

# **PGP SupportPac for IBM Integration Bus v9**

## **Part-4: PGP Command-Line Tool (pgpkeytool) User Manual**

**By**

**Dipak Kumar Pal  
([dipakpal.opentech@gmail.com](mailto:dipakpal.opentech@gmail.com))**

## Summary

This article is the fourth in a multi-part series of articles describing PGP security implementation in IBM Integration Bus v9. This series of articles introduces an industry standard solution to Data Security in IBM Integration Bus, enforcing data confidentiality and integrity by implementing **PGP** cryptographic solution. This solution is developed as a custom pluggable feature (or **SupportPac**) of IBM Integration Bus v9. This article introduces a Java based command-line tool (**pgpkeytool**) for PGP key pair generation and key/repository management, an integral part of the end-to-end PGP security implementation in IBM Integration Bus.

## Introduction

PGP SupportPac (v1.0.0.1) for IBM Integration Bus (v9) ships with a Java based command-line tool (**pgpkeytool**) for PGP key pair generation and key/repository management. You do not need any third-party (open source or commercial) tool for PGP key management. Refer to the first article (Part-1) of this series to know how to use this command-line tool to generate and manage PGP keys and key-repositories.

## Supported Operations

Table-1 describes list of operations supported by this command-line tool.

**Table-1: List of operations supported by *pgpkeytool***

S/N	Operation	Description
1	<b>generatePGPKeyPair</b>	Generate PGP Private/Public key pair.
2	<b>changePrivateKeyPassphrase</b>	Change Private key passphrase.
3	<b>importPrivateKey</b>	Import specified Private key into Private key Repository file.
4	<b>importPublicKey</b>	Import specified Public key into Public key Repository file.
5	<b>exportPrivateKey</b>	Export specified Private key from Private key Repository file into separate Private key file.
6	<b>exportPublicKey</b>	Export specified Public key from Public key Repository file into separate Public key file.
7	<b>deletePrivateKey</b>	Delete specified Private key from Private key Repository file.
8	<b>deletePublicKey</b>	Delete specified Public key from Public key Repository file.
9	<b>listPrivateKeys</b>	List all Private keys hosted by Private key Repository file.
10	<b>listPublicKeys</b>	List all Public keys hosted by Public key Repository file.
11	<b>encrypt</b>	Encrypt file.
12	<b>signAndEncrypt</b>	Sign and encrypt file.

13	<b>decrypt</b>	Decrypt encrypted file and validate signature.
----	----------------	--

**Command to display list of supported operations.**

**java pgpkeytool -help**

**Figure-1: Help command to display list of supported operations.**

```

cmd.exe
C:\PGP\pgpkeytool>java pgpkeytool -help
Operation not supported: -help

Supported Operations:
generatePGPKeyPair:      Generate PGP key pair.
changePrivateKeyPassphrase: Change passphrase for specified private key.
encrypt:                 PGP Encryption.
signAndEncrypt:          PGP Encryption with Signature.
decrypt:                 PGP Decryption with Signature validation.
importPrivateKey:         Import specified Private key into Private key Repository file.
importPublicKey:          Import specified Public key into Public key Repository file.
exportPrivateKey:         Export specified Private key from Private key Repository file into separate Private key file.
exportPublicKey:          Export specified Public key from Public key Repository file into separate Public key file.
deletePrivateKey:         Delete specified Private key from Private key Repository file.
deletePublicKey:          Delete specified Public key from Public key Repository file.
listPrivateKeys:          List all Private keys in Private key Repository file.
listPublicKeys:           List all Public keys in Public key Repository file.

For help, execute:
Help example:            java pgpkeytool [operation] -help
                          java pgpkeytool generatePGPKeyPair -help
C:\PGP\pgpkeytool>_

```

## Operation details

To display command/operation details, execute following command.

**java pgpkeytool <operation> -help**

**Operation: *generatePGPKeyPair***

**Description:** This operation generates PGP Private/Public key pair. Following snap-shot describes various option details including examples.

