Encryption Facility for z/OS
Version 1.Release 2

Using Encryption Facility for OpenPGP

SA23-2230-60

Note

Before using this information and the product it supports, read the information in "Notices" on page 191.

This edition applies to Version 1 Release 2 of IBM Encryption Facility for z/OS (5655-P97) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2024-06-24

Contents

Figures

Tables

About this document

The document contains information about using Encryption Facility for OpenPGP that is part of the product feature IBM® Encryption Services included in IBM Encryption Facility for z/OS (5655-P97).

Encryption Facility for z/OS® support for OpenPGP allows you to encrypt and decrypt messages and data files that comply with OpenPGP standards.

This document provides you with the information to use Encryption Facility for OpenPGP.

Who should read this document

Anyone who plans, installs, customizes, administers, and uses Encryption Facility for OpenPGP should use this document. It should also be used by those who install, configure, or provide support for Encryption Facility in the following areas:

- Integrated Cryptographic Services Facility (ICSF)
- Resource Access Control Facility

This product assumes that you have experience installing, configuring, and using z/OS, ICSF, and RACF®. It also assumes that you understand Java™–related concepts and tasks.

How to use this document

The major topics are:

- Chapter 1, "Overview of IBM Encryption Facility for OpenPGP," on page 1 presents an overview of Encryption Facility for OpenPGP, its functions, and hardware and software requirements.
- Chapter 2, "Getting started," on page 15 presents information about installation and getting started with Encryption Facility for OpenPGP.
- Chapter 3, "Using Encryption Facility for OpenPGP," on page 19 presents information on how to use Encryption Facility for OpenPGP for encryption, decryption, and authentication.
- Chapter 4, "Encryption Facility for OpenPGP commands," on page 23 presents information on Encryption Facility for OpenPGP commands.
- Chapter 5, "Encryption Facility for OpenPGP messages," on page 83 presents information on Encryption Facility for OpenPGP messages.
- Chapter 6, "JCL, command examples, and reference," on page 179 presents user scenarios for Encryption Facility for OpenPGP.

Where to find more information

Where necessary, this document references information in other documents. For complete titles and order numbers for all elements of z/OS, see *z/OS Information Roadmap*.

Related publications

The Encryption Facility library contains the following documents:

- *IBM Encryption Facility for z/OS: Licensed Program Specifications*
- *IBM Encryption Facility for z/OS: Program Directory*
- *IBM Encryption Facility for z/OS: Planning and Customizing*
- *IBM Encryption Facility for z/OS: Using Encryption Facility for OpenPGP*

Documentation for Cryptographic Coprocessors is found on the web at CryptoCards (www.ibm.com/security/cryptocards)

Other sources of information

IBM provides customer-accessible discussion areas where PKI Services and RACF may be discussed by customer and IBM participants. The following resources are available through the Internet to provide additional information about PKI Services, RACF, and many other security-related topics:

- **PKI Services home page**

  To visit the PKI Services home page, see PKI Services for z/OS (www.ibm.com/systems/z/os/zos/features/pki). Check this site for updates regarding PKI Services.

- **Techdocs**

  To visit the Techdocs - Technical Sales Library home page, see IBM Techdocs (www.ibm.com/support/techdocs/atsmastr.nsf/Web/TechDocs). Use the search keyword “crypto” to help narrow your search.

- **RACF home page**

  You can visit the RACF home page at RACF download page (github.com/IBM/IBM-Z-zOS/tree/master/zOS-RACF/Downloads).

- **RACF-L discussion list**

  Customers and IBM participants may also discuss RACF on the RACF-L discussion list. RACF-L is not operated or sponsored by IBM; it is run by the University of Georgia.

  To subscribe to the RACF-L discussion and receive postings, send a note to:

  ```
  listserv@listserv.uga.edu
  ```

  Include the following line in the body of the note, substituting your first name and last name as indicated:

  ```
  subscribe racf-l first_name last_name
  ```

  To post a question or response to RACF-L, send a note, including an appropriate Subject: line, to:

  ```
  racf-l@listserv.uga.edu
  ```

- **RACF sample code**

You can get sample code, internally-developed tools, and exits to help you use RACF. This code works in our environment, at the time we make it available, but is not officially supported. Each tool or sample has a README file that describes the tool or sample and any restrictions on its use.

To access this code from a Web browser, go to the RACF home page and select the 'Downloads' topic from the navigation bar, or go to ftp://ftp.software.ibm.com/eserver/zseries/zos/racf/.

The code is also available from `ftp.software.ibm.com` through anonymous FTP. To get access:

1. Log in as user anonymous.
2. Change the directory, as follows, to find the subdirectories that contain the sample code or tool you want to download:

   ```
   cd eserver/zseries/zos/racf/
   ```

An announcement will be posted on RACF-L discussion list and on newsgroup *ibm.servers.mvs.racf* whenever something is added.

Note: Some Web browsers and some FTP clients (especially those using a graphical interface) might have problems using `ftp.software.ibm.com` because of inconsistencies in the way they implement the FTP protocols. If you have problems, you can try the following:

- Try to get access by using a Web browser and the links from the RACF home page.
- Use a different FTP client. If necessary, use a client that is based on command line interfaces instead of graphical interfaces.

- If your FTP client has configuration parameters for the type of remote system, configure it as UNIX instead of MVS™.

Restrictions

Because the sample code and tools are not officially supported,

- There are no guaranteed enhancements.
- No APARs can be accepted.

How to provide feedback to IBM

We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information. For more information, see How to send feedback to IBM.

Summary of changes

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line to the left of the change.

Note: IBM z/OS policy for the integration of service information into the z/OS product documentation library is documented on the z/OS Internet Library under IBM z/OS Product Documentation Update Policy (www.ibm.com/docs/en/zos/latest?topic=zos-product-documentation-update-policy).

Changes made in IBM Encryption Facility for z/OS Version 1 Release 2

This release of *IBM Encryption Facility for z/OS: Using Encryption Facility for OpenPGP* contains maintenance information for Version 1.2. This topic summarizes the changes made to this document. This topic does not summarize the changes made to the product.

This document contains information previously presented in SA23-2230-40. This latest information supports IBM z/OS Version 3 Release 1 and earlier.

Changed

The following content is changed.

June 2024 release

- Updated the following for APAR OA66324:
  - "Encryption Facility considerations when changing Java release levels" on page 10.
  - "Supported key sizes" on page 10.
  - "RNG_JCE_PROVIDER" on page 32.
- Updated "-t — Treat input as text" on page 71.

Chapter 1. Overview of IBM Encryption Facility for OpenPGP

This topic presents an overview of Encryption Facility for OpenPGP, its functions, and hardware and software requirements.

What is Encryption Facility for OpenPGP?

Encryption Facility for OpenPGP is part of the IBM Encryption Facility for z/OS product feature Encryption Services. Encryption Facility for OpenPGP provides encryption and decryption of messages and data files in accordance with the OpenPGP standards.

For complete information about IBM Encryption Facility for z/OS, including the features that are available, see *IBM Encryption Facility for z/OS: Planning and Customizing*.

What is OpenPGP?

OpenPGP is an Internet draft standard protocol for ensuring the confidentiality and integrity of data that can be exchanged between trusted partners. It defines the following requirements and suggested practices:

- Public key and passphrase-based encryption to ensure confidentiality of the data.
- Digital signatures for partner authentication and to help ensure that a transferred message has not been altered in transit (data integrity) and that the message has been sent by the party claiming to have sent it (non-repudiation).
- OpenPGP certificates for the exchange of key information that can provide the data integrity service.

For a definition of the open standards for OpenPGP, see RFC 4880 (tools.ietf.org/html/rfc4880).

What does Encryption Facility for OpenPGP do?

The OpenPGP internet draft standard protocol defines a syntax for packaging data into packets, where each packet provides the context for a data integrity service like encryption or decryption. Encryption Facility for OpenPGP implements all of the required services as described in the internet draft standard protocol for OpenPGP and specifically provides the following services:

- Public key-based encryption
- Passphrase-based encryption (PBE)
- Modification detection of encrypted data
- Compression of packaged data
- Importing and exporting of OpenPGP certificates in binary or ASCII "armorized" formats
- Digital signatures of data

Encryption Facility for OpenPGP is also able to make use of X.509 certificates for public key infrastructure (PKI) to extend the basis of trust for OpenPGP environments.

With Encryption Facility for OpenPGP, you can apply many of these services to the same data to form an OpenPGP message that you can exchange with other OpenPGP-compliant applications. A software application must support the OpenPGP message format to be compatible with Encryption Facility and other OpenPGP-compliant products. Encryption Facility for OpenPGP also can leverage the existing security facilities of z/OS to provide a secure and scalable OpenPGP client. For example, with Encryption Facility for OpenPGP you can do the following tasks:

- Use as input or output UNIX Systems Services files or z/OS partitioned data sets (PDS and PDSE), or z/OS sequential data sets.

- Perform cryptographic acceleration with certain kinds of IBM Z® hardware.
- Use Security Server Resource Access Control Facility (RACF) and Integrated Cryptographic Services Facility (ICSF) key repositories.

You cannot use the Encryption Services batch program CSDFILDE or the Decryption Client to process Encryption Facility for OpenPGP encrypted data. For complete information about the Encryption Facility for z/OS product and its functions, see *IBM Encryption Facility for z/OS: Planning and Customizing*.

To implement Encryption Facility for OpenPGP services, you must use the IBM Java Development Kit.

Understanding OpenPGP

Encryption Facility for OpenPGP is designed to comply with the OpenPGP standards for encryption, decryption, and other integrity functions.

Understanding session keys and data encryption

Encryption Facility for OpenPGP encrypts data using a randomly-generated session key and a symmetric encryption algorithm (such as TDES or AES). It encrypts the session key and includes it with the encrypted data. The receiving application can decrypt the session key and, in turn, decrypt the data.

Two kinds of session key encryption are available to OpenPGP:

- Public-key encryption, which creates a public-key encrypted session key packet using the public key of the recipient to encrypt the data; only the recipient can decrypt this data with the corresponding private key.
- Passphrase-based encryption (PBE), which creates a symmetric-key encrypted session key packet using a passphrase (like a “password”) to encrypt the data; only this password can be used to decrypt the data.

Encryption Facility for OpenPGP can package an OpenPGP message so that multiple trusted partners can securely exchange data. Encryption Facility for OpenPGP generates one random symmetric session key to encrypt the data to be exchanged. Then, in the case of public-key encryption, it encrypts the session key with the public keys of all the trusted partners; while in the case of PBE, it encrypts the session key with a shared passphrase.

When unpacking an OpenPGP message, Encryption Facility for OpenPGP searches its key respositories for a match to the public key that has been used to encrypt the session key. The OpenPGP standard defines a quick check that allows Encryption Facility for OpenPGP to know if its key can decrypt the packaged data. If this check succeeds, Encryption Facility for OpenPGP decrypts the data, and, if necessary, validates the signature and modification detection code of the data.

Understanding public-key encryption

Public-key encryption makes use of the public-key encrypted session key packet. A public-key encrypted session key packet holds the session key encrypted with a public-key encryption algorithm, such as Rivest-Shamir-Adelman (RSA) or ElGamal. The message itself is encrypted with the session key. A public-key encrypted session key packet contains the key identifier (ID) of the public key that the session key is encrypted with, an identifier of the asymmetric algorithm used to encrypt the session key, and the encrypted session key itself. Unlike the PBE encryption packet, the public-key encryption session key packet must contain a session key with the following information:

- Version number.
- Key identifier (ID). (OpenPGP standards define an algorithm to calculate the key ID of a public key.)
- Algorithm identifier for the asymmetric algorithm to encrypt the session key.
- Encrypted session key data.

Understanding passphrase-based encryption

Passphrase-based encryption (PBE) makes use of the symmetric-key encrypted session key packet. The symmetric-based key encryption session key packet contains the following information:

- Version number
- Algorithm identifier for the symmetric algorithm to encrypt the session key
- A string-to-key (S2K) specification
- Encrypted session key data, which is optional

With the S2K specification OpenPGP standards allow the system to prompt a user for the correct passphrase. When decrypting an OpenPGP message, Encryption Facility for OpenPGP uses the passphrase to decrypt the session key.

How Encryption Facility for OpenPGP works

Encryption Facility for OpenPGP uses the IBM Java SDK and the Java Cryptographic Extension (JCE) providers to implement most of the "primitives" that are described in Internet draft standard protocol for OpenPGP.

