

# ifHMAC

## Implemented By

- [roHMAC](#)

## Supported Methods

- [Setup\(digestType as String, key as Object\) as Integer](#)
- [Reinit\(\) as Integer](#)
- [Process\(message as Object\) as Object](#)
- [Update\(partialMesssage as Object\) as Void](#)
- [Final\(\) as Object](#)

## Description of Methods

### Setup(digestType as String, key as Object) as Integer

Initialize new HMAC context. The `digestType` parameter selects one of the supported digest algorithms, as documented in [roEVPDigest](#). The `key` parameter must be an `roByteArray` containing the key for the MAC. Returns 0 on success, -1 on failure.

### Reinit() as Integer

Re-initialize an existing HMAC context. This can be called to reuse an existing `roHMAC` object to authenticate new data. Returns 0 on success or non-zero on failure

### Process(message as Object) as Object

The parameter should be an `roByteArray`. The data in the array is digested and an MAC is generated. Returns an `roByteArray` containing the MAC.

```
mac = hmac.Process(message)
```

is equivalent to

```
hmac.Reinit()  
hmac.Update(message)  
mac = hmac.Final()
```

### Update(partialMesssage as Object) as Void

Add more data to be digested. The parameter should be an `roByteArray`. The data in the array is added to the current digest.

### Final() as Object

Return an `roByteArray` containing the final MAC.