**Figure-2: pgpkeytool help for *generatePGPKeyPair* command**

```

cmd.exe
C:\PGP\pgpkeytool>java pgpkeytool generatePGPKeyPair -help
Usage:
java pgpkeytool generatePGPKeyPair -sa SignatureAlgorithm -pa PublicKeyAlgorithm -i identity -a asciiArmor -kr idel keysize
-c cipher -s privateKeyFile -o publicKeyFile

Example:
java pgpkeytool generatePGPKeyPair -sa DSA -pa RSA -i "IBM <ibm-pgp-keys@in.ibm.com>" -a true -kr 1024 -kd 1024 -c AES_256 -s
C:/PGP/KeyRepository/SecretKey.asc -o C:/PGP/KeyRepository/PublicKey.asc

Example (All default options)
java pgpkeytool generatePGPKeyPair -i "IBM <ibm-pgp-keys@in.ibm.com>" -s C:/PGP/KeyRepository/SecretKey.asc -o C:/PGP/KeyRepository/PublicKey.asc

Options:
-sa SignatureAlgorithm: <Optional>      Supported Signature Algorithms: RSA, DSA. Default: RSA
-pa PublicKeyAlgorithm: <Optional>      Supported PublicKey Algorithms: RSA, ELG. Default: RSA
-i identity:                            Key Identity <Key User Id> e.g. "IBM <ibm-pgp-keys@in.ibm.com>"
-a asciiArmor:                          ASCII encoding [true|false]. Default: true
-kr idel keysize:                        Key size. kr - RSA Key Size, kd - DSA Key Size, ke - EL GAMAL Key Size. Default: 1024
-bit:                                   Key size.
-c cipher:                              Supported Cipher Algorithms: IDEA, TRIPLE_DES, CAST5, BLOWFISH, DES, AES_128, AES_192
AES_256, TWOFISH. Default: CAST5
-s privateKeyFile:                      Private Key File Name <Absolute path> to export the private key.
-o publicKeyFile:                       Public Key File Name <Absolute path> to export the public key.
C:\PGP\pgpkeytool>_

```

**Operation:** *changePrivateKeyPassphrase, importPrivateKey, importPublicKey, exportPrivateKey, exportPublicKey, deletePrivateKey, deletePublicKey, listPrivateKeys, listPublicKeys*

**Description:** These are various operations on key repository files. Following snap-shot describes various option details for each operation including examples.

**Figure-3: pgpkeytool help for commands on key repository management**



```

C:\PGP\pgpkeytool>java pgpkeytool importPrivateKey -help
Usage:
java pgpkeytool importPrivateKey -sr privateKeyRepositoryFile -i asciiArmor -sf privateKeyFile
java pgpkeytool importPublicKey -pr publicKeyRepositoryFile -i asciiArmor -pf publicKeyFile
java pgpkeytool exportPrivateKey -sr privateKeyRepositoryFile -su privateKeyUserId -i asciiArmor -sf privateKeyFile
java pgpkeytool exportPublicKey -pr publicKeyRepositoryFile -pu publicKeyUserId -i asciiArmor -pf publicKeyFile
java pgpkeytool changePrivateKeyPassphrase -sr privateKeyRepositoryFile -su privateKeyUserId