Figure 1 on page 3 shows how Encryption Facility for OpenPGP works to encrypt and decrypt data and manage certificates and keys for use with OpenPGP systems:

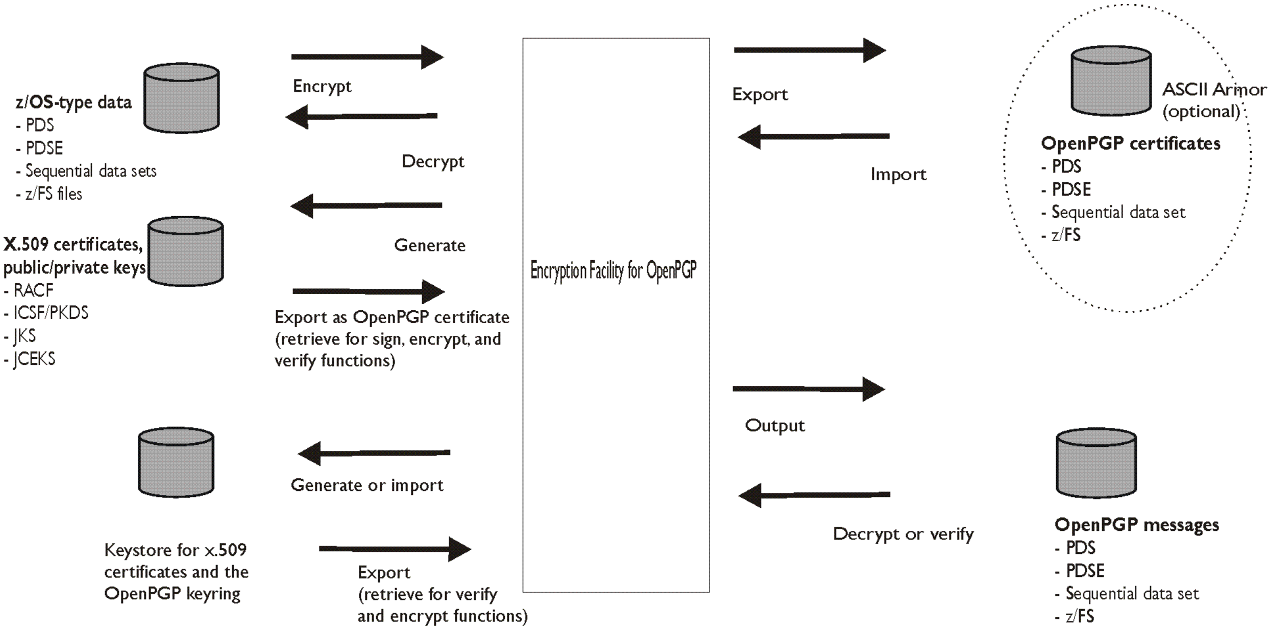

Figure 1. Encrypting and decrypting data and processing certificates and keys with Encryption Facility for OpenPGP

Using z/OS data sets

Encryption Facility for OpenPGP allows you to use z/OS data as input or output for OpenPGP encryption and decryption services. Encryption Facility for OpenPGP uses the IBM JZOS component to read and write to z/OS data sets and accepts the following kinds of z/OS data sets:

- Sequential data sets.
- Partitioned data sets (PDS) and partitioned data sets extended (PDSE).
- Large data sets (DSNTYPE=LARGE) for z/OS V1R8 with Encryption Facility APAR OA22067 applied or later releases.

Encryption Facility for OpenPGP does NOT accept VSAM data sets as input. All z/OS output data sets must be preallocated. See "Reading and writing to z/OS data sets" on page 19.

Compressing data

Compressing data before encryption can make the encryption more efficient. In compliance with OpenPGP standards that recommends compressing data for encryption, Encryption Facility for OpenPGP supports compression and decompression of OpenPGP messages and other data. Compressed data is formatted by the OpenPGP standard and is only compatible with other software products that support OpenPGP messages. Certain file types, like Windows system compressed files (.zip), are not OpenPGP messages and are not necessarily compatible with Encryption Facility.

Encryption Facility for OpenPGP also supports zEnterprise Data Compression (zEDC). In order for Encryption Facility for OpenPGP to use the zEDC feature, you must be using IBM 31-bit SDK for z/OS, Java Technology Edition, Version 7 Release 1 or later. zEDC also requires the following:

- z/OS V2R1 operating system.
- IBM zEnterprise EC12 (with GA2 level microcode) or IBM zEnterprise zBC12.
- zEDC Express adapter.

Note: zEDC requires a minimum input buffer size for compression and decompression. If the input data is smaller than the minimum threshold, the data is processed by using traditional software-based compression and decompression.

For more information about zEDC, see *z/OS MVS Programming: Callable Services for High-Level Languages*.

Using ASCII Armor

Encryption Facility for OpenPGP can provide OpenPGP radix-64 encoding of messages (ASCII Armor). Furthermore, Encryption Facility for OpenPGP can import OpenPGP certificates encoded in ASCII Armor. ASCII Armor is a term defined in the Internet draft standard protocol for OpenPGP. If you use a z/OS data set as output for a certificate that is protected by ASCII armor, the data must be in EBCDIC, not ASCII, format.

Authenticating through digital signatures

OpenPGP specifies how to sign documents and how to use OpenPGP keys to encrypt, decrypt, or protect data. Encryption Facility for OpenPGP conforms to these specifications and is able to sign both binary and text documents. It self signs any OpenPGP certificates that it exports and verifies any signatures that it encounters when it imports an OpenPGP certificate.

Using security keys, certificates, and repositories

Encryption Facility for OpenPGP manages key information by using OpenPGP certificates and X.509 certificates. It relies on the following repositories:

- An OpenPGP keyring that is stored in an HFS/zFS file system. The OpenPGP keyring stores public-key information that is contained within the OpenPGP certificate.
- The Java keystore framework to access and generate public-key information. Public-key information is contained within the X.509 certificates and the private key (if available) of the key pair.

Table 1 on page 5 describes the name, keystore type, and software provider of each repository, operations that you can perform, and any notes about use. For information about the IBMJCECCA hardware provider, see IBMJCECCA Overview (www.ibm.com/docs/en/sdk-java-technology/8?topic=components-ibmjcecca). To change the keystore type, see "-keystore-type — Specify the keystore type" on page 63:

Table 1. Keystore and keyring repositories

Name	Keystore type	Operations	Notes
CCA RACF	JCECCARACFKS (IBMJCECCA provider)	• Read only • RSA sign, verify • RSA encrypt, decrypt	The hardware JCE provider must be set in the configuration. For signing and decrypting, a keystore password must be provided. Additionally, for decrypting, a key password must be provided and for signing, a signer's key password must be provided. The passwords can be anything, but they must match the keystore password. To remove the need to provide these passwords, see the DISABLE_RACF_PASSWORD_PROMPTS configuration option. Existing PKI infrastructure must be used to import or generate key information. Use this keystore type if you are using RACF and ICSF.
CCA	JCECCAKS (IBMJCECCA provider)	• RSA key generation (ICSF PKDS or clear key generated) • Prepare for use with existing ICSF key • RSA sign, verify • RSA encrypt, decrypt	The hardware JCE provider must be set in the configuration. Use this keystore type if you are using ICSF. If you use the hardware provider to generate keys, you must use the JCECCAKS keystore type.
JCEKS	JCEKS (software JCE provider)	• RSA, Digital Signature Algorithm (DSA) ElGamal key generation • RSA, DSA sign, verify • RSA, ElGamal encrypt, decrypt	Use this keystore type if you are using only Java software.
RACF	JCERACFKS (IBMJCECCA provider or other software JCE provider)	• Read only • RSA sign, verify • RSA encrypt, decrypt	For signing and decrypting, a keystore password must be provided. Additionally, for decrypting, a key password must be provided and for signing, a signer's key password must be provided. The passwords can be anything, but they must match the keystore password. To remove the need to provide these passwords, see the DISABLE_RACF_PASSWORD_PROMPTS configuration option. Existing PKI infrastructure must be used to import or generate key information. Use this keystore type for key access to certificates and keys that reside in RACF and are not in ICSF. JCERACF keystores are compatible with both JCECCA and JCE, that is, with both hardware and software providers.
JKS	JKS (Software IBMJCECCA provider)	• RSA, DSA ElGamal key generation • RSA, DSA sign, verify • RSA, ElGamal encrypt, decrypt	Use this keystore type if you are using only Java software.
OpenPGP keyring	N/A	• RSA, ElGamal encrypt • RSA, DSA verify	When you add new keys to a Java keystore (either through a generate or prepare), or export keys from a Java keystore, an OpenPGP certificate is generated.

Using ICSF and RACF

Encryption Facility for OpenPGP can make use of the following z/OS components and functions:

- Uses ICSF for cryptographic hardware acceleration
- Allows you to use existing cryptographic keys that ICSF maintains in the ICSF public key data set (PKDS)
- Generates ICSF clear or PKDS keys
- Allows you to use the existing RACF services for maintaining keys and X.509 certificates

Encryption Facility for OpenPGP uses the CCA JCE provider to allow the use of ICSF and RACF hardware acceleration and key services. For more information, see "ICSF considerations" on page 15 and "RACF considerations" on page 16.

Participating in OpenPGP key exchange

Encryption Facility for OpenPGP exchanges key information with trusted partners using OpenPGP certificates. You can exchange X.509 certificates by using existing PKI technology. As a result, Encryption Facility for OpenPGP only exports OpenPGP certificates either by retrieving an OpenPGP certificate from the keyring or by generating an OpenPGP certificate from an existing X.509 certificate in a keystore. Encryption Facility for OpenPGP, however, does not participate in the Web of Trust model. As a result, you need to ensure that all key information is authenticated before you import it into a system.

When it imports information, Encryption Facility for OpenPGP verifies all of the signatures whose public key it can access for an OpenPGP certificate and ensures that the format of the certificate adheres to Internet Standard RFC 4880.

Java algorithm support for Encryption Facility for OpenPGP

Encryption Facility for OpenPGP through the JCE providers is able to use the following symmetric algorithms to protect data:

- TripleDES (triple-length DES with 168-bit key)
- AES (128-bit, 192-bit, and 256-bit keys)
- Blowfish (128-bit keys)

Hardware acceleration support

Encryption Facility for OpenPGP can make use of ICSF and, depending on the hardware installed, hardware acceleration for encryption. Hardware acceleration depends on the kind of ICSF cryptographic hardware installed.

Table 2 on page 6 summarizes the processor type and cryptographic hardware and whether ICSF or the JCE supports cipher feedback (CFB) mode encryption required for processing OpenPGP encrypted messages.

Table 2. Hardware and CFB mode encryption support for symmetric algorithm

Processor and	Cryptographic hardware	CFB mode encryption support for symmetric algorithm
IBM z900, z800	with CCF with or without PCICC	• AES algorithm supported by ICSF • TDES algorithm supported by software JCE
IBM z990, z890	with or without PCIXCC with or without CEX2C	• AES algorithm supported by ICSF • TDES algorithm supported by CPACF and ICSF
IBM z9	with or without CEX2C	• AES 128-bit algorithm supported by CPACF and ICSF • AES 192-bit and 256-bit algorithm supported by ICSF • TDES supported by CPACF and ICSF

Table 2. Hardware and CFB mode encryption support for symmetric algorithm (continued)

Processor and	Cryptographic hardware	CFB mode encryption support for symmetric algorithm
IBM z10	with or without CEX2C with or without CEX3C	• AES 128-bit, 192-bit and 256-bit algorithm supported by CPACF and ICSF • TDES supported by CPACF and ICSF
IBM zEnterprise	with or without CEX3C with or without CEX4C	• AES 128-bit, 192-bit and 256-bit algorithm supported by CPACF and ICSF • TDES supported by CPACF and ICSF

Asymmetric algorithm support

Table 3 on page 7 summarizes the type of asymmetric algorithms that Encryption Facility for OpenPGP uses and whether Encryption Facility for OpenPGP or the JCE provider supports the algorithm for OpenPGP.

Table 3. Asymmetric algorithm support

Algorithm	Support for asymmetric keys
RSA	CCA JCE provider
ElGamal	Software JCE provider

Compression algorithm support

Table 4 on page 7 summarizes the type of compression algorithms that Encryption Facility for OpenPGP uses and where they are supported for OpenPGP.

Table 4. Compression algorithm support

Compression algorithm	Support for compression algorithm
ZIP	IBM Java Development Kit (SDK)
ZLIB	IBM Java Development Kit (SDK)

Note: Encryption Facility for OpenPGP uses the zEDC feature for compression if available and running with the required level of Java (IBM 31-bit SDK for z/OS, Java Technology Edition, Version 7 Release 1 or later).

zEDC requires a minimum input buffer size for compression and decompression. If the input data is smaller than the minimum threshold, the data is processed using traditional software-based compression and decompression.

For additional information about zEDC, see *z/OS MVS Programming: Callable Services for High-Level Languages*.

Digital signature support

Table 5 on page 8 summarizes the type of digital signature algorithms that Encryption Facility for OpenPGP uses and where they are supported for OpenPGP.

Table 5. Digital signature algorithm support

Digital signature algorithm	Support for digital signature algorithm
DSA/SHA1	For a z900 processor, CCA JCE provider. For all other hardware types, software JCE provider.
DSA/SHA224	Software JCE provider. CCA JCE provider.
DSA/SHA256	Software JCE provider. CCA JCE provider.
RSA/SHA1	Software JCE provider. CCA JCE provider.
RSA/SHA224	Software JCE provider. CCA JCE provider.
RSA/SHA256	Software JCE provider. CCA JCE provider.
RSA/SHA384	Software JCE provider. CCA JCE provider.
RSA/SHA512	Software JCE provider. CCA JCE provider.
RSA/MD2	CCA JCE provider.
RSA/MD5	CCA JCE provider.

Message digest algorithm support

Table 6 on page 8 summarizes the type of message digest algorithms that Encryption Facility for OpenPGP uses and where they are supported for OpenPGP.

