PGP Encryption Utilities

 The PGP Encryption Utilities Plugin can be used to encrypt and sign Files and also to decrypt and verify the same files.It has the following smart services.Encryption can be done in combination with signing in any order.The receiver has to perform the decryption and verification in the same order as encryption and signing.

# Encrypt File With PGP

The Smart Service encrypts the input file with the receiver's PGP public key and saves the encrypted file into a folder.

## Parameters

| **Sl No** | **Name** | **Type** | **Description** |
| --- | --- | --- | --- |
| 1 | Public Key | Document (Required) | The receiver Public Key file to encrypt the input file. |
| 2 | File to encrypt | Document (Required) | Name for the input Document to be encrypted. |
| 3 | Name of the encrypted file | Text (Required) | Name of the new Document to be created. |
| 4 | Extension of the encrypted file | Text (Optional) | Extension in which new file to be created. If this is left empty or null, the default extension of the generated document will be ‘pgp’. |
| 5 | Folder to save the encrypted file into | Folder (Required) | Folder to which the output Document to be saved in. |
| 6 | Create new document | Boolean (Optional) | Provide whether to create a new Appian document to store the generated file or not. This parameter defaults to true. If provided with false as a value, the smart service expects the Existing Document parameter to have a valid Appian document as a value. This document will be overwritten with the contents of the generated file. |
| 7 | Existing Document | Document (Optional) | Provide the document which will be overwritten with the generated file if Create new document is set to ‘false’. If Create new document is set to false and Existing Document is not provided, then an error is thrown in the Error Message output parameter. |

## Outputs

* Encrypted file : (Document)
	+ The output encrypted file.
* Error Message : ( Text )
	+ Error message received.
* Error : ( Boolean )
	+ Set to true on the occurrence of error.

# Decrypt File With PGP

The Smart Service decrypts the input file with the receiver's PGP private key and saves the encrypted file into a folder.

## Parameters

| **Sl No** | **Name** | **Type** | **Description** |
| --- | --- | --- | --- |
| 1 | Private Key | Document (Required) | The receiver Private Key file to decrypt the input file. |
| 2 | File to encrypt | Document (Required) | Name for the input Document to be decrypted. |
| 3 | Name of the decrypted file | Text (Required) | Name of the new Document to be created. |
| 4 | PGP Secure Credentials Store | Text (Required) | Secure Credentials Store containing the password for the private key in the attribute 'password’. |
| 5 | Folder to save the decrypted file into | Folder (Required) | Folder to which the output Document to be saved in. |
| 6 | Create new document | Boolean (Optional) | Provide whether to create a new Appian document to store the generated file or not. This parameter defaults to true. If provided with false as a value, the smart service expects the Existing Document parameter to have a valid Appian document as a value. This document will be overwritten with the contents of the generated file. |
| 7 | Existing Document | Document (Optional) | Provide the document which will be overwritten with the generated file if Create new document is set to ‘false’. If Create new document is set to false and Existing Document is not provided, then an error is thrown in the Error Message output parameter. |

## Outputs

* Decrypted file : (Document)
	+ The output decrypted file.
* Error Message : ( Text )
	+ Error message received.
* Error : ( Boolean )
	+ Set to true on the occurrence of error.

# Sign File With PGP

The Smart Service Signs the Document with the sender PGP private key and writes the signed file as a document

## Parameters

| **Sl No** | **Name** | **Type** | **Description** |
| --- | --- | --- | --- |
| 1 | Private Key | Document (Required) | The receiver Private Key file to decrypt the input file. |
| 2 | File to sign | Document (Required) | Name for the input Document to be signed. |
| 3 | Name of the signed file | Text (Required) | Name of the new Document to be created. |
| 4 | PGP Secure Credentials Store | Text (Required) | Secure Credentials Store containing the password for the private key in the attribute ‘clientpassword’. |
| 5 | Folder to save the signed file into | Folder (Required) | Folder to which the output Document to be saved in. |
| 6 | Create new document | Boolean (Optional) | Provide whether to create a new Appian document to store the generated file or not. This parameter defaults to true. If provided with false as a value, the smart service expects the Existing Document parameter to have a valid Appian document as a value. This document will be overwritten with the contents of the generated file. |
| 7 | Existing Document | Document (Optional) | Provide the document which will be overwritten with the generated file if Create new document is set to ‘false’. If Create new document is set to false and Existing Document is not provided, then an error is thrown in the Error Message output parameter. |
| 8 | Extension of the signed file | Text (Optional) | Extension in which new file to be created. If this is left empty or null, the default extension of the generated document will be ‘pgp’. |

## Outputs

* Signed file : (Document)
	+ The output signed file.
* Error Message : ( Text )
	+ Error message received.
* Error : ( Boolean )
	+ Set to true on the occurrence of error.

# Verify File With PGP

The Smart Service Verifies the Signed file for a valid sender signature using the sender PGP public key and writes the verified file as a document

## Parameters

| **Sl No** | **Name** | **Type** | **Description** |
| --- | --- | --- | --- |
| 1 | Public Key | Document (Required) | The receiver Public Key file to verify the input file. |
| 2 | File to verify | Document (Required) | Name for the input Document to be verified. |
| 3 | Name of the verified file | Text (Required) | Name of the new Document to be created. |
| 4 | Extension of the verified file | Text (Optional) | Extension in which new file to be created. If this is left empty or null, the default extension of the generated document will be ‘pgp’. |
| 5 | Folder to save the verified file into | Folder (Required) | Folder to which the output Document to be saved in. |
| 6 | Create new document | Boolean (Optional) | Provide whether to create a new Appian document to store the generated file or not. This parameter defaults to true. If provided with false as a value, the smart service expects the Existing Document parameter to have a valid Appian document as a value. This document will be overwritten with the contents of the generated file. |
| 7 | Existing Document | Document (Optional) | Provide the document which will be overwritten with the generated file if Create new document is set to ‘false’. If Create new document is set to false and Existing Document is not provided, then an error is thrown in the Error Message output parameter. |

## Outputs

* Verified file : (Document)
	+ The output Verified file (after removing signature digest).
* Error Message : ( Text )
	+ Error message received.
* Error : ( Boolean )
	+ Set to true on the occurrence of error.
* Is Verified: ( Boolean )
	+ Set to true on the valid signature