java pgpkeytool deletePublicKey -pr publicKeyRepositoryFile -pu publicKeyUserId
java pgpkeytool deletePrivateKey -sr privateKeyRepositoryFile -su privateKeyUserId
java pgpkeytool listPrivateKeys -sr privateKeyRepositoryFile
java pgpkeytool listPublicKeys -pr publicKeyRepositoryFile

Supported Operations on PGP Key Repositories:
changePrivateKeyPassphrase: Change passphrase for specified private key.
importPrivateKey: Import specified Private key into Private key Repository file.
importPublicKey: Import specified Public key into Public key Repository file.
exportPrivateKey: Export specified Private key from Private key Repository file into separate Private key file.
exportPublicKey: Export specified Public key from Public key Repository file into separate Public key file.
deletePrivateKey: Delete specified Private key from Private key Repository file.
deletePublicKey: Delete specified Public key from Public key Repository file.
listPrivateKeys: List all Private keys in Private key Repository file.
listPublicKeys: List all Public keys in Public key Repository file.

Options:
-sr privateKeyRepositoryFile : PrivateKey Repository File <Absolute Path>.
-pr publicKeyRepositoryFile : PublicKey Repository File <Absolute Path>.
-sf privateKeyFile : PrivateKey File <Absolute Path>.
-pf publicKeyFile : PublicKey File <Absolute Path>.
-su privateKeyUserId : PrivateKey User Id
-pu publicKeyUserId : PublicKey User Id
-i asciiArmor [true|false] : Whether Key file is Ascii armored. <Optional> Default: true

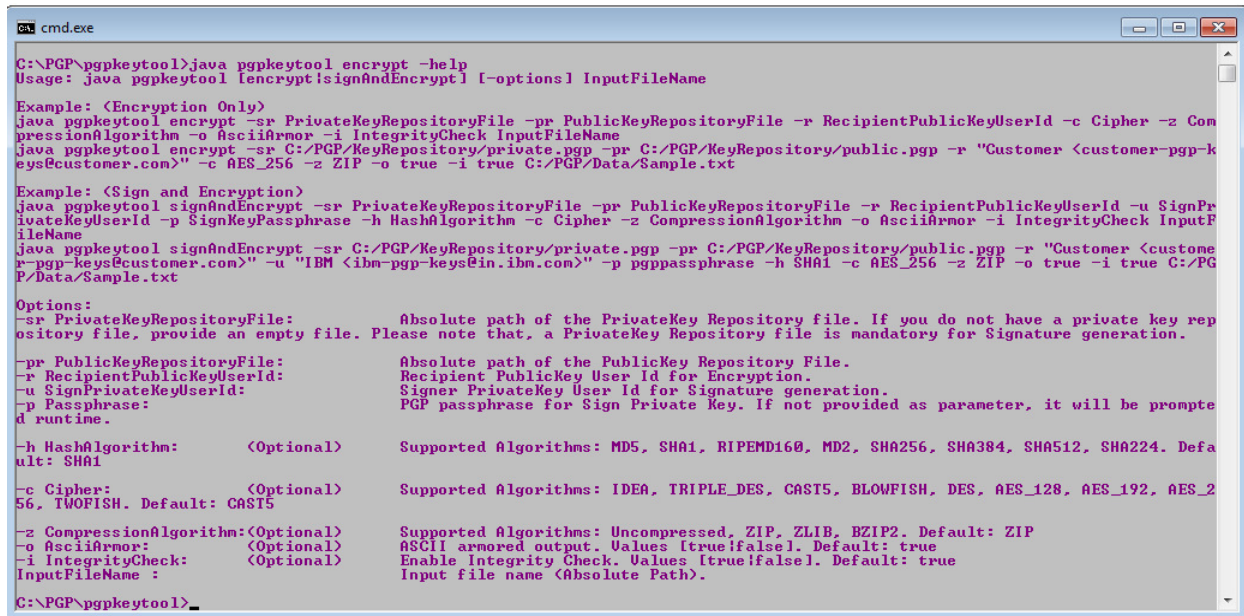
Examples:
java pgpkeytool importPrivateKey -sr C:/PGP/KeyRepository/private.pgp -i true -sf C:/PGP/KeyRepository/SecretKey.asc
java pgpkeytool importPublicKey -pr C:/PGP/KeyRepository/public.pgp -i true -pf C:/PGP/KeyRepository/PublicKey.asc
java pgpkeytool exportPrivateKey -sr C:/PGP/KeyRepository/private.pgp -su "IBM <ibm-pgp-keys@in.ibm.com>" -i true -sf C:/PGP/KeyRepository/SecretKeyExported.asc
java pgpkeytool exportPublicKey -pr C:/PGP/KeyRepository/public.pgp -pu "IBM <ibm-pgp-keys@in.ibm.com>" -i true -pf C:/PGP/KeyRepository/PublicKeyExported.asc
java pgpkeytool changePrivateKeyPassphrase -sr C:/PGP/KeyRepository/private.pgp -su "IBM <ibm-pgp-keys@in.ibm.com>"
java pgpkeytool deletePublicKey -pr C:/PGP/KeyRepository/public.pgp -pu "IBM <ibm-pgp-keys@in.ibm.com>"
java pgpkeytool deletePrivateKey -sr C:/PGP/KeyRepository/private.pgp -su "IBM <ibm-pgp-keys@in.ibm.com>"
java pgpkeytool listPrivateKeys -sr C:/PGP/KeyRepository/private.pgp
java pgpkeytool listPublicKeys -pr C:/PGP/KeyRepository/public.pgp

C:\PGP\pgpkeytool>