Table 6. Message digest algorithm support

Message digest algorithm	Support for message digest algorithm
MD2	CCA JCE provider.
MD5	CCA JCE provider.
SHA-1	Software JCE provider. CCA JCE provider.
SHA224	Software JCE provider. CCA JCE provider.

Table 6. Message digest algorithm support (continued)

Message digest algorithm	Support for message digest algorithm
SHA256	Software JCE provider. CCA JCE provider.
SHA384	Software JCE provider. CCA JCE provider.
SHA512	Software JCE provider. CCA JCE provider.

Default OpenPGP algorithm preferences

Encryption Facility for OpenPGP supports the use of preference lists indicating the preference of certain algorithms for the certificate owner. The preference order is processed from the first item in the list to the last item in the list with the first item being the most preferred. The following topics list the default preferred algorithm lists created when generating an OpenPGP certificate:

- "Symmetric algorithm preference list" on page 9.
- "Message digest algorithm preference list" on page 9.
- "Compression algorithm preference list" on page 10.

Note: Each of the preferred algorithm settings can be overridden by the following command options in "Encryption Facility for OpenPGP Commands" on page 75:

- "CIPHER_NAME" on page 36.
- "DIGEST_NAME" on page 36.
- "COMPRESS_NAME" on page 37.
- "-cipher-name — Specify the algorithm for encryption " on page 56.
- "-compress-name — Specify the algorithm to use for compression" on page 56.
- "-digest-name — Specify the algorithm for the message digest" on page 58.

Symmetric algorithm preference list

The default preference list of symmetric algorithms that Encryption Facility for OpenPGP creates when generating an OpenPGP Certificate is:

- DES3
- AES256
- AES192
- AES128

Note: The preferred algorithm settings can be overridden by the following command options:

- "CIPHER_NAME" on page 36.
- "-cipher-name — Specify the algorithm for encryption " on page 56.

Message digest algorithm preference list

The default preference list of message digest algorithms that Encryption Facility for OpenPGP creates when generating an OpenPGP Certificate is:

- SHA1
- SHA512
- SHA384
- SHA256

Note: The preferred algorithm settings can be overridden by the following command options:

- “DIGEST_NAME” on page 36.
- “-digest-name — Specify the algorithm for the message digest” on page 58.

Compression algorithm preference list

The default preference list of compression algorithms that Encryption Facility for OpenPGP creates when generating an OpenPGP Certificate is:

- ZIP
- ZLIB

Note: The preferred algorithm settings can be overridden by the following command options:

- “COMPRESS_NAME” on page 37.
- “-compress-name — Specify the algorithm to use for compression” on page 56.

Supported key sizes

Keys that Encryption Facility for OpenPGP supports are as follows:

- RSA (JCEKS, JKS, JCERACFKS, JCECCAKS, and JCECCARACFKS keystores).
- DSA (JCEKS and JKS keystores) - 1024 or higher.
- ElGamal (JCEKS and JKS keystores) - 1024 or higher.

Large key sizes (for example, 1024, 2048, 4096, and so forth) can be generated and imported from other applications. To use these large key sizes, ensure that you have properly set up the Java environment by installing the unrestricted policy files. For information about the setup of the IBM Java Cryptography Extension Common Cryptographic Architecture (IBMJCECCA) hardware cryptographic provider and the IBM Java Cryptography Extension (IBMJCE) software cryptographic provider, see Security and Legacy Services Components (www.ibm.com/systems/z/os/zos/tools/java/products/sc.html).

Note: The IBMJCE provider is no longer supported on IBM Semeru Runtime Certified Edition for z/OS 11 and later releases.

Encryption Facility considerations when changing Java release levels

IBM Java Technology Edition, Version 7 Release 1 and earlier allowed for the creation of and the use of X.509 certificates with a null distinguished name (DN). X.509 certificates with a null DN are invalid because they are essentially certificates with no identity and may not be accepted by other products. While it is unlikely that you would be using an X.509 certificate with a null DN, it is possible. Because IBM Java Technology Edition, Version 8 and later no longer supports X.509 certificates with a null DN, Java keystores will not load and are unusable if they contain a null DN. Therefore, X.509 certificates with a null DN must be removed prior to migrating to IBM Java Technology Edition, Version 8 or later.

Before migrating to IBM Java Technology Edition, Version 8 or later, use Encryption Facility's list commands (-pA or -pK) to determine if you have any X.509 certificates with a null DN and then use Encryption Facility's delete commands (-xA or -xK) to remove them.

If you migrate to IBM Java Technology Edition, Version 8 or later before removing all X.509 certificates with a null DN, you will need to remove these with other tooling (for example, Java keytool) or migrate back to your previous IBM Java level and use Encryption Facility's delete commands (-xA or -xK) to remove them.

With the latest level of service, Encryption Facility checks for and does not allow the creation of a X.509 certificate with a null DN.

Note: IBM Java keystores created with IBM Java Technology Edition, Version 7 or later cannot be accessed by IBM Java Technology Edition, Version 6.0.1 or earlier due to stronger encryption. Once you have migrated to IBM Java Technology Edition, Version 7 or later, it is not recommended that you migrate to a prior IBM Java version.

For more information, see Greater Key and Keystore strength for IBMJCECCA available in z/OS Java 7 and 7.1 (www.ibm.com/support/docview.wss?uid=isg3T1022007).

Note: Before deleting an X.509 certificate, ensure that you do not have data encrypted using the public key within it. Data encrypted using the public key must be decrypted and then encrypted using another public key before you delete the X.509 certificate and its public key.

IBM Semeru Runtime Certified Edition for z/OS 11 and later releases no longer support the IBMJCE provider.

When invoking Encryption Facility for OpenPGP from the z/OS Unix shell with IBM Semeru Runtime Certified Edition for z/OS 17, some commands must include one or both of the following Java options:

```
--add-exports=java.base/sun.security.x509=ALL-UNNAMED
--add-exports=java.base/sun.security.util=ALL-UNNAMED
```

When invoking Encryption Facility for OpenPGP using JCL with IBM Semeru Runtime Certified Edition for z/OS 17, some commands require one or both of the following Java options in the IBM_JAVA_OPTIONS environment variable:

```
JAVA_OPT="--add-exports=java.base/sun.security.x509=ALL-UNNAMED"
JAVA_OPT="$JAVA_OPT --add-exports=java.base/sun.security.util=ALL-UNNAMED"
export IBM_JAVA_OPTIONS="$IJO $JAVA_OPT”
```

Supported character sets

Character sets that Encryption Facility for OpenPGP supports are as follows:

- Big5 CESU-8 COMPOUND_TEXT EUC-CN EUC-JP
- EUC-TW GB18030 GB2312 GBK hp-roman8
- IBM-1006 IBM-1025 IBM-1026 IBM-1027 IBM-1041
- IBM-1046 IBM-1046S IBM-1047 IBM-1088 IBM-1097
- IBM-1112 IBM-1114 IBM-1115 IBM-1122 IBM-1123
- IBM-1140 IBM-1141 IBM-1142 IBM-1143 IBM-1144
- IBM-1146 IBM-1147 IBM-1148 IBM-1149 IBM-1351
- IBM-1363 IBM-1363C IBM-1364 IBM-1370 IBM-1371
- IBM-1381 IBM-1382 IBM-1383 IBM-1385 IBM-1386
- IBM-1390 IBM-1399 IBM-273 IBM-277 IBM-278
- IBM-284 IBM-285 IBM-290 IBM-297 IBM-300
- IBM-33722 IBM-33722C IBM-420 IBM-420S IBM-424
- IBM-500 IBM-720 IBM-737 IBM-775 IBM-808
- IBM-834 IBM-835 IBM-836 IBM-837 IBM-838
- IBM-852 IBM-855 IBM-856 IBM-857 IBM-858
- IBM-860 IBM-861 IBM-862 IBM-863 IBM-864
- IBM-865 IBM-866 IBM-867 IBM-868 IBM-869

- IBM-871 IBM-874 IBM-875 IBM-897 IBM-918
- IBM-922 IBM-924 IBM-927 IBM-930 IBM-932
- IBM-935 IBM-937 IBM-939 IBM-942 IBM-942C
- IBM-943C IBM-947 IBM-948 IBM-949 IBM-949C
- IBM-951 IBM-954 IBM-954C IBM-964 IBM-971
- ISO-2022-CN ISO-2022-CN-GB ISO-2022-JPISO-2022-KR ISO-8859-1
- ISO-8859-13 ISO-8859-14 ISO-8859-15 ISO-8859-16 ISO-8859-2
- ISO-8859-4 ISO-8859-5 ISO-8859-6 ISO-8859-6S ISO-8859-7
- ISO-8859-9 JIS0201 JIS0208 JIS0212 Johab
- KOI8-RU KOI8-U KSC5601 MacArabic MacCentralEurope
- MacCyrillic MacDingbat MacGreek MacHebrew MacIceland
- MacRomania MacSymbol MacThai MacTurkish MacUkraine
- Shift_JIS TIS-620 US-ASCII UTF-16 UTF-16BE
- UTF-32 UTF-32BE UTF-32LE UTF-8 UTF-8J
- windows-1251 windows-1252 windows-1253 windows-1254 windows-1255
- windows-1256S windows-1257 windows-1258 windows-874 windows-932

Hardware and software requirements

The following topics describe hardware and software requirements for Encryption Facility for OpenPGP. For hardware and software requirements for Encryption Facility for z/OS Version 1.2 and later, see *IBM Encryption Facility for z/OS: Planning and Customizing*.

Hardware requirements

IBM Encryption Facility for z/OS Version 1.2 runs on IBM Z processors that are currently in service with IBM. If an IBM Z processor level goes out of service with IBM, Encryption Facility for z/OS will no longer be supported on that processor level and you must upgrade to an IBM Z processor level that is still in service.

OpenPGP support and hardware cryptography:

- For AES or TDES symmetric encryption, use one of the following:
  - CPACF only (no cryptographic coprocessors). The -c command for passphrase-based encryption (PBE) is supported in a CPACF only environment with no cryptographic coprocessors. The -e command for public-key cryptography is not available in a CPACF only environment because a cryptographic coprocessor is required to encrypt the symmetric session key.
  - CPACF with PCIXCC / CEX2C / CEX3C / CEX4C
  - CCF
  - CCF with PCICC
- For signatures or session key encryption using 2048-bit keys or 2048-bit RSA key generation, use one of the following:
  - CEX2C / CEX3C / CEX4C
  - PCIXCC
  - PCICC with PCI Crypto 2048 bit Enablement Feature 0867
- For signatures or session key encryption using RSA keys generated through RACF and ICSF, or directly through ICSF, use one of the following:
  - CEX2C / CEX3C / CEX4C
  - PCIXCC

- PCICC

Software requirements

IBM Encryption Facility for z/OS Version 1.2 has the following software requirements:

- z/OS
- IBM 31-bit SDK for z/OS, Java Technology Edition
- Integrated Cryptographic Services Facility (ICSF)

IBM Encryption Facility for z/OS requires its software levels to be at a level that is still in service with IBM. At the time of this publication, the minimum service levels for these software programs are z/OS (5694-A01) Version 1 Release 12 or later release, IBM 31-bit SDK for z/OS, Java Technology Edition (5655-R31) Version 6 or later release, and Integrated Cryptographic Services Facility (ICSF) FMID HCR7770 or later release.

Note: If a software requirement level goes out of service with IBM, Encryption Facility for z/OS will no longer support that software level and you must upgrade to a software level that is still in service.

You cannot use IBM Z data that has been processed through CSDFILEN, CSDFILDE or through the Encryption Facility for z/OS Client (including the Decryption Client) in Encryption Facility Version 1.1 and Version 1.2 with Encryption Facility for OpenPGP Version 1.2. For information about available product functions for Encryption Facility for z/OS, see *IBM Encryption Facility for z/OS: Planning and Customizing*.

To use z/OS large data sets (DSNTYPE=LARGE on the JCL DD statement) with Encryption Facility for OpenPGP, you must apply Encryption Facility APAR OA22067 or later service that supersedes APAR OA22067.

For detailed instructions specifying full function cryptography including large key sizes and for using Java cryptography on z/OS, see the z/OS Java home page (www.ibm.com/docs/en/semeru-runtime-ce-z).

Chapter 2. Getting started

This topic describes installation tasks and considerations for getting started using Encryption Facility for OpenPGP:

- "How do I install Encryption Facility for OpenPGP?" on page 15
- "ICSF considerations" on page 15
- "RACF considerations" on page 16
- "Batch, UNIX System Services, and Java considerations" on page 16
- "Java algorithm support for Encryption Facility for OpenPGP" on page 6
- "Getting started basic steps" on page 16

How do I install Encryption Facility for OpenPGP?

Encryption Facility for OpenPGP is part of the licensed code for the Encryption Facility for z/OS product optional feature Encryption Services Version 1.2 and later. For detailed installation information, see *IBM Encryption Facility for z/OS: Program Directory*.

ICSF considerations

If you have ICSF installed, see "Software requirements" on page 13 to ensure that you are using the required level for Encryption Facility.

If you need information about installing, planning, and implementing ICSF, see the following publications:

- *z/OS Cryptographic Services ICSF Overview*
- *z/OS Cryptographic Services ICSF System Programmer's Guide*
- *z/OS Cryptographic Services ICSF Administrator's Guide*

Encryption Facility for OpenPGP can make use of ICSF to manage cryptographic keys for encrypted data.

