.

CARD_CHARACTERS - Invalid card characters (non-numeric) entered.

CARD_LENGTH - Credit card number entry length is invalid based on card prefix.

CARD_CHECK_DIGIT - Credit card failed MOD-10 check digit validation.

Sample Validation Rules (sys/config/credit_params.txt)

```
Parameter format: <code>;<desc>;<pxfs>;<clens>;<vlen>
<code> - Single character credit card code
<desc> - Card description
<pxfs> - Allowable card prefixes
<clens> - Allowable card number lengths
<vlen> - Card validation value length

M;Mastercard;51,52,53,54,55;16;3
V;Visa;4;13,16;3
A;American Express;34,37;15;4
C;Carte Blanche/Diners Club;300,301,302,303,304,305,36,38;14;3
D;Discover;6011;16;3
E;enRoute;2014,2149;15;3
J;JCB;1800,2131;15;3
```

Sample Program

```
use ::sys/prog/bao_valid_card.bbj::CreditCard
declare CreditCard CCard!
CCard! = new CreditCard()

card$ = "4234567890123456"

CCard!.setCardType("V")
print CCard!.validateCard(card$)
print CCard!.getMaskedCardValue(card$)

config$ = "BAR_CREDIT_CARD"
enc_card$ = CCard!.encryptCard(card$, config$)
print enc_card$

dec_card$ = CCard!.decryptCard(enc_card$, config$)
print dec_card$

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