```

**Operation:** *encrypt, signAndEncrypt*

**Description:** These operations sign and encrypt file.

**Figure-4: pgpkeytool help for *encrypt* and *signAndEncrypt* command**


```

cmd.exe
C:\PGP\pgpkeytool>java pgpkeytool encrypt -help
Usage: java pgpkeytool [encrypt|signAndEncrypt] [-options] InputFileName

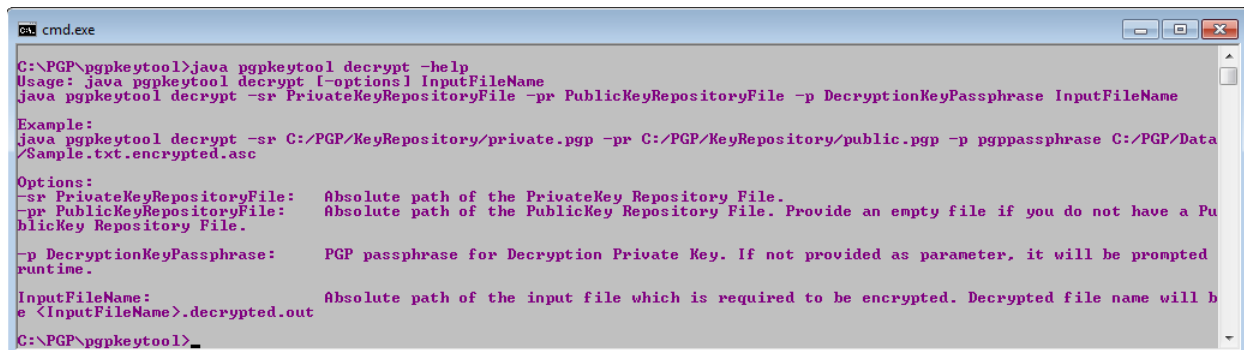
Example: (Encryption Only)
java pgpkeytool encrypt -sr PrivateKeyRepositoryFile -pr PublicKeyRepositoryFile -r RecipientPublicKeyUserId -c Cipher -z CompressionAlgorithm -o AsciiArmor -i IntegrityCheck InputFileName
java pgpkeytool encrypt -sr C:/PGP/KeyRepository/private.pgp -pr C:/PGP/KeyRepository/public.pgp -r "Customer <customer-pgp-keys@customer.com>" -c AES_256 -z ZIP -o true -i true C:/PGP/Data/Sample.txt

Example: (Sign and Encryption)
java pgpkeytool signAndEncrypt -sr PrivateKeyRepositoryFile -pr PublicKeyRepositoryFile -r RecipientPublicKeyUserId -u SignPrivateKeyUserId -p SignKeyPassphrase -h HashAlgorithm -c Cipher -z CompressionAlgorithm -o AsciiArmor -i IntegrityCheck InputFileName
java pgpkeytool signAndEncrypt -sr C:/PGP/KeyRepository/private.pgp -pr C:/PGP/KeyRepository/public.pgp -r "Customer <customer-pgp-keys@customer.com>" -u "IBM <ibm-pgp-keys@in.ibm.com>" -p pgppassphrase -h SHA1 -c AES_256 -z ZIP -o true -i true C:/PGP/Data/Sample.txt

Options:
-sr PrivateKeyRepositoryFile: Absolute path of the PrivateKey Repository file. If you do not have a private key repository file, provide an empty file. Please note that, a PrivateKey Repository file is mandatory for Signature generation.
-pr PublicKeyRepositoryFile: Absolute path of the PublicKey Repository File.
-r RecipientPublicKeyUserId: Recipient PublicKey User Id for Encryption.
-u SignPrivateKeyUserId: Signer PrivateKey User Id for Signature generation.
-p Passphrase: PGP passphrase for Sign Private Key. If not provided as parameter, it will be prompted at runtime.
-h HashAlgorithm: (Optional) Supported Algorithms: MD5, SHA1, RIPEMD160, MD2, SHA256, SHA384, SHA512, SHA224. Default: SHA1
-c Cipher: (Optional) Supported Algorithms: IDEA, TRIPLE_DES, CAST5, BLOWFISH, DES, AES_128, AES_192, AES_256, TWOFISH. Default: CAST5
-z CompressionAlgorithm: (Optional) Supported Algorithms: Uncompressed, ZIP, ZLIB, BZIP2. Default: ZIP
-o AsciiArmor: (Optional) ASCII armored output. Values {true|false}. Default: true
-i IntegrityCheck: (Optional) Enable Integrity Check. Values {true|false}. Default: true
InputFileName: Input file name (Absolute Path).

C:\PGP\pgpkeytool>

```

**Operation: *decrypt*****Description:** This operation decrypts and validate signature.**Figure-5: pgpkeytool help for *decrypt* command**


```

cmd.exe
C:\PGP\pgpkeytool>java pgpkeytool decrypt -help
Usage: java pgpkeytool decrypt [-options] InputFileName
java pgpkeytool decrypt -sr PrivateKeyRepositoryFile -pr PublicKeyRepositoryFile -p DecryptionKeyPassphrase InputFileName

Example:
java pgpkeytool decrypt -sr C:/PGP/KeyRepository/private.pgp -pr C:/PGP/KeyRepository/public.pgp -p pgppassphrase C:/PGP/Data/Sample.txt.encrypted.asc

Options:
-sr PrivateKeyRepositoryFile: Absolute path of the PrivateKey Repository File.
-pr PublicKeyRepositoryFile: Absolute path of the PublicKey Repository File. Provide an empty file if you do not have a PrivateKey Repository File.
-p DecryptionKeyPassphrase: PGP passphrase for Decryption Private Key. If not provided as parameter, it will be prompted at runtime.
InputFileName: Absolute path of the input file which is required to be encrypted. Decrypted file name will be <InputFileName>.decrypted.out

C:\PGP\pgpkeytool>

```

**Installation and Configuration**

Following environment variables are used while describing various directory paths. Make sure you use correct directory paths as per your operating system. For this example IBM JRE is used, however you can use ORACLE JRE also.

**PGPKEYTOOL\_HOME:** pgpkeytool home directory. (Examples: **C:\PGP\pgpkeytool** in Windows, **/var/pgp/pgpkeytool** in UNIX)

**JRE\_HOME:** Java Runtime home. (Example: **C:\IBM\MQSI\v9\jre17** in Windows, **/opt/ibm/mqsi/v9/jre17** in UNIX)

### Step 1: Installation.

Download **PGP SupportPac v1.0.0.1.zip** file from GitHub repository (<https://github.com/dipakpal/MyOpenTech-PGP-SupportPac/tree/master/binary/IIBv9>), unzip and copy following jar files to **PGPKEYTOOL\_HOME** directory.