ICSF supports the following cryptographic standards and architectures:

- IBM Common Cryptographic Architecture (CCA) that is based on the ANSI Data Encryption Standard (DES)
- Advanced Encryption Standard (AES).

ICSF Cryptographic keys: In an OpenPGP-based system, two parties must obtain a shared secret key that is used to protect data. Sharing secret keys establishes a secure communications channel. The OpenPGP Internet draft RFC 4880 describes a format that encrypts the shared secret key and then includes the protected secret key in the produced OpenPGP encrypted message. You can use either passphrase-based encryption (PBE) or public key-based encryption. Public key-based encryption allows the use of ICSF keys to protect the shared secret key.

For public key cryptography, ICSF supports the RSA algorithm. For digitially-signing data, ICSF supports the RSA algorithm. For OpenPGP public key-based data exchange or digital signature verification, each party establishes a pair of cryptographic keys, which includes a public key and a private key. For ICSF and hardware support information, see *z/OS Cryptographic Services ICSF Administrator's Guide*.

Both parties publish their public keys in a reliable information source or by exchanging X.509 or OpenPGP certificates that contain the public key information while they maintain their private keys in secure storage. The public key information can be used to encrypt a message that only the trusted partner can decrypt, or it can be used to verify a signature that the trusted partner produces.

Generating and storing RSA keys in the PKDS: Encryption Facility for OpenPGP can generate and store RSA public and private keys in the ICSF public key data set (PKDS). In addition, Encryption Facility for OpenPGP can use existing ICSF keys in the PKDS after preparing a CCA keystore. These RSA keys are used by Encryption Facility to protect the symmetric keys that protect the data, digitally sign the data, or both. To enable this function, the CCA JCE provider must be specified in the JCE provider list and the keystore type must be JCECCAKS. See Table 1 on page 5.

RSA public and private keys for encryption can be stored in the ICSF PKDS. These RSA keys are used by Encryption Facility to protect the symmetric keys that protect the data.

ICSF keystores are limited to 2048-bit keys.

For information about Encryption Facility for z/OS and ICSF, see *IBM Encryption Facility for z/OS: Planning and Customizing*.

RACF considerations

You can use RACF to help you store RSA public and private keys for encryption in the ICSF PKDS. You can also specify the PKDS labels to use when you store public or private keys in the PKDS and can list PKDS labels of public/private key pairs from existing certificates that reside in the RACF database.

The certificate management services of RACF allow you to establish a limited scope certificate authority for your internal and external users, issuing and administering digital certificates in accordance with your own organization’s policies.

Encryption Facility for OpenPGP uses Java's JCE keystore so that JCERACFKS and JCECCARACFKS keystore types can successfully interpret and retrieve X.509 certificates stored in RACF. See Table 1 on page 5.

As required by the JCE, PERSONAL and TRUSTED type certificates must have existing private keys. The caller must also have permission to access the private key material in the PKDS. If a certificate marked PERSONAL and TRUSTED has a null private key, or if the caller does not have the proper permissions, the loading of the entire keystore fails with the following message:

```
CSD0050I Command processing ended abnormally:
The private key of <cert label> is not available or no
authority to access the private key.
```

Where <*cert label*> is the label of the certificate whose private key material cannot be accessed.

For more information, see the RACF and key rings topic in *z/OS Security Server RACF Security Administrator's Guide*.

RACF keystores are limited to 2048-bit keys.

For information about using RACF to store keys and generate labels, see *IBM Encryption Facility for z/OS: Planning and Customizing*.

Batch, UNIX System Services, and Java considerations

Batch

To launch Encryption Facility for OpenPGP from batch, IBM provides the IBM JZOS Batch Toolkit for z/OS (JZOS). For sample code, see “Sample JCL and code” on page 179. For complete information about setting up and using Java and the SDK for z/OS and for JZOS, see the z/OS Java home page (www.ibm.com/docs/en/semeru-runtime-ce-z).

UNIX Systems Services

Use UNIX System Services with Encryption Facility for OpenPGP commands like -g that generate key pairs to serve as the system key for signatures.

Environment variables for the Java JVM

IBM provides sample code in SAMPLIB for the Java environment script to configure any environment variables for the Java JVM. For sample code, see “Sample JCL and code” on page 179.

Getting started basic steps

Consider the following basic steps for getting started with Encryption Facility for OpenPGP:

1. Ensure that you have the required hardware installed. See “Hardware requirements” on page 12.
2. Ensure that you have the required software levels installed. See “Software requirements” on page 13.
3. Use the OpenPGP keyring to store OpenPGP certificates. Use the appropriate keystore repository as supplied by the software JCE provider to store X.509 certificates. See Table 1 on page 5.
4. Ensure that you have defined the following Java runtime variable in the shell script code:

   ```
   export LIBPATH=$LIBPATH:/usr/lib/java_runtime
   ```

   For sample code, see “Sample JCL and code” on page 179.

Chapter 3. Using Encryption Facility for OpenPGP

This topic presents information about using Encryption Facility for OpenPGP.

- "Reading and writing to z/OS data sets" on page 19
- "OpenPGP messages" on page 20
- "Authenticating digital signatures" on page 21
- "Using the OpenPGP keyring" on page 22

For complete information about OpenPGP standards, see RFC 4880 (tools.ietf.org/html/rfc4880).

Reading and writing to z/OS data sets

Encryption Facility for OpenPGP allows you to use data from z/OS data sets that you can then process on any OpenPGP-compliant system. Encryption Facility for OpenPGP uses the JZOS component to access z/OS data sets.

Types of data sets

Encryption Facility for OpenPGP accepts the following data sets as input and output:

- Sequential
- PDS
- PDSE
- Large (DSNTYPE=LARGE) for z/OS V1R8 with Encryption Facility APAR OA22067 applied or later releases

You must pre-allocate storage for all output data sets that you use with Encryption Facility for OpenPGP.

Restrictions using data sets

Keep in mind the following restrictions for data sets:

- Encryption Facility supports using data definition (DD) statements with DDNAMES in the JCL for access to the supported data set types, but you cannot use DDNAMES for UNIX System Services files.
- Encryption Facility for OpenPGP does NOT accept VSAM data sets as input or output.
- You cannot use fixed block (RECFM=FB) format data sets as the output for encryption or signature command processing.

  Encryption Facility does not retain data set information in the encrypted or signed binary data. For example, when you encrypt a data set that contains incomplete records, Encryption Facility does not retain information about the number of bytes for each record. Thus, if you decrypt to an output data set that might have the same attributes as the source data set, the output might not have the same data set record format. However, every record is completely filled before Encryption Facility starts a new record.
- You cannot create empty records in z/OS data sets. As a result, if an empty record is required for an empty line of text, Encryption Facility for OpenPGP writes a record of one space. In this case, when Encryption Facility for OpenPGP writes output to variable block data sets, it writes a record of one space. If an empty space is not acceptable, use a UNIX Systems Services file as output to Encryption Facility for OpenPGP. If the output must reside in a data set, transfer the output to a data set through TSO/E commands.
- When an input or output data set resides on tape instead of DASD, you must specify a DDNAME instead of the data set name on any argument value for Encryption Facility for OpenPGP.

Allocating data sets through the data definition (DD) statement

When you use the Java batch program and the JCL data definition (DD) statement to allocate a data set, be sure to specify the DD name instead of the data set name on Encryption Facility for OpenPGP command options. For example, consider the following JCL that defines the data set EFR2.ENC.OUT:

```
DDDEF DD DSN=EFR2.ENC.OUT,
               DISP=(NEW,KEEP),
               DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760),
              UNIT=3390,VOL=SER=SEVMW2,
              SPACE=(CYL,(5,1))
```

To specify the data set on the -o option, for example, specify the DD statement label DDDEF instead of the name of the data set EFR2.ENC.OUT as follows:

```
 -o '//DD:DDDEF'
```

For sample code, see "Sample JCL and code" on page 179.

Language Environment (LE)

When you use variable record length data sets, text data, or ASCII armored certificates, ensure that the following environment variable is set for Language Environment® (LE):

```
export _EDC_ZERO_RECLEN=Y
```

If you do not set the environment variable, Encryption Facility for OpenPGP ignores any record without bytes, and an error can occur. Records without bytes are essential for processing ASCII Armor messages.

Other data set considerations

If you plan to use z/OS data sets for OpenPGP encryption, consider the following:

- Ensure that ICSF is active on the z/OS system.
- Ensure that users have access to the UNIX System Services files.
- Use the necessary JCL to run batch programs that use Encryption Facility services. Encryption Facility V1R2 ships sample JCL and an environment file. This JCL leverages the Java batch component of the IBM Java SDK. For more information, see the z/OS Java home page (www.ibm.com/docs/en/semeru-runtime-ce-z).

OpenPGP messages

OpenPGP messages consist of message packets. Each packet consists of a packet header, followed by the packet body. The packet header is of variable length. The first octet of the packet header is called the "Packet Tag." It determines the format of the header and denotes the packet contents. The remainder of the packet header is the length of the packet.

For details about the structure of packets, see RFC 4880 (tools.ietf.org/html/rfc4880).

Using Encryption Facility for OpenPGP commands and options

Encryption Facility for OpenPGP supports all OpenPGP packets and also processes the expiration subpacket and the preference subpacket that OpenPGP defines. Encryption Facility for OpenPGP uses Java-based commands and options on UNIX System Services for OpenPGP messages, certificates, and data. You can use a configuration file to specify options for the commands, or you can specify options on the command line itself to override the values in the configuration file. For command syntax and options, see Chapter 4, "Encryption Facility for OpenPGP commands," on page 23.

Table 7 on page 21 shows the services that Encryption Facility for OpenPGP can perform and the command that performs it.

Table 7. OpenPGP command services

Service	Encryption Facility for OpenPGP command
Signs the contents of an OpenPGP message	-s
Encrypts the contents of an OpenPGP message	-e, -c
Decrypts the contents of an OpenPGP message	-d
Verifies a signed OpenPGP message	-v
Lists information about public keys in the keyring file	-pP,-pK
Lists information about public keys in the Java keystore	-pA,-pK
Generates key pairs	-g
Imports OpenPGP certificates	-i
Exports OpenPGP certificates from the Open PGP keyring or keystore	-eK
Exports OpenPGP certificates from the OpenPGP keyring	-eP
Creates OpenPGP certificates from an x.509 certificate in the keystore, updates the newly created certificate in the OpenPGP keyring, and exports the OpenPGP certificate from the OpenPGP keyring	-eA
Deletes OpenPGP certificates from the OpenPGP keyring or keystore	-xK
Deletes OpenPGP certificates from the OpenPGP keyring	-xP
Deletes OpenPGP certificates for x.509 aliases from the keystore	-xA
Sets up Java keystores for use with predefined ICSF and RACF keys	-prepare
Rebuilds the indexes for the OpenPGP keyring file	-rebuild-key-index

Authenticating digital signatures

OpenPGP standards provide authentication methods for signing documents and generating OpenPGP keys. Encryption Facility for OpenPGP conforms to these standards and provides the following support. The Encryption Facility for OpenPGP commands are indicated:

Table 8. OpenPGP services and Encryption Facility commands

Service	Encryption Facility for OpenPGP command
Accepts and produces OpenPGP certificates only. Also, self-signs any OpenPGP certificates that it exports as well as attempts to verify any signatures it encounters when it imports an OpenPGP certificate.	-eA,-eP,-eK, -i
Signs data and packages the data into an OpenPGP message containing the signature packets that can be used to verify the integrity of the data.	-s

Table 8. OpenPGP services and Encryption Facility commands (continued)

Service	Encryption Facility for OpenPGP command
Generates an OpenPGP messages for a detached signature. The signed data is not altered or included in the output OpenPGP message.	-b
Verifies a detached signature or signed data	-v
Prepares an existing key in the ICSF PKDS for use with Encryption Facility for OpenPGP.	-prepare

Encryption Facility for OpenPGP does not support the following:

- Clear-text signatures
- Importing X.509 certificates

Using the OpenPGP keyring

Within an OpenPGP keyring, Encryption Facility for OpenPGP stores any OpenPGP certificates when any of the following conditions occur:

- Generating a certificate during key pair generation.
- Importing a certificate.
- Generating a certificate as a result of exporting an OpenPGP representation (alias) of an X.509 certificate in a Java keystore.

The OpenPGP keyring by default uses the following directory path and file name:

```
/etc/encryptionfacility/ibmpkring.ikr
```

This default can be changed by specifying a different directory path and file name using the KEY_RING_FILENAME option in the ibmef.config file. The OpenPGP keyring file uses two index files located in the same directory with the suffixes of .kidx and .uidx. For example, the default OpenPGP keyring uses the following two index files:

```
/etc/encryptionfacility/ibmpkring.ikr.kidx
/etc/encryptionfacility/ibmpkring.ikr.uidx
```

Use UNIX System Services permissions to restrict access to only authorized users in order to protect the OpenPGP keyring, the associated keyring index files, and the directory they are contained in. See *z/OS UNIX System Services Planning* for information about establishing UNIX security and setting up access control lists to control access to files and directories.

