|  |
| --- |
| GTS Integrations |
| Mulesoft - Password Encryption |
| Implementation/Management Steps |
|  |
| **Hemant Chopra** |
| **5/25/2018** |

|  |
| --- |
| This document contains the steps required to implement/ manage encrypted passwords in Mulesoft. |

Table of Contents

[1 Implementation 0](#_Toc521916124)

[1.1 Mule Runtime - Server Level Changes 0](#_Toc521916125)

[1.1.1 Include Mule Encryption key in wrapper.conf 0](#_Toc521916126)

[1.2 Anypoint Studio 0](#_Toc521916127)

[1.2.1 Install Mule Security Module version 1.6.9 0](#_Toc521916128)

[1.3 Create Secure Property File 0](#_Toc521916129)

[1.3.1 Create mulecustom.properties, as shown in below image. 0](#_Toc521916130)

[1.3.2 Open file using Mule Properties Editor 0](#_Toc521916131)

[1.3.3 Use the circled options to update property file 0](#_Toc521916132)

[1.3.4 Click “+” sign to add new property. Enter information and click “Encrypt”. 1](#_Toc521916133)

[1.3.5 Enter encryption information, if not already entered. Select algorithm as “AES” and enter key as “CUSTOMPROPERTIES”. 1](#_Toc521916134)

[1.3.6 Clicking OK on above image will encrypt the password as shown in below image. 2](#_Toc521916135)

[1.3.7 Newly added entry will appear as below. 2](#_Toc521916136)

[1.4 Domain Package 3](#_Toc521916137)

[1.4.1 Create Spring Bean for Secure Property Placeholder 3](#_Toc521916138)

[1.4.2 Copy Jar files to resources folder and include them in build path 3](#_Toc521916139)

[1.5 Implementation Package (Example – Generic Support Utilities) 3](#_Toc521916140)

[1.5.1 Global.xml 3](#_Toc521916141)

[1.5.2 Read Password loaded from encrypted file as a property 3](#_Toc521916142)

# Implementation

## Mule Runtime - Server Level Changes

### Include Mule Encryption key in wrapper.conf

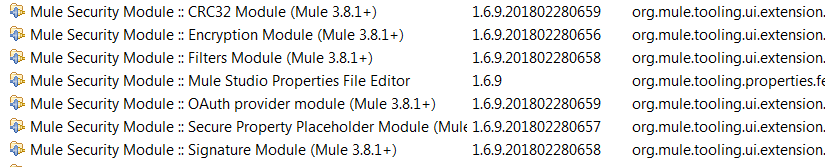
Example:

wrapper.java.additional.20=-DMEKey=CUSTOMPROPERTIES

## Anypoint Studio

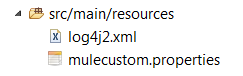
### Install Mule Security Module version 1.6.9

http://anypoint-enterprise-security-update-site.s3.amazonaws.com/1.6.9

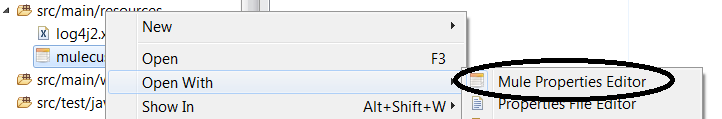


## Create Secure Property File

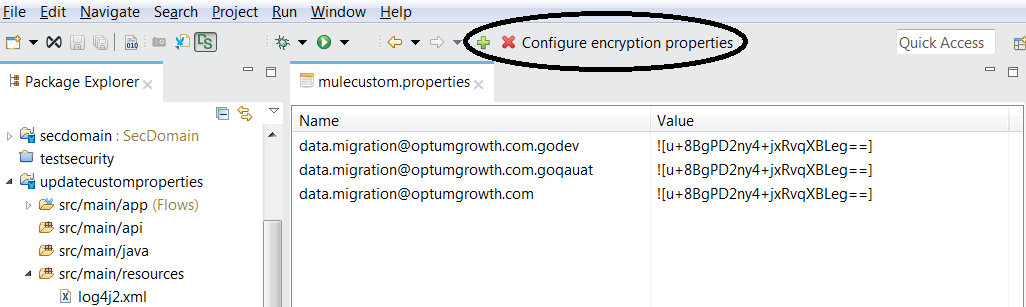
### Create mulecustom.properties, as shown in below image.



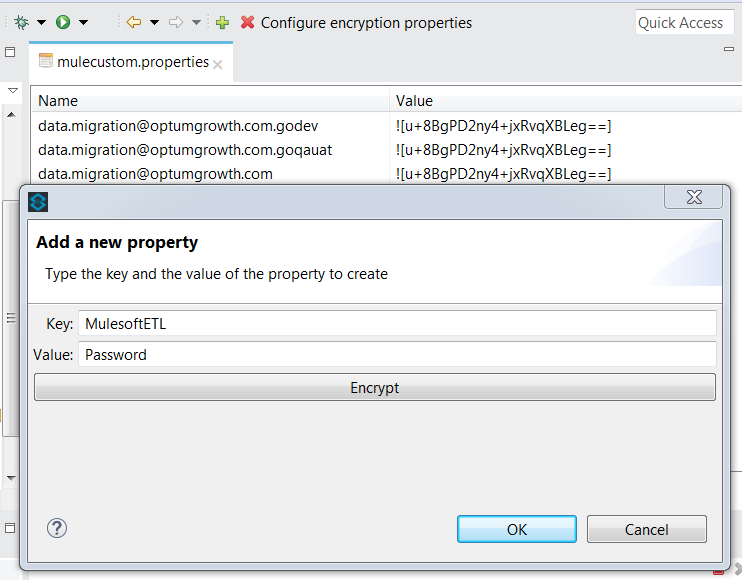
### Open file using Mule Properties Editor