**bcpjg-jdk15on-149.jar**  
**bcprov-ext-jdk15on-149.jar**  
**com.ibm.broker.supportpac.PGP.jar**

### Step 2: Update JRE with unrestricted JCE policy jar files.

In comply with the United States of America export restrictions, IBM's SDKs (JREs) ship with strong but limited jurisdiction policy files. Unlimited jurisdiction policy files can be obtained from the IBM site (<https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=jcesdk>). The ZIP file should be unpacked and the two JAR files placed in the IBM JRE's **jre/lib/security/** directory.

To work with strong encryption and larger key size, replace following two jar files in **JRE\_HOME/ lib/security** with unrestricted JCE policy jar files obtained from IBM site.

**local\_policy.jar**  
**US\_export\_policy.jar**

### Step 3: Environment setup.

Setup command line environment before executing various pgpkeytool supported operations.

#### Windows:

```
SET PGPKEYTOOL_HOME=C:\PGP\pgpkeytool
SET JRE_HOME=C:\IBM\MQSI\v9\jre17
SET PATH=%JRE_HOME%\bin;%PATH%
SET
CLASSPATH=%PGPKEYTOOL_HOME%\com.ibm.broker.supportpac.PGP.jar;%CLASSPATH%
H%
```

#### UNIX:

```
export PGPKEYTOOL_HOME=/var/pgp/pgpkeytool
export JRE_HOME=/opt/ibm/mqsi/v9/jre17
export PATH=$JRE_HOME/bin:$PATH
export
CLASSPATH=$PGPKEYTOOL_HOME/com.ibm.broker.supportpac.PGP.jar:$CLASSPATH
```

### Step 4: Execute any command supported by pgpkeytool command-line tool.

Following table describes a sample command (Windows) to generate a PGP private/public pair with DSA as Signature key algorithm (key size: 1024 bits) and El



Gamal (ELG) as Encryption algorithm (key size: 2048 bits). Keys generated by the command are exported into respective files specified at command.

```
java pgpkeytool generatePGPKeyPair -sa DSA -pa ELG -i "IBM <ibm-pgp-keys@in.ibm.com>" -a true -
ke 2048 -kd 1024 -c AES_256 -s C:/PGP/KeyRepository/SecretKey.asc -o
C:/PGP/KeyRepository/PublicKey.asc
```

```
C:\>java pgpkeytool generatePGPKeyPair -sa DSA -pa ELG -i "IBM <ibm-pgp-keys@in.ibm.com>" -a
true -ke 2048 -kd 1024 -c AES_25
6 -s C:/PGP/KeyRepository/SecretKey.asc -o C:/PGP/KeyRepository/PublicKey.asc
Please enter PGP Passphrase:
passphrase
Please Re-enter PGP Passphrase:
passphrase
PGP Signature Key Algorithms: DSA
PGP PublicKey Algorithms: ELG
Identity: IBM <ibm-pgp-keys@in.ibm.com>
PassPhrase: passphrase
AsciiArmor: true
Keysize (DSA): 1024
Keysize (ELG): 2048
Cipher Algorithm: AES_256
SecretKeyFile: C:/PGP/KeyRepository/SecretKey.asc
PublicKeyFile: C:/PGP/KeyRepository/PublicKey.asc
Generating a prime number >= 2048 bits
Prime:
927225713254366876108267478739990596564292808862376347820292311290170823893178918
4603573445019947368156898300098848325
770098616643236642564036341045551732790563999271224080669671343081648887881006958
80440394688948693049994378896703535453408953
431906853666381595650151040740113973746932504719864596144557611884768517519727046
90385105526290436790638733846602902225043927
063475707214768674508742462513100627361159336976179648290655873159262394169601718
94873658340743410637970140128420734336830218
584287368716243109497677072588193817913607981151754130754576102801397550233481447
16143260953368788430281658845383589100283121
11215237
***** PGP Private Key *****
```