Chapter 4. Encryption Facility for OpenPGP commands

You can use the Encryption Facility for OpenPGP Java-based commands and options to sign OpenPGP messages, encrypt or decrypt OpenPGP messages, verify OpenPGP messages, and manage OpenPGP certificates and keys. This topic includes the following topics:

- "Configuration file and home directory" on page 23
- "Latest command options and the updated ibmef.config file" on page 50
- "Encryption Facility for OpenPGP options and commands" on page 52

For command usage and examples, see Chapter 6, "JCL, command examples, and reference," on page 179.

Configuration file and home directory

Encryption Facility for OpenPGP makes use of the configuration file `ibmef.config` that is stored in the directory `-homedir`.

The file is optional. You can specify command options in the configuration file to perform integrity services for OpenPGP messages. You can specify all options for Encryption Facility for OpenPGP commands through the configuration file, or you can specify options on the command line itself.

If you do not specify a directory by using the `-homedir` command line option, Encryption Facility for OpenPGP tries to read the configuration file `/etc/encryptionfacility/ibmef.config`.

Each option in the configuration file `ibmef.config` is summarized including the format, description (with the default value and reference to the appropriate command option), and any arguments for the configuration file option.

The `ibmef.config` file is shipped encoded as IBM-1047 and is always expected in this format to be processed properly.

OUTPUT_FILE

Format

OUTPUT_FILE *user-specified-name*

Description

Specifies the name of the destination file or z/OS-type data set.

For data sets, you must preallocate the data set and specify a prefix of //. For example:

```
//U1.HIGHRISK.EF.OUTPUT
```

For PDSE data sets, you must enclose the name in single quotations. For example:

```
'//SYS.TEST.PDS(MEMBER1)'
```

When you use a DD statement in the JCL to allocate the data set, be sure to specify the DD name instead of the data set name and enclose it in quotations. For example, for a data set specified on the DD statement labeled DDDEF:

```
 -o '//DD:DDDEF'
```

See “Allocating data sets through the data definition (DD) statement” on page 20 and “Sample JCL and code” on page 179.

Default: None.

Equivalent command option: “-o — Specify an output location” on page 64.

Arguments

For *user-specified-name*, the name of the file or data set.

KEY_RING_FILENAME

Format

KEY_RING_FILENAME *file-name*

Description

Specifies the name of the OpenPGP keyring file where you can store OpenPGP certificates.

Default: **/etc/encryptionfacility/ibmpkring.ikr**

Equivalent command option: None.

Arguments

For *file-name*, the name of the OpenPGP keyring file.

USE_ASYNC_IO

Format

USE_ASYNC_IO

Description

Activates asynchronous I/O processing.

Default: If not specified, does not activate.

Equivalent command option: None.

Arguments

None.

USE_ASYNC_COMPRESS

Format

USE_ASYNC_COMPRESS

Description

Activates asynchronous compression.

Default: If not specified, does not activate.

Equivalent command option: None.

Arguments

None.

USE_ASYNC_CIPHER

Format

USE_ASYNC_CIPHER

Description

Activates asynchronous encryption.

Default: If not specified, does not activate.

Equivalent command option: None.

Arguments

None.

JAVA_KEY_STORE_TYPE

Format

JAVA_KEY_STORE_TYPE *type*

Description

Sets the Java keystore type.

Default: Type as specified in the `keystore.type` property in the Java security properties file, `JAVA_HOME/lib/security/java.security`; if the property does not exist, **JKS** is the default. If you use the hardware provider to generate keys, you must use the JCECCAKS keystore type.

Equivalent command option: "-keystore-type — Specify the keystore type" on page 63.

Arguments

For *type*, one of the following:

- JKS
- JCEKS
- JCECCAKS
- JCECCARACFKS - Read only
- JCERACFKS - Read only

JAVA_KEY_STORE_NAME

Format

JAVA_KEY_STORE_NAME *user-specified-name*

Description

Sets the file name of the Java keystore. The value can either be one of the following:

- UNIX System Services filename

- RACF keyring name.

For a RACF keyring name, the keystore type must be JCERACFKS. For a RACF keyring with keys in PKDS, the keystore type must be JCECCARACFKS and the hardware provider is required. You cannot specify z/OS-type data set names.

Default: None.

Equivalent command option: "-keystore — Specify the name of the Java keystore" on page 63.

Arguments

For *user-specified-name*, the name of the Java keystore.

KEYSTORE_PASSWORD

Format

KEYSTORE_PASSWORD *user-specified-password*

Description

Specifies the password used to access the keystore.

Default: None.

Equivalent command option: "-keystore-password — Specify the keystore password" on page 63 .

Arguments

For *user-specified-password*, the password for the keystore.

KEY_PASSWORD

Format

KEY_PASSWORD *user-specified-password*

Description

Specifies the password for generating a new key in the keystore or for self-signing certificates when exporting a keystore key as an OpenPGP certificate.

Default: None.

Equivalent command option: "-key-password — Specify the password for a new key" on page 62.

Arguments

For *user-specified-password*, the password for generating a new key in the keystore.

KEY_ALIAS

Format

KEY_ALIAS *user-specified-alias*

Description

Specifies the alias for generating a new key in the keystore.

Default: None.

Equivalent command option: “-key-alias — Specify the alias of a new key” on page 62.

Arguments

For *user-specified-alias*, the alias for generating a new key in the keystore.

KEY_SIZE

Format

KEY_SIZE *size*

Description

Specifies the key size for generating a new key in the keystore.

Default: **1024**.

Equivalent command option: “-key-size — Specify the key size to generate” on page 62.

Arguments

For *size*, a key size to generate in the keystore. Key sizes depend on the hardware and software you are using. See “Java algorithm support for Encryption Facility for OpenPGP” on page 6.

SIGNERS_KEY_PASSWORD

Format

SIGNERS_KEY_PASSWORD *user-specified-password*

Description

Specifies the password to access the signer's key in the keystore on this system. Encryption Facility uses this key when it works with message signatures.

Default: None.

Equivalent command option: “-signers-key-password — Specify a password for the system key” on page 69.

Arguments

For *user-specified-password*, the password to access the signer's key.

SIGNERS_KEY_ALIAS

Format

SIGNERS_KEY_ALIAS *user-specified-alias*

Description

Specifies the alias of the signer's key in the keystore for the key on this system. Encryption Facility uses this key when it works with message signatures.

Default: None.

Equivalent command option: “-signers-key-alias — Specify an alias for the system key” on page 69.

Arguments

For *user-specified-alias*, the alias of the signer's key.

SYSTEM_CA_KEY_ALIAS

Format

SYSTEM_CA_KEY_ALIAS *user-specified-alias*

Description

Specifies the alias of a certificate authority (CA) key in the keystore.

Default: None. Self-signs generated certificates.

Equivalent command option: “-system-CA-key-alias — Specify an alias for a new key pair certificate” on page 70.

Arguments

For *user-specified-alias*, the alias of the CA key.

SYSTEM_CA_KEY_PASSWORD

Format

SYSTEM_CA_KEY_PASSWORD *user-specified-password*

Description

Specifies the password to access a certificate authority (CA) key in the keystore.

Default: None. Self-signs generated certificates.

Equivalent command option: “-system-CA-key-password — Specify a password for the certificate authority key” on page 70.

Arguments

For *user-specified-password*, the password to access a CA key.

LOG_FILE

Format

LOG_FILE *file-name*

Description

Enables logging to a file. You must ensure that ACTIVE_LOGGERS and DEBUG_LEVEL are set for log data to be written. Log output is in XML so the file has an `.xml` extension.

Default: None. Log not active.

Equivalent command option: “-log-file — Write trace information to a file” on page 64.

Arguments

For *file-name*, the name of the log file.

CREATE_TRACE

Format

```
CREATE_TRACE
```

Description

Enables the logging of trace and debug information to STDERR. You must ensure that **ACTIVE_LOGGERS** and **DEBUG_LEVEL** are set for data to be written.

Default: If not specified, does not display the trace information to STDERR.

Equivalent command option: "-debug-on — Activate debugging information" on page 57.

Arguments

None.

ACTIVE_LOGGERS

Format

ACTIVE_LOGGERS *value*

Description

When you specify LOG_FILE, CREATE_TRACE, or both, specifies the components that produce debugging and log information.

Default: ACTIVE_LOGGERS **-1** is initially set in the configuration file and indicates logging for all components. If you do not specify a value, **0** is the default.

Equivalent command option: "-debug number— Specify a bit mask value for logging" on page 57.

Arguments

For *value*, one of the following component trace options:

0
: No logging active

1
: Async facility

2
: Cipher facility

4
: Compress facility

8
: Digital signature facility

16
: I/O facility

32
: Message component

64
Packet component

128
ASCII Armor facility

256
Primitives component

512
Passphrase-based encryption component

1024
General facility

2048
Initialization

4096
Command processor

-1
All components

DEBUG_LEVEL

Format

DEBUG_LEVEL *value*

Description

Specifies how debug information is to be collected. For all levels, specify 0.

Default: DEBUG_LEVEL **700** is initially set in the configuration file. If you do not specify a value, **0** is the default.

Equivalent command option: "-debug-level level — Specify a level for trace information to be sent to the log file" on page 57.

Arguments

For *value*, one of the following options:

1000
SEVERE (error information only)

900
WARNING and SEVERE (error and warning information)

800
WARNING, SEVERE, INFO (error, warning, and informational messages)

700
WARNING, SEVERE, INFO, CONFIG (error, warning, informational and configuration messages)

500
WARNING, SEVERE, INFO, CONFIG, Fine TRACE/DEBUG (error, warning, informational and configuration messages and fine level of debug tracing)

400
WARNING, SEVERE, INFO, CONFIG, Finer TRACE/DEBUG (error, warning, informational and configuration messages and finer level of debug tracing)

300
WARNING, SEVERE, INFO, CONFIG, Finest TRACE/DEBUG (error, warning, informational and configuration messages and finest level of debug tracing)

0
All

LITERAL_TEXT_CHARSET

Format

LITERAL_TEXT_CHARSET *set*

Description

Specifies a character set. Encryption Facility performs character conversions as follows:

- When producing an Encryption Facility message (commands **-e**, **-s**, and **-c**), Encryption Facility converts the data from the system's character set to this value. In addition to the character conversions, Encryption Facility converts end-of-line characters to carriage return and line feed.
- When processing an RFC 4880 message or an Encryption Facility message (commands **-d** and**-v**), Encryption Facility converts the data from this value to the system's character set. In addition to the character conversions, Encryption Facility converts end-of-line characters to line feed.
- When creating a detached signature (command **-b**), Encryption Facility converts the data from the local code page to UTF-8 and uses the UTF-8 characters to calculate or verify the detached signature. (Note that in this instance, the specified value is ignored as the local code page is assumed for the text.) In addition to the character conversions, Encryption Facility converts end-of-line characters to carriage return and line feed.
- When verifying with detached signatures (command **-v**), Encryption Facility converts the data from this character set to UTF-8 and uses the UTF-8 characters to calculate or verify the detached signature. In addition to the character conversions, Encryption Facility converts end-of-line characters to carriage return and line feed.

A value of _LOCAL is equivalent to the system's current character set.

Default: If not specified, the data is processed as binary. If specified without a value, UTF-8 is the default.

Equivalent command option: "-t — Treat input as text" on page 71.

Arguments

For *set*, a character set value.

JCE_PROVIDER_LIST

Format

JCE_PROVIDER_LIST *string*

Description

Prefixes the list of JCE providers in the java.security file that resides in `$JAVA_HOME/lib/security/java.security`

A JCE provider in the list implements all cryptographic functions. See the java.security file in the `${`*`java-home`*`}/lib/security/java.security` directory for more information on the provider list.

Default: List as specified in the "security.provider.<*n*>" properties in the Java security properties file JAVA_HOME/lib/security/java.security, JCE_PROVIDER_LIST com.ibm.crypto.hdwrCCA.provider.IBMJCECCA.

For hardware cryptographic acceleration, set the value to JCE_PROVIDER_LIST com.ibm.crypto.hdwrCCA.provider.IBMJCECCA. However, depending on the algorithms that you use and your ICSF and hardware and software configuration, you might obtain some errors.

Equivalent command option: "-jce-providers — Specify JCE class names" on page 61.

Arguments

For hardware cryptographic acceleration that ICSF provides, use the following default value for JCE_PROVIDER_LIST:

```
com.ibm.crypto.hdwrCCA.provider.IBMJCECCA
```

Otherwise, for *string*, your own value for your hardware provider.

RNG_JCE_PROVIDER

Format

RNG_JCE_PROVIDER *value*

Description

Sets the JCE provider according to the system random number generator.

For ICSF hardware cryptographic acceleration with an enabled cryptographic module, set the value for RNG_JCE_PROVIDER as follows:

```
com.ibm.crypto.hdwrCCA.provider.IBMJCECCA
```

If an ICSF cryptographic module is not enabled, set the value for RNG_JCE_PROVIDER as follows:

```
com.ibm.crypto.provider.IBMJCE
```

Note: The IBMJCE provider is no longer supported on IBM Semeru Runtime Certified Edition for z/OS 11 and later releases.

If you do not specify a value, the random number generator uses the FIRST provider defined in the JCE provider list.

Default: The first JCE provider in the list.

Equivalent command option: None.

Arguments

For *value*, the name of a fully-qualified JCE provider class name.