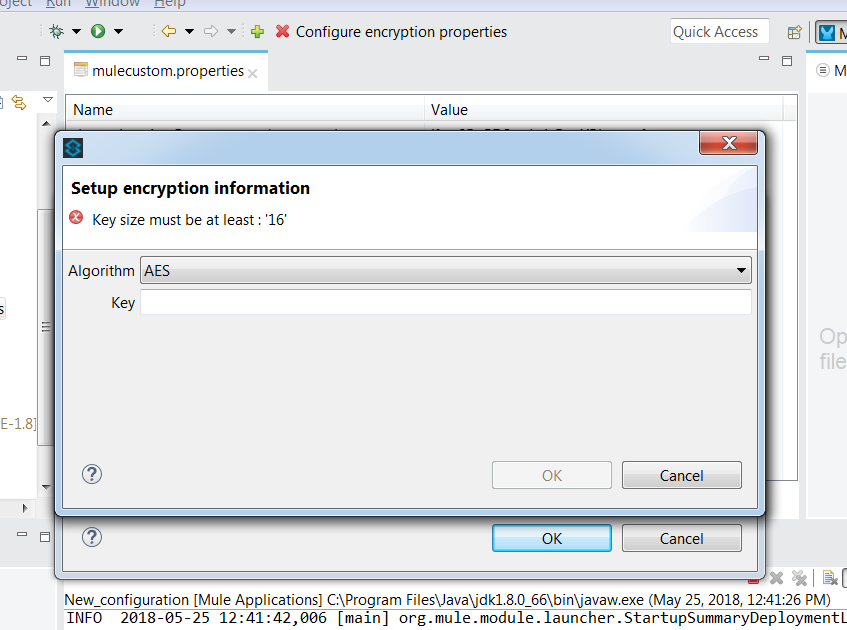
### Use the circled options to update property file



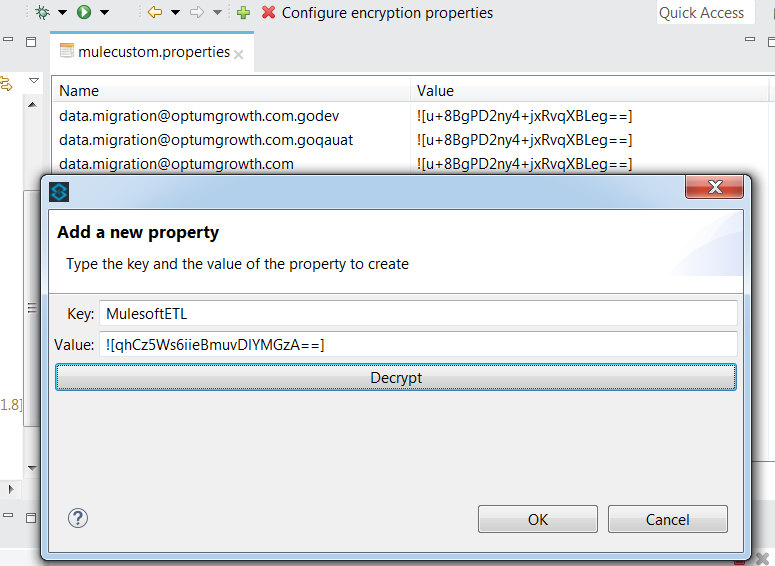
### Click “+” sign to add new property. Enter information and click “Encrypt”.



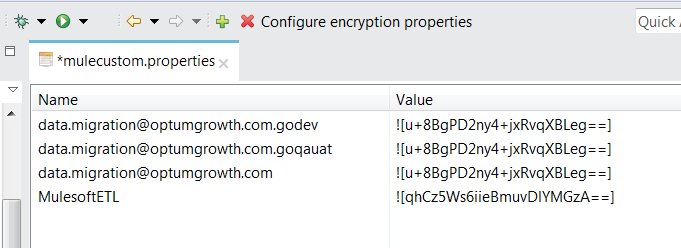
### Enter encryption information, if not already entered. Select algorithm as “AES” and enter key as “CUSTOMPROPERTIES”.



### Clicking OK on above image will encrypt the password as shown in below image.



### Newly added entry will appear as below.



Update the property file as required and place it at the required folder location on server.

## Domain Package

### Create Spring Bean for Secure Property Placeholder

<spring:bean

class=*"org.mule.modules.security.placeholder.SecurePropertyPlaceholderModule"*>

<spring:property name=*"encryptionMode"* value=*"CBC"* />

<spring:property name=*"encryptionAlgorithm"* value=*"AES"* />

<spring:property name=*"location"* value=*"*file://///apsep0406/IPS/Salesforce/PropertyStore/mulecustom.properties*"* />

<spring:property name=*"ignoreUnresolvablePlaceholders"*

value=*"false"* />

<spring:property name=*"ignoreResourceNotFound"* value=*"false"* />

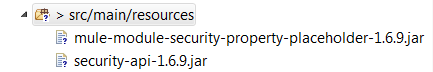
<spring:property name=*"key"* value=*"${MEKey}"* />

</spring:bean>

If Applicable, add ignore-unresolvable in “context:property-placeholder”.

<context:property-placeholder location=*"${mule.env}Domain.properties"* ignore-unresolvable=*"true"*/>

### Copy Jar files to resources folder and include them in build path



## Implementation Package (Example – Generic Support Utilities)

### Global.xml

Add **ignore-unresolvable=*"****true****"***, as mentioned below.

<context:property-placeholder location=*"classpath:envproperties/${mule.env}.properties"* ignore-unresolvable=*"true"*/>

### Read Password loaded from encrypted file as a property

password="#[dw(&quot;p('${GOsalesforce.username}')&quot;)]"