-----BEGIN PGP PRIVATE KEY BLOCK-----

Version: BCPG v1.46

```
IQHpBFI7b+YRBADV2YSw3jA2bEm5N7C+omj8EYE82kGVn+Diw3WsQL7thDYXn2r5
aJe9ooGr9N/pjE9A381ZjsYOIL5Sk/3rGrCfEktJ/JdnFOjw8DVvl1lIFiBgSwMH
Bu+1rkbKzALAUlJDoXxUk1R8bc1w3UnHvrtcnNKiksEaAbpON8p23bij3wCg0L1E
vrOxIPcUj1zTEd99Zd0a1TsD/0ejN8KRI6T8TmcXq0afFBIzo97h19PQ4dC8aG9i
vNXBLzKeSvT0UaGWzs+6a/N5n5OWeSpV1ssWkaEuXNGKhLQER9pk8fYUeaxe9O/F
TyWTGVqXgOSc2rbWskzXLT1WfQfjHXC8ep94ffjXNr6oQGBLtk7omXAevYL0bRhF
lic4BAC806THURufLXHfjEh3qq9MZkbE+7iiDXRrINpVpBVfGKRe+1iu2ylhO4fR
Xo8HyqThzrFqCztjXogwVFK7NsqRKvNn43r5nNP/RnOeqqrYt+Ls1Nve0QZaQw
6oqICePoz2HNbdg1IA+sb/cy93KQkPMrRfTW848jQbC6ezixFv4JAwIaPlesMqEs
oGBRzH27rEGYyF1J7zyVCEHkeYUAXVPInuurI9cLzzBZcy0TVN5HipVg6t3myyL
tdYGe8pk0t3TW/vxtB1JQk0gPGlibS1wZ3Ata2V5c0Bpbi5pYm0uY29tPohGBBMR
AgAGBQJSO2/mAAoJEMpxq54Ye/B/QEMAOmrlQguSJgJ0v0a5a3dGbqB5Q0HeAJOS
2QND0e7e9S8RpoNjD0Q0G7DqbEZ0ETwRSO2/mEAggqwPdZNCtjNxDGyxvBdwX
E48dHvCQeBMhOR6ACh+eDJOUj0SRjSHALduVLcC4+g1+V4OstVxXV9/xcauBGnnN
KpchetNLn0AkHDq2xjEpstBYzPPFchbManOlObKe3/6U+79tU53kMFDCXHfkCefO
```

```
kyroOvcjTJ0tZi2IIRsdpvtJw2oLSfMpSbhNWQIPzTI0kj5Q+sl9D6CsnQQiFJHh
wr24fQMnKbV9fOhGcbv2z0svdo7edJ5Dqnxl2afzqO7Iu7WbrZVSGqtoH4xCE4XA
VZdaD75mUaM/8MU6D2kgwkc82+8sDBuL2jjpvPJxYtFCfPLeDFCW+qdnNUVLPhUH
8oUIH0PDj/F70MY2MxDfOSV78pN0AD2r3HQjDBooaQSTVcnXLYFhTyZKkNOMzSo9
0umwqssTKiJJ++sYvZHwi9CvkrFBIGIO/y/lfElJtFyKcGX6L20Cez9tKTyTznY
4/XV3bO7iUrqqkJA2goT8SfbGKbAiAfYztMfG7/szXDqKoJ1WQtoNyjXbpiZ6rRv
f9ssSk1i0JxyypvsMhqERN/VTfhmSojE/BIihyomASWst7bg/XmJQpnWpUcumzRx
ZwX1pVeOGES2IR/UoVMd9TL2LunJJcbZgG251UxT2fnjH3ofz5UTSgMKHTVAzvJ0
5qCw7ykbqT4YviHh1CNz9mzXx3fd1v76CB9iH16A37FhocYAY8x85zRS9QZDZeb9
ME02i1GhSck/dzwvn92hIXAJu5oL9IhwhkRU4reGXu6y/kiGYisXGcylMJxRrsjZ
q9N/IPgq3aKuJ7q4MPQH9qE+cKuYspGdEFwIqIqWHbBvx/OSPD3ssn8DIUvSwBYy
izrWAq7+iQkWMvWVjWcJ5fDGhWwy0WtxAKbe1E3H7c33ckb/sA6Jkqfc+CGe7Fec
8ZiUqgxK1+Xx2YiXj21pgrN0Go02QWZDZszmfjHOUmhj2Jhk8m9NJ1gPQK8gFZL3
MNe0lbo1Q1mk1tRL/836pof4JBWmfq7jLXBnQhrnTdiMOCb2I8wKXBOI9eqV9/4J
AwIJConD+1MX5mDv5ZOmowJnUL1Fs9ODoDzzI5I39Vgu3h9oUpZ20pnCOWpzBANq
jsNfHGSjrpzuox22AeJIKHxQYT0yEL5ojSnsY+ulbqMZOCvf4YIuBNwg2yWrP1Qv
2bgruI2V1RAW8Xw6Sdz6LpUFB/lD6te/EyvbI8O9hY6Yergabv6bEva424FJIYn
YrPeUbKp8BcazLxPG8X2cZDFKaQZm/V/l8w5tz7g/12fAbSH44/1VW+Y5PpvhBLJ
GxX+YxzgrZDRq4bqDu5n+mPYgRuB3pKOfnZpeDNqll3EFTJQjGT1LTCPX9aLLQFf
1XFcTmptybod/vfdU3WkuABMRjkY3zWbOSLD+hS/8CW2imim9G5Xj3+nQPoLIS7QN
Y5fhd0x5kS97HM6qMVbeD0fJaiuniEYEGBECAAYFAI7b+cACgkQynGrnhh78H8+
EACglIu41fqWt3Nmuxwl2qE9nGAhii8AoKagIE+p1pLQr+PFjvGgvCQOfZNB
=ULTN
-----END PGP PRIVATE KEY BLOCK-----
```