USE_ASCII_ARMOR

Format

USE_ASCII_ARMOR

Description

Specifies that when you export an OpenPGP certificate or create an OpenPGP Message (Encrypt, Sign or Compress), the output will be encoded using the ASCII Armor format.

Default: If not specified, do not use ASCII armor.

Equivalent command option: "-a — Use ASCII Armor for the OpenPGP certificate output" on page 53.

Arguments

None.

ARMOR_COMMENT

Format

ARMOR_COMMENT *user-specified-comment*

Description

Adds a comment to an OpenPGP certificate or message that is encoded by ASCII Armor.

Default: None.

Equivalent command option: "-comment — Add a comment header to ASCII Armorized messages" on page 56.

Arguments

For *user-specified-comment*, a comment string.

DISABLE_RACF_PASSWORD_PROMPTS

Format

DISABLE_RACF_PASSWORD_PROMPTS

Description

Disables prompts for passwords when using RACF-based keystores or keys. By default, Encryption Facility prompts users to enter a password when using RACF-based keystores or keys even though they are not actually used to protect access.

Default: If this option is omitted, then by default, prompts for RACF passwords will be enabled.

Equivalent command option: None.

Arguments

None.

RECIPIENT_USER_ID

Format

RECIPIENT_USER_ID *user-specified-IDs*

Description

Specifies one or more user IDs for the recipients of an encrypted message. Encryption Facility attempts to find the public key for the recipient in the keyring and uses asymmetric encryption of the session key.

Default: None.

Equivalent command option: "-rP — Encrypt for a specified user ID" on page 67.

Arguments

For *user-specified-IDs*, one or more user IDs separated by commas.

RECIPIENT_KEY_ID

Format

-RECIPIENT_KEY_ID *user-specified-key IDs*

Description

Specifies one or more 8-byte hexadecimal values for the key ID of each recipient of an encrypted message. Encryption Facility attempts to find the public key for the recipient in the keyring and uses asymmetric encryption of the session key.

Default: None.

Equivalent command option: "-rK — Encrypt for a specified key ID" on page 66.

Arguments

For *user-specified key-IDs*, one or more key IDs separated by commas.

RECIPIENT_ALIAS

Format

RECIPIENT_ALIAS *user-specified-aliases*

Description

Specifies one or more aliases in the keystore for each recipient of an encrypted message. Encryption Facility attempts to find the public key for the recipient in the keyring and uses asymmetric encryption of the session key.

Default: None.

Equivalent command option: "-rA — Encrypt using the public key from the Java keystore" on page 66.

Arguments

For *user-specified-aliases*, one or more aliases separated by commas.

COMPRESSION

Format

COMPRESSION *value*

Description

Specifies compression of an encrypted message.

Default: 0.

Equivalent command option: "-z — Compress data" on page 74.

Arguments

For *value*, a compression value. You can specify one of the following values:

0
: Do not use compression.

9
: Use the best compression possible. **Setting this value can result in a considerable impact to performance.**

1
: Use the best performance for compression.

-1
: Use default compression.

CONFIDENTIAL

Format

CONFIDENTIAL

Description

When processing an OpenPGP message or an Encryption Facility message, does not store the data in the message to a data set or file; instead, sends the data to STDOUT.

Default: If not specified, do not process as confidential.

Equivalent command option: "-no-save — Display data to STDOUT only " on page 64.

Arguments

None.

USE_EMBEDDED_FILENAME

Format

USE_EMBEDDED_FILENAME *file-name*

Description

When consuming an OpenPGP message or an Encryption Facility message, stores the data in the message to the file or data set that is specified in the message.

If you specify DEFAULT_OUTPUT_DIRECTORY, and the embedded filename does not refer to a data set, Encryption Facility writes the data to this directory.

Default: if not specified, do not use an embedded file name as output.

Equivalent command option: "-use-embedded-file — Write data to a file specified in the data packet" on page 72.

Arguments

For *file-name*, a file or data set name.

DEFAULT_OUTPUT_DIRECTORY

Format

DEFAULT_OUTPUT_DIRECTORY

Description

When using embedded filenames and the embedded name does not refer to a data set, stores the data in this directory using the embedded filename.

Default: Current working directory.

Equivalent command option: None.

Arguments

None.

CIPHER_NAME

Format

CIPHER_NAME *algorithm*

Description

When producing an encryption facility message, uses the specified algorithm for encryption.

Note: If this command option is not set, the default algorithm preferences is used as defined in “Symmetric algorithm preference list” on page 9.

Default: If the recipient preference is not available in the OpenPGP certificate, TRIPLE_DES.

Equivalent command option: “-cipher-name — Specify the algorithm for encryption ” on page 56.

Arguments

For *algorithm*, specify a valid encryption algorithm. You can run the `-list-algo` command to see valid algorithm values.

DIGEST_NAME

Format

DIGEST_NAME *algorithm*

Description

When producing an encryption facility message, uses the specified algorithm for hashing.

Note: If this command option is not set, the default algorithm preferences is used as defined in “Message digest algorithm preference list” on page 9.

Default: If the recipient preference is not available in the OpenPGP certificate, SHA_1.

Equivalent command option: “-digest-name — Specify the algorithm for the message digest” on page 58.

Arguments

For *algorithm*, specify a valid digest name for hashing. You can run the `-list-algo` command to see valid algorithm values.

COMPRESS_NAME

Format

COMPRESS_NAME *algorithm*

Description

When producing an encryption facility message, uses the specified algorithm for compression.

Note: If this command option is not set, the default algorithm preferences is used as defined in "Compression algorithm preference list" on page 10.

Default: If the recipient preference is not available in the OpenPGP certificate, ZIP.

Equivalent command option: "-compress-name — Specify the algorithm to use for compression" on page 56.

Arguments

For *algorithm*, specify an algorithm for compression. You can run the `-list-algo` command to see valid algorithm values.

S2K_CIPHER_NAME

Format

S2K_CIPHER_NAME *algorithm*

Description

When producing an encryption facility message, uses the specified algorithm for the passphrase-based encryption (PBE). PBE makes use of the hashing value and the encryption of the session key.

Default: If the recipient preference is not available in the OpenPGP certificate, TRIPLE_DES.

Equivalent command option: "-s2k-cipher-name — Specify the algorithm to use for passphrase-based encryption (PBE)" on page 67.

Arguments

For *algorithm*, specify an algorithm for PBE. You can run the `-list-algo` command to see valid algorithm values.

S2K_DIGEST_NAME

Format

S2K_DIGEST_NAME *algorithm*

Description

When producing an encryption facility message with passphrase-based encryption (PBE), uses the specified digest algorithm for password based encryption of the session key

Default: If the recipient preference is not available in the OpenPGP certificate, SHA_1.

Equivalent command option: "-s2k-digest-name — Specify the digest algorithm for passphrase-based encryption (PBE)" on page 68.

Arguments

For *algorithm*, specify a digest algorithm. You can run the `-list-algo` command to see valid algorithm values.

S2K_MODE

Format

S2K_MODE *value*

Description

When producing an encryption facility message with passphrase-based encryption (PBE), uses a specified value for hashing and encryption of the session key.

It is suggested that you specify 1 or 3. You can only use one password to encrypt the message.

Default: S2K_MODE **3**.

Equivalent command option: "-s2k-mode — Specify the mode for passphrase-based encryption (PBE)" on page 68.

Arguments

For *value*, one of the following:

0
Simple

1
Salted

3
Salted and iterated

S2K_PASSPHRASE

Format

S2K_PASSPHRASE *passphrase*

Description

When producing an encryption facility message, use the passphrase for passphrase-based encryption (PBE) or decryption.

Default: None. Prompts for passphrase. You need to run S2K_PASSPHRASE from the UNIX Systems Services environment.

Equivalent command option: "-s2k-passphrase — Specify the passphrase to use for passphrase-based encryption (PBE) and decryption" on page 69.

Arguments

For *passphrase*, specify a passphrase.

S2K_SUBKEY_PASSPHRASE

Format

S2K_SUBKEY_PASSPHRASE *passphrase*

Description

When you are producing an encryption facility message, use the passphrase for passphrase-based encryption (PBE) and decryption of a subkey.

Default: None. Prompts for *passphrase*. You need to run S2K_SUBKEY_PASSPHRASE from the UNIX Systems Services environment.

Equivalent command option: "-s2k-subkey-passphrase – Specify the passphrase to use for passphrase-based encryption (PBE) and decryption of a subkey" on page 69.

Arguments

For *passphrase*, specify a passphrase.

ANSWER_YES

Format

ANSWER_YES

Description

Assumes yes for yes/no questions.

Default: If not specified, prompt for any yes/no question. You need to run ANSWER_YES from the UNIX Systems Services environment.

Equivalent command option: "-yes — Specify yes to prompts" on page 74.

Arguments

None.

ANSWER_NO

Format

ANSWER_NO

Description

Assumes no for yes/no questions.

Default: If not specified, prompt for any yes/no question. You need to run ANSWER_NO from the UNIX Systems Services environment.

Equivalent command option: "-no — Specify no to prompts" on page 64.

Arguments

None

HIDDEN_PASSWORD

Format

HIDDEN_PASSWORD

Description

Does not display the password in response to the prompt. If you are using a TELNET 3270 session, Encryption Facility displays the password.

Default: if not specified, do not hide responses to password prompts. You need to run HIDDEN_PASSWORD from the UNIX Systems Services environment.

Equivalent command option: None.

Arguments

None.

RACF_KEYRING_USERID

Format

RACF_KEYRING_USERID *RACF-id*

Description

Specifies the RACF user ID to use when loading a RACF keyring.

It is suggested that you use this option. If it is not specified, the user ID under which Encryption Facility runs might NOT be used even if it is specified in the JCL.

Default: The RACF user ID that Encryption Facility runs under.

Equivalent command option: "-racf-keyring-userid — Specify a RACF user ID" on page 66.

Arguments

For *RACF-id*, the RACF user id.

USE_MDC

Format

USE_MDC

Description

While encrypting data for OpenPGP uses modification detection code (MDC), which specifies a symmetric integrity protected data packet.

Default: if not specified, do not set.

Equivalent command option: "-use-mdc — Specify the use of modification detection code" on page 72.

Arguments

None.

TRUST_VALUE

Format

TRUST_VALUE *number*

Description

Specifies the level of trust for an OpenPGP certificate.

Default: 10.

Equivalent command option: “-trust-value — Specify a trust value” on page 72.

Arguments

For *number*, you can specify a value from 0 to 255.

TRUSTED_COMMENT

Format

TRUSTED_COMMENT *text*

Description

Specifies a comment for the level of trust of an OpenPGP certificate.

Default: Trusted.

Equivalent command option: “-trusted-comment — Specify a trust comment” on page 72.

Arguments

For *text*, you can specify any comment string.

HARDWARE_KEY_TYPE

Format

HARDWARE_KEY_TYPE *type*

Description

Specifies the type of hardware key to generate.

PKDS keys are managed by ICSF.

Default: None. If nothing is specified, in a UNIX Systems Services environment, the user is prompted to enter a value.

Equivalent command option: None.

Arguments

For *type*, one of the following:

- PKDS
- CLEAR

For hardware type information, see *z/OS Cryptographic Services ICSF Application Programmer's Guide*.

BATCH_EXPORT

Format

```
BATCH_EXPORT
```

Description

Specifies batch public key export to enable batch mode processing for the export by alias `-eA` and export by key ID `-eK` commands. Batch mode processing is not enabled by default, and the `-eA` and `-eK` commands are interactive requiring you to respond to a series of command line prompts.

Batch mode processing for export allows you to specify all required command options and run the export command as a batch job without having to interact with the command line. In order to use this option, batch export processing requires that you specify the following options; otherwise, the request might fail:

- OUTPUT_FILE
- KEY_RING_FILENAME
- JAVA_KEY_STORE_TYPE
- JAVA_KEY_STORE_NAME
- KEYSTORE_PASSWORD
- KEY_PASSWORD
- ANSWER_YES
- ANSWER_NO
- RACF_KEYRING_USERID (if configured with a RACF keyring)
- USERID_NAME
- OPENPGP_DAYS_VALID.

Default: None.

Equivalent command option: "-batch-export — Specify batch public key export" on page 54.

Arguments

None.

BATCH_GENERATE

Format

BATCH_GENERATE *algorithm*,*number_of_multiples*

Description

Enables batch mode processing for key pair generation. Batch key generation is only supported with the JKS, JCEKS, and JCECCAKS Java Key Store types. Note that Encryption Facility OpenPGP only supports JCECCARACFKS and JCERACFKS RACF keyrings in read-only mode and does not support batch key generation with either of these key store types. Batch mode processing is not enabled by default, and key pair generation is an interactive command that requires you to respond to a series of command line prompts.

When you specify multiples (more than one key pair), a sequence number starting at 1 is appended to the key alias name, subkey alias name if applicable, and the user ID. The sequence number appended for multiples is used to prevent overwriting previously generated key material. In order to use this specification, batch key generation processing requires that you specify the following options; otherwise, the request might fail;

- KEY_RING_FILENAME
- JAVA_KEY_STORE_TYPE
- JAVA_KEY_STORE_NAME
- KEYSTORE_PASSWORD
- KEY_PASSWORD
- KEY_ALIAS
- KEY_SIZE,
- ANSWER_YES
- ANSWER_NO
- HARDWARE_KEY_TYPE (if configured with a JCECCAKS keystore)
- X509_DAYS_VALID
- USERID_NAME
- OPENPGP_DAYS_VALID
- SUB_KEY_ALIAS (if an ElGamal subkey has been requested with RSAELG or DSAELG).