\*\*\*\*\* PGP Public Key \*\*\*\*\*

-----BEGIN PGP PUBLIC KEY BLOCK-----

Version: BCPG v1.46

```
mQGIBFI7b+YRBADV2YSw3jA2bEm5N7C+omj8EYE82kGVn+Diw3WsQL7thDYXn2r5
aJe9ooGr9N/pjE9A381ZjsYOIL5Sk/3rGrCfEkt/JdnFOjw8DVv1lIIFiBgSwMH
Bu+1rkbKzALAUJDoXxUk1R8bc1w3UnHvrtcnNKiksEaAbpON8p23bij3wCg0L1E
vrOxIPcUj1zTEd99Zd0a1TsD/0ejN8KR16T8TmcXq0affBIZo97h19PQ4dC8aG9i
vNXBLzKeSvT0UaGWzs+6a/N5n5OWeSpV1ssWkaEuXNGKhLQER9pk8fYUeaeE9O/F
TyWTGVqXgOSC2rbWskZLT1WfQfjHXC8ep94ffJXNr6oQGBLtk7omXAevYL0bRhF
lic4BAC806THURufLXHfjEh3qq9MZkbE+7iidxRrINpVpBVfGKRe+1iu2ylhO4fR
Xo8HyqThzrFqFczjtXogwVFK7NsqrKvNn43r5nNP/RnOeqqrYt+Ls1Nve0QZaQw
6oqICePoz2HNbdg1IA+sb/cy93KQkPMrRfTW848jQbC6ezixFrQdSUJNIDxpYmOt
cGdwLWtleXNAaW4uaWJtLmNvbT6IRgQTEQIABgUCUjtv5gAKCRDKcaueGHvwf0BD
AKDK5UILLkiYCdL9GuWt3Rm6geUNB3gCdEtKdQ6O3vUvEaaDYw9ENBuw6mxG5AxxE
Ujtv5hAIIKsD3WTXE4zcSHRssQZLIW3cFxOPHR7wkHgTiaEegAofngyTII9EkY0h
wC3bIS3AuPoNfleDrLvCv1ff8XGrgRp5zSqXIXrTS59AJBw6tsYxKbLQWMzzxXIW
zGpzpaGynt/+IPu/bVOD5DBQwLx35AnnzpMq6Dr3I0ydLWSNiJUbHab7ScNqC0nz
KUM4TVkCD80yNJI+UPrJfQ+grJOEhSR4cK9uH0DJym1fXzoRgm79s9LL3aO3nSe
Q6p8Zdmn86juylu1m62VUhqraB+MQhOFwFWXWg++ZIGjP/DFOG9pIMJHPNvvLawb
i9o46bzycWLRQn6S3gxQlvqnZzVFSz4VB/KFCB9Dw4/xe9DGNjMQ3zkle/KTdaA9
q9x0IwwaKGkEk1XJ1y2BYU8mSpDTjM0qPdLpsKrLEyoiSfvrGL2R8IvQr5KxQSBi
Dv8v5XxJsbRcinBl+i9tAns/bSk8k2Z4WOP11d2zu4IK6pJCQNoKE/En2ximwIgh
2M7THxu/7M1w6iqCdVklLaDco126Ymeq0b3/bLEpNYtCcscqb7DIahETf1U34ZkqI
xPwSIocqJgElrLe24P15iUKZ1qVHLps0cWcF9aVXjhhLNIef1KFTHFUy9i7pySxG
2YBtdvMU9n54x96H8+VE0oDCh01QM7ydOagsO8pG6k+GL4h4dQjc/Zs18d33db+
+ggfYh9egN+xcaHGAGPMFOc0UvUGQ2Xm/TBNNotRoUnJP3c8L5/doSFwCbuaC/SI
cIZEVOK3hl7usv5IhmIrFxnMpTCcUa7I2avTfyD4Kt2irie6uDD0B/ahPnCrmLKR
```



```
nRBcCKiKlh2wb8fzkjw97LJ/A5VL0sAWMos61gKu/okJfJL1IY1nCeXwxoVsMtFr
cQCm3tRNx+3N93JG/7AOiZKn3PghnuxXnPGYIKoMStfl8dmIl49taYKzdBqNNkFm
Q2bM5n4xzlJoY9iYZPJvTSdYD0CvIBWS9zDXtJW6NUNZpNbUS//N+qaH+CQVpn6u
4y1wZ0Ia503YjDgm9iPMClwTiPXqlfeIRgQYEQIABgUCUjtv5wAKCRDKcaueGHvw
fz4QAKCUI7jV+pa3c2a7HCXaoT2cYCGKLwCgpqAgT6nWktCv48WO8aC8JA59k0E=
=++mI
-----END PGP PUBLIC KEY BLOCK-----
```

C:\>

## Conclusion

This series of articles provides an industry standard solution that mitigates a huge gap in IBM Integration Bus Data Security zone, where this article primarily introduces a Java based command-line tool known as **pgpkeytool**, an integral part of the end-to-end PGP solution in IBM Integration Bus. This command-line tool (**pgpkeytool**) is responsible for PGP key and repository management, however you can use any other open source or commercial tools for key pair generation. Future version of **pgpkeytool** will be enhanced with user-friendly GUI similar to IBM Key Management tool shipped with Websphere MQ.

You can post any query regarding to this PGP SupportPac at following IBM DeveloperWorks public community forum, author of this article will address those queries.

### [PGP SupportPac for IBM Integration Bus](https://www.ibm.com/developerworks/community/groups/community/pgpsupportpaciib)

<https://www.ibm.com/developerworks/community/groups/community/pgpsupportpaciib>

## References

- **PGP Basics**
  - **PGP Basics: PGP basic concepts** (<http://www.pgpi.org/doc/pgpintro/>)
  - **Bouncy Castle: Bouncy Castle Resources** (<http://www.bouncycastle.org/>)
  - **Gpg4Win: PGP encryption/decryption command line and GUI tool** (<http://www.gpg4win.org/index.html>)
  - **Portable PGP: Java based GUI tool for PGP** (<http://ppgp.sourceforge.net/>)
  - **GnuPG: GnuPG PGP library** (<http://www.gnupg.org/>)
  - **GitHub: Samples and other Artifacts** (<https://github.com/dipakpal/MyOpenTech-PGP-SupportPac>)
- **Public Community at IBM DeveloperWorks**
  - **PGP SupportPac for IBM Integration Bus:**  
<https://www.ibm.com/developerworks/community/groups/community/pgpsupportpaciib>
- **IBM Integration Bus resources**

- [IBM Integration Bus product page](#)  
Product descriptions, product news, training information, support information, and more.
- [IBM Integration Bus V7 information center](#)  
A single Web portal to all IBM Integration Bus V6 documentation, with conceptual, task, and reference information on installing, configuring, and using your IBM Integration Bus environment
- [Download free trial version of IBM Integration Bus](#)  
IBM Integration Bus is an ESB built for universal connectivity and transformation in heterogeneous IT environments. It distributes information and data generated by business events in real time to people, applications, and devices throughout your extended enterprise and beyond.
- [IBM Integration Bus documentation library](#)  
IBM Integration Bus specifications and manuals.
- [IBM Integration Bus forum](#)  
Get answers to technical questions and share your expertise with other IBM Integration Bus users.
- [IBM Integration Bus support page](#)  
A searchable database of support problems and their solutions, plus downloads, fixes, and problem tracking.
- **WebSphere resources**
  - [developerWorks WebSphere](#)  
Technical information and resources for developers who use WebSphere products. developerWorks WebSphere provides product downloads, how-to information, support resources, and a free technical library of more than 2000 technical articles, tutorials, best practices, IBM Redbooks, and online product manuals. Whether you're a beginner, an expert, or somewhere in between, you'll find what you need to build enterprise-scale solutions using the open-standards-based WebSphere software platform.
  - [developerWorks WebSphere application integration developer resources](#)  
How-to articles, downloads, tutorials, education, product info, and other resources to help you build WebSphere application integration and business integration solutions.
  - [Most popular WebSphere trial downloads](#)  
No-charge trial downloads for key WebSphere products.
  - [WebSphere forums](#)  
Product-specific forums where you can get answers to your technical questions and share your expertise with other WebSphere users.
  - [WebSphere demos](#)  
Download and watch these self-running demos, and learn how WebSphere products can provide business advantage for your company.
  - [WebSphere-related articles on developerWorks](#)  
Over 3000 edited and categorized articles on WebSphere and related technologies by top practitioners and consultants inside and outside IBM. Search for what you need.
  - [developerWorks WebSphere weekly newsletter](#)  
The developerWorks newsletter gives you the latest articles and information only on those topics that interest you. In addition to WebSphere, you can

select from Java, Linux, Open source, Rational, SOA, Web services, and other topics. Subscribe now and design your custom mailing.

- [WebSphere-related books from IBM Press](#)  
Convenient online ordering through Barnes & Noble.
- [WebSphere-related events](#)  
Conferences, trade shows, Webcasts, and other events around the world of interest to WebSphere developers.