Default: None.

Equivalent command option: "-batch-generate — Specify batch key generation" on page 54.

Arguments

For *algorithm*, an asymmetric algorithm or a combination of asymmetric algorithms. Valid asymmetric algorithms for this option include the following options:

RSA
For an RSA only key pair

DSA
For an DSA only key pair

RSAELG
For a combination of an RSA primary key pair and an ElGamal subkey key pair

DSAELG
For a combination of an DSA primary key pair and an ElGamal subkey key pair

Optionally, to request multiples, you can append the number of multiples, *number_of_multiples*, to the asymmetric algorithm or combination of asymmetric algorithms separated by a comma. The *number_of_multiples* argument is not required.

For example, to use this option to generate one RSA key pair, specify:

```
BATCH_GENERATE RSA
```

To use this option to generate two DSA key pairs, specify:

```
BATCH_GENERATE DSA,2
```

To use this option to generate three RSAELG key pairs, specify:

```
BATCH_GENERATE RSAELG,3
```

There are no default values for this option.

DN_COMMON_NAME

Format

DN_COMMON_NAME *string*

Description

Specifies the common name of a distinguished name (DN) for an X.509 certificate during key generation. If you do not specify this value, you are prompted for a value during key generation.

Default: None.

Equivalent command option: "-dn-common-name — Specify the common name of a distinguished name" on page 58.

Arguments

For *string*, the common name of a distinguished name (DN) for an X.509 certificate.

DN_COUNTRY_CODE

Format

DN_COUNTRY_CODE *string*

Description

Specifies the country code of a distinguished name (DN) for an X.509 certificate during key generation. If you do not specify this value, you are prompted for a value during key generation.

Default: None.

Equivalent command option: "-dn-country-code— Specify the country code of a distinguished name" on page 59.

Arguments

For *string*, the valid country code of a distinguished name (DN) for an X.509 certificate.

DN_LOCALITY

Format

DN_LOCALITY *string*

Description

Specifies the locality of a distinguished name (DN) for an X.509 certificate during key generation. If you do not specify this value, you are prompted for a value during key generation.

Default: None.

Equivalent command option: "-dn-locality — Specify the locality of a distinguished name" on page 59.

Arguments

For *string*, the locality of a distinguished name (DN) for an X.509 certificate.

DN_ORGANIZATION

Format

DN_ORGANIZATION *string*

Description

Specifies the organization of a distinguished name (DN) for an X.509 certificate during key generation. If you do not specify this value, you are prompted for a value during key generation.

Default: None.

Equivalent command option: "-dn-organization — Specify the organization of a distinguished name" on page 59.

Arguments

For *string*, the organization of a distinguished name (DN) for an X.509 certificate.

DN_ORGANIZATIONAL_UNIT

Format

DN_ORGANIZATIONAL_UNIT *string*

Description

Specifies the organization unit of a distinguished name (DN) for an X.509 certificate during key generation. If you do not specify this value, you are prompted for a value during key generation.

Default: None.

Equivalent command option: "-dn-organizational-unit — Specify the organization unit of a distinguished name" on page 59.

Arguments

For *string*, the organization unit of a distinguished name (DN) for an X.509 certificate.

DN_STATE

Format

DN_STATE *string*

Description

Specifies the state of a distinguished name (DN) for an X.509 certificate during key generation. If you do not specify this value, you are prompted for a value during key generation.

Default: None.

Equivalent command option: "-dn-state — Specify the state of a distinguished name" on page 60.

Arguments

For *string*, the state of a distinguished name (DN) for an X.509 certificate.

HIDDEN_KEY_ID

Format

HIDDEN_KEY_ID

Description

Supports speculative key IDs as described by RFC 4880 during public key encryption and decryption. To use this support during public key encryption (-e), you must either specify the option on the command line or in the IBM Encryption Facility configuration file *ibmef.config*.

No command options are required to use speculative key ID support on the decrypt command (-d). Encryption Facility automatically detects and performs speculative key ID support when decrypting OpenPGP messages that contain hidden key ID values. This value activates speculative key ID support when encrypting OpenPGP messages. Speculative key ID support hides the key ID values in encrypted messages by using a value of zero in the Public-Key Encrypted Session Key Packets for the key ID field. This option is not set by default, and encrypted messages contain the key IDs of the public keys that are used to wrap the session key. When key IDs are included in the encrypted message, an implementation can easily find the associated private key. The private key is then used to unwrap the session key and decrypt the message.

When you use speculative key ID support, the Public-Key Encrypted Session Key Packets in the Encrypted Message contain zeroed-out key IDs fields. This implementation removes the risk that the key IDs can be being intercepted by an unauthorized user. When the key IDs are zeroed out, an implementation must try to decrypt the message with all available private keys in order to determine the correct private key that is able to successfully decrypt the message.

When the decrypt command discovers a hidden Key ID with the value zero, Encryption Facility attempts to decrypt the session key by using all available private keys, until a valid session key is retrieved. If a valid session key is retrieved, Encryption Facility encrypts the encrypted data packet with the session key. If a valid session key is not found, Encryption Facility fails with the following message:

```
CSD0600I Could not find a valid session key to decrypt the message.
```

Default: None.

Equivalent command option: "-hidden-key-id — Specify speculative key ID support" on page 60.

Arguments

None.

OPENPGP_DAYS_VALID

Format

OPENPGP_DAYS_VALID *days*

Description

Specifies how many days a newly generated OpenPGP certificate is to be valid. If you do not specify this value, you are prompted for a value during key generation, key export by alias, and key export by key ID. When you are performing batch key generation or batch export, this option is required, and the request fails if you do not specify this option.

Default: None.

Equivalent command option: "-openPGP-days-valid — Specify the number of days a newly generated OpenPGP certificate is to be valid" on page 65.

Arguments

For *days*, the number of days this OpenPGP certificate is to be valid. The range for *days* is a number from 0 to 2147483647, where 0 indicates that the OpenPGP certificate never expires. The minimum number of days is 1, and the maximum number of days is 2147483647.

SUB_KEY_ALIAS

Format

SUB_KEY_ALIAS *alias*

Description

Specifies the alias to be used when generating a new subkey during key generation. If you do not specify this value, you are prompted for a value during key generation when an ElGamal subkey has been requested. When you are performing batch key generation with an ElGamal subkey, this option is required, and the request fails if you do not specify this option.

Default: None.

Equivalent command option: "-sub-key-alias — Specify the alias for a new subkey during key generation" on page 70.

Arguments

For *alias*, the alias to use for the ElGamal subkey to be generated. No default.

USERID_COMMENT

Format

USERID_COMMENT *string*

Description

Specifies a user ID comment for an OpenPGP certificate during key generation and key export. If you do not specify this option, you are prompted for a value during key generation, key export by alias, and key export by key ID.

Default: None.

Equivalent command option: "-userID-comment — Specify a user ID comment for an OpenPGP certificate during key generation and key export" on page 73.

Arguments

For *string*, a comment to use with the user ID for the OpenPGP certificate. The comment section of an OpenPGP certificate user ID is optional.

USERID_EMAIL

Format

USERID_EMAIL *string*

Description

Specifies a user ID email address for an OpenPGP certificate during key generation and key export. If you do not specify this option, you are prompted for a value during key generation, key export by alias, and key export by key ID.

Default: None.

Equivalent command option: "-userID-email — Specify a user ID email address for an OpenPGP certificate during key generation and key export." on page 73.

Arguments

For *string*, the user ID email address to use with the OpenPGP certificate.

USERID_NAME

Format

USERID_NAME *name*

Description

Specifies a user ID for an OpenPGP certificate during key generation and key export. If you do not specify this option, you are prompted for a value during key generation, key export by alias, and key export by key ID. When you are performing batch key generation or batch export, this option is required, and the request fails if you do not specify this option.

Default: None.

Equivalent command option: "-userID-name — Specify a user ID for an OpenPGP certificate during key generation and key export" on page 73.

Arguments

For *name*, the user ID name to use with the OpenPGP certificate.

X509_DAYS_VALID

Format

X509_DAYS_VALID *days*

Description

Specifies how many days a newly generated X.509 certificate should be valid when you use the generate command *-g*. If you do not specify this value, you are prompted for a value during key generation. When you are performing batch key generation this option is required, and the request fails if you do not specify the option.

Default: None.

Equivalent command option: "-x509-days-valid — Specify the number of days an X509 certificate is to be valid " on page 74.

Arguments

For *days*, the number of days this X.509 certificate is to be valid. The range for *days* is a number from 1 to 9999, where 1 is the minimum number of days and 9999 is the maximum number of days allowed for an X.509 certificate to be valid.

EXPORT_SECRET_KEY

Format

EXPORT_SECRET_KEY

Description

Exports the secret key of the specified OpenPGP certificate along with the public key when export commands are given.

Default: None.

Equivalent command option: "-export-secret-key — Enable exporting of secret key when exporting an OpenPGP Certificate" on page 60.

Arguments

None.

PRESERVE_DATASET_RECORD_LENGTH

Format

```
PRESERVE_DATASET_RECORD_LENGTH
```

Description

Used to preserve the record structure when encrypting and decrypting variable length formatted data sets.

Default: Disabled, dataset record lengths are not stored along with the record data in the encrypted form.

Equivalent command option: "-preserve-reclen — Preserve dataset record length" on page 66.

Arguments

None.

ACCEPT_UNRECOGNIZED_SUBPACKETS

Format

```
ACCEPT_UNRECOGNIZED_SUBPACKETS
```

Description

Specifies that Encryption Facility processes, ignores, and retains any unrecognized signature subpackets.

Default: None.

Equivalent command option: "-accept-unrecognized — Accept unrecognized subpackets" on page 54.

Arguments

None.

S2K_COUNT

Format

```
S2K_COUNT
```

Description

Used to allow users to specify the count value in an Iterated and Salted S2K password-based encryption process..

Default: None.

Equivalent command option: “-s2k-count — Specify the count for passphrase-based encryption (PBE)” on page 68.

Arguments

None.

IGNORE_TEXTMODE

Format

IGNORE_TEXTMODE

Description

Used to prevent the conversion of the output data to the system's character set.

Default: None.

Equivalent command option: “-ignore-textmode — Prevent conversion of output data to the system's character set” on page 61.

Arguments

None.

Latest command options and the updated ibmef.config file

You can specify the latest command options on the command line or in the IBM EF configuration file, `ibmef.config`. A new `ibmef.config` file will not be shipped in the service stream for this update as this could potentially overwrite pre-configured user settings for a given Encryption Facility installation. You can copy and paste this section to the end of an existing `ibmef.config` file. When this new section is appended, you can turn on the new command options by "uncommenting" them and following the instructions described in section. For command details, see “Encryption Facility for OpenPGP options and commands” on page 52.

The updated `ibmef.config` has the latest command options:

```
#------------------------------------------------------------------------------------------------------------
#This value turns on speculative key ID support for encrypted messages. Speculative key ID support hides the
#key ID values in encrypted messages by using a value of zero in the Public-Key Encrypted Session Key Packets
#for the key ID field.
#By default encrypted messages contain the key IDs of the public keys used to wrap the session key.
#When Key IDs are included in the encrypted message, an implementation can easily find the associated private
#key.  The private key is then used to unwrap the session key and decrypt the message.
#When speculative key ID support is used, the Public-Key Encrypted Session Key Packets in the Encrypted
#Message will contain zeroed out key IDs fields.
#Zeroing out the key IDs removes the risk of the key IDs being intercepted by an unauthorized user.
#When the key IDs are zeroed out, an implementation must try to decrypt the message with all available
private
#keys in order to determine the correct private key that will be able to successfully decrypt the message.
#
#Default: none, key IDs are written in the Public-Key Encrypted Session Key Packets.
#HIDDEN_KEY_ID
#------------------------------------------------------------------------------------------------------------

#------------------------------------------------------------------------------------------------------------
#This value enables batch mode processing for key pair generation.
#Batch key generation is only supported with the JKS, JCEKS, and JCECCAKS Java Key Store types. Encryption
#Facility OpenPGP only supports JCECCARACFKS and JCERACFKS RACF keyrings in Read Only mode and does not
#support batch key generation with these key store types.
#By default batch mode processing is not enabled and key pair generation is an interactive command that
#requires the user to respond to a series of command line prompts.
#Batch mode processing for key pair generation allows the user to specify all required command options and
#execute the generate command, -g, as a batch job without having to interact with the command line.
#Additionally, multiples may be specified for a given key pair generation request. This support allows for
#more than 1 key pair to be generated at a time using the same command options. When multiples are specified
#a sequence number starting at 1 is appended to the key alias name, subkey alias name if applicable, and the
#user ID. The sequence number appended for multiples is used to prevent overwriting previously generated key
#material.
#
#In order to use this option, batch key generation processing requires that the following options are also
```

```
#specified, otherwise the request may fail; KEY_RING_FILENAME, JAVA_KEY_STORE_TYPE, JAVA_KEY_STORE_NAME,
#KEYSTORE_PASSWORD, KEY_PASSWORD, KEY_ALIAS, KEY_SIZE, (ANSWER_YES or ANSWER_NO),
#HARDWARE_KEY_TYPE (If configured with a JCECCAKS keystore), X509_DAYS_VALID,
#USERID_NAME, OPENPGP_DAYS_VALID, (SUB_KEY_ALIAS if an ElGamal subkey has been requested
#with RSAELG or DSAELG)
#
#This option requires an asymmetric algorithm, or a combination of asymmetric algorithms to be
#specified. Valid asymmetric algorithms that can be used with this option include:
#  RSA    - For an RSA only key pair
#  DSA    - For an DSA only key pair
#  RSAELG - For a combination of an RSA primary key pair and an ElGamal subkey key pair
#  DSAELG - For a combination of an DSA primary key pair and an ElGamal subkey key pair
#
#Optionally, to request multiples, the asymmetric algorithm or combination of asymmetric algorithms may be
#followed by a comma, and then the number of multiples.
#
#For example:
#To use this option to generate 1 RSA key pair, specify: BATCH_GENERATE RSA
#To use this option to generate 2 DSA key pairs, specify: BATCH_GENERATE DSA,2
#To use this option to generate 3 RSAELG key pairs, specify: BATCH_GENERATE RSAELG,3
#
#Default: none, batch mode for key pair generation is not enabled.
#BATCH_GENERATE RSA
#------------------------------------------------------------------------------------------------------------
#------------------------------------------------------------------------------------------------------------
#This value enables batch mode processing for the export by alias, -eA, and export by Key ID, -eK, commands.
#By default batch mode processing is not enabled and the export by alias and export by Key ID commands are
#interactive and require the user to respond to a series of command line prompts.
#Batch mode processing for export allows the user to specify all required command options and execute the
#export command as a batch job without having to interact with the command line.
#This option does not take any arguments.
#
#In order to use this option, batch export processing requires that the following options are also specified,
#otherwise the request may fail; OUTPUT_FILE, KEY_RING_FILENAME, JAVA_KEY_STORE_TYPE, JAVA_KEY_STORE_NAME,
#KEYSTORE_PASSWORD, KEY_PASSWORD, (ANSWER_YES or ANSWER_NO), RACF_KEYRING_USERID (If configured with a RACF
#keyring), USERID_NAME, OPENPGP_DAYS_VALID.
#
#Default: none, batch mode for export is not enabled.
#BATCH_EXPORT
#------------------------------------------------------------------------------------------------------------

#------------------------------------------------------------------------------------------------------------
#This value specifies how many days a newly generated X.509 certificate should be valid for when using the
#generate command, -g.
#The minimum days allowed for an X.509 certificate to be valid is 1.
#The maximum days allowed for an X.509 certificate to be valid is 9999.
#If this value is not specified the user will be prompted for a value during key generation.
#When performing batch key generation this option is required, and the request will fail if this option is not
#specified.
#
#Default: none
#X509_DAYS_VALID 9999
#------------------------------------------------------------------------------------------------------------

#------------------------------------------------------------------------------------------------------------
#This value specifies how many days a newly generated OpenPGP certificate should be valid for.
#The minimum value allowed for an OpenPGP certificate to be valid is 0.
#0 Indicates that the OpenPGP certificate will never expire.
#The minimum days allowed for an OpenPGP certificate to be valid is 1.
#The maximum days allowed for an OpenPGP certificate to be valid is 2147483647.
#If this value is not specified the user will be prompted for a value during key generation, key export by
#alias, and key export by key ID.
#When performing batch key generation or batch export, this option is required, and the request will fail if
#this option is not specified.
#
#Default: none
#OPENPGP_DAYS_VALID 0
#------------------------------------------------------------------------------------------------------------


#------------------------------------------------------------------------------------------------------------
#This value is used to specify the Common Name of a Distinguished Name (DN) for an X.509 certificate during
#key generation.
#If this value is not specified the user will be prompted for a value during key generation.
#
#Default: none
#DN_COMMON_NAME
#------------------------------------------------------------------------------------------------------------
#------------------------------------------------------------------------------------------------------------

#This value is used to specify the Organizational Unit of a Distinguished Name (DN) for an X.509 certificate
#during key generation.
#If this value is not specified the user will be prompted for a value during key generation.
#
#Default: none
#DN_ORGANIZATIONAL_UNIT
#------------------------------------------------------------------------------------------------------------
#------------------------------------------------------------------------------------------------------------

#This value is used to specify the Organization of a Distinguished Name (DN) for an X.509 certificate during
#key generation.
#If this value is not specified the user will be prompted for a value during key generation.
#
```

```
#Default: none
#DN_ORGANIZATION
#-------------------------------------------------------------------------------------------------------------

#-------------------------------------------------------------------------------------------------------------

#This value is used to specify the Locality of a Distinguished Name (DN) for an X.509 certificate during key
#generation.
#If this value is not specified the user will be prompted for a value during key generation.
#
#Default: none
#DN_LOCALITY
#-------------------------------------------------------------------------------------------------------------

#-------------------------------------------------------------------------------------------------------------

#This value is used to specify the State of a Distinguished Name (DN) for an X.509 certificate during key
#generation.
#If this value is not specified the user will be prompted for a value during key generation.
#
#Default: none
#DN_STATE
#-------------------------------------------------------------------------------------------------------------

#-------------------------------------------------------------------------------------------------------------
#This value is used to specify the Country Code of a Distinguished Name (DN) for an X.509 certificate during
#key generation.
#If this value is not specified the user will be prompted for a value during key generation.
#
#Default: none
#DN_COUNTRY_CODE
#-------------------------------------------------------------------------------------------------------------

#-------------------------------------------------------------------------------------------------------------
#This value is used to specify a User ID for an OpenPGP certificate during key generation and key export.
#If this value is not specified the user will be prompted for a value during key generation, key export by
#alias, and key export by key ID.
#When performing batch key generation or batch export, this option is required, and the request will fail if
#this option is not specified.
#
#Default: none
#USERID_NAME
#-------------------------------------------------------------------------------------------------------------


#-------------------------------------------------------------------------------------------------------------

#This value is used to specify a User ID Comment for an OpenPGP certificate during key generation and key
#export.
#If this value is not specified the user will be prompted for a value during key generation, key export by
#alias, and key export by key ID.
#
#Default: none
#USERID_COMMENT
#-------------------------------------------------------------------------------------------------------------
#-------------------------------------------------------------------------------------------------------------
#This value is used to specify a User ID Email address for an OpenPGP certificate during key generation and
#key export.
#If this value is not specified the user will be prompted for a value during key generation, key export by
#alias, and key export by key ID.
#
#Default: none
#USERID_EMAIL
#-------------------------------------------------------------------------------------------------------------

#-------------------------------------------------------------------------------------------------------------
#This value specifies the alias to be used when generating a new subkey during key generation.
#If this value is not specified the user will be prompted for a value during key generation when an ElGamal
#subkey has been requested.
#When performing batch key generation with an ElGamal subkey, this option is required, and the request will
#fail if this option is not specified.
#
#Default: none
#SUB_KEY_ALIAS
#-------------------------------------------------------------------------------------------------------------
```

Encryption Facility for OpenPGP options and commands

All Encryption Facility for OpenPGP commands have the following syntax. `-homedir` must appear before all the options, and all the options must appear before the commands:

```
com.ibm.encryptionfacility.EFOpenPGP[-homedir name]|[options] commands [arguments]
```

where:

- *homedir name* is the name of the configuration file `ibmef.config` that contains specified options to use with the command.
- *options* is the name of one or more options to use on the command line and always starts with `-`. This option value overrides values in the configuration file. See the "Command options" on page 53.
- *commands* is the name of one or more commands and always starts with `-`.
- *arguments* specifies one or more targets of the command, for example, file name, certificate, alias, and so forth.

File names can be fully qualified names of preallocated z/OS data sets or DD names. For example, to specify the fully-qualified z/OS data set SYS1.TEXT.MESSAGE or the DD statement for SYSUT1, be sure to use two forward slashes when you specify the names:

```
//SYS1.TEXT.MESSAGE
//DD:SYSUT1
```

For PSE data sets, be sure to enclose within single quotation marks:

```
'//SYS.TEST.PDS(MEMBER1)'
```

When you use a DD statement in the JCL to allocate the data set, be sure to specify the DD name instead of the data set name and enclose it in quotations. For example, for a data set specified on the DD statement labeled DDDEF:

```
 '//DD:DDDEF'
```

See "Allocating data sets through the data definition (DD) statement" on page 20 and "Sample JCL and code" on page 179.

Command options

Each command option includes format, description, and any arguments and default values for Encryption Facility for OpenPGP commands. Any option that you specify on a command overrides the value in the configuration file. The following options are arranged alphabetically.

-a — Use ASCII Armor for the OpenPGP certificate output

Format

-a

Description

Specifies that when you export an OpenPGP certificate, you are to use ASCII Armor. For a z/OS data set that is protected by ASCII armor, the data must be in EBCDIC, not ASCII format. This option only applies to the export commands (-eA, -eK, or -eP). If used with any other command, this option is ignored. Encryption Facility will not output encrypted data or signatures in ASCII Armor.

Default: If not specified, do not use ASCII armor.

Configuration file option: USE_ASCII_ARMOR

Arguments

None.

-accept-unrecognized — Accept unrecognized subpackets

Format

-accept-unrecognized

Description

Used to specify that Encryption Facility processes, ignores, and retains any unrecognized signature subpackets.

Configuration file option: "ACCEPT_UNRECOGNIZED_SUBPACKETS" on page 49.

Arguments

None.

-batch-export — Specify batch public key export

Format

```
-batch-export
```

Description

This option enables batch mode processing for the export by alias -eA and export by key ID -eK commands. Batch mode processing is not enabled by default, and the -eA and -eK commands are interactive requiring you to respond to a series of command line prompts.

Batch mode processing for export allows you to specify all required command options and run the export command as a batch job without having to interact with the command line. In order to use this option, batch export processing requires that you specify the following options; otherwise, the request might fail:

- OUTPUT_FILE
- KEY_RING_FILENAME
- JAVA_KEY_STORE_TYPE
- JAVA_KEY_STORE_NAME
- KEYSTORE_PASSWORD
- KEY_PASSWORD
- ANSWER_YES
- ANSWER_NO
- RACF_KEYRING_USERID (if configured with a RACF keyring)
- USERID_NAME
- OPENPGP_DAYS_VALID.

Configuration file option: BATCH EXPORT

Arguments

None.

-batch-generate — Specify batch key generation

Format

-batch-generate *algorithm*,*number_of_multiples*

Description

This option enables batch mode processing for key pair generation. Batch key generation is only supported with the JKS, JCEKS, and JCECCAKS Java Key Store types. Note that Encryption Facility OpenPGP only supports JCECCARACFKS and JCERACFKS RACF keyrings in read-only mode and does not support batch key generation with either of these key store types. Batch mode processing is not enabled by default, and key pair generation is an interactive command that requires you to respond to a series of command line prompts.

Batch mode processing for key pair generation allows you to specify all required command options and execute the generate command `-g` as a batch job without having to interact with the command line. Additionally, you can specify multiples for a given key pair generation request. This support allows for more than one key pair to be generated at a time using the same command options.

When you specify multiples (more than one key pair), a sequence number starting at 1 is appended to the key alias name, subkey alias name if applicable, and the user ID. The sequence number appended for multiples is used to prevent overwriting previously generated key material. In order to use this option, batch key generation processing requires that you specify the following options; otherwise, the request might fail;

- KEY_RING_FILENAME
- JAVA_KEY_STORE_TYPE
- JAVA_KEY_STORE_NAME
- KEYSTORE_PASSWORD
- KEY_PASSWORD
- KEY_ALIAS
- KEY_SIZE,
- ANSWER_YES
- ANSWER_NO
- HARDWARE_KEY_TYPE (if configured with a JCECCAKS keystore)
- X509_DAYS_VALID
- USERID_NAME
- OPENPGP_DAYS_VALID
- SUB_KEY_ALIAS (if an ElGamal subkey has been requested with RSAELG or DSAELG).

Configuration file option: BATCH GENERATE

Arguments

For *algorithm*, an asymmetric algorithm or a combination of asymmetric algorithms. Valid asymmetric algorithms for this option include the following options:

RSA
For an RSA only key pair

DSA
For an DSA only key pair

RSAELG
For a combination of an RSA primary key pair and an ElGamal subkey key pair

DSAELG
For a combination of an DSA primary key pair and an ElGamal subkey key pair

Optionally, to request multiples, you can append the number of multiples, *number_of_multiples*, to the asymmetric algorithm or combination of asymmetric algorithms separated by a comma. The *number_of_multiples* argument is not required.

For example, to use this option to generate one RSA key pair, specify:

```
BATCH_GENERATE RSA
```

To use this option to generate two DSA key pairs, specify:

```
BATCH_GENERATE DSA,2
```

To use this option to generate three RSAELG key pairs, specify:

```
BATCH_GENERATE RSAELG,3
```

There are no default values for this option.

-cipher-name — Specify the algorithm for encryption

Format

-cipher-name *name*

Description

This option establishes the cipher algorithm to use for encryption. The value takes precedence over preferences established in a partner's OpenPGP certificate.

Note: If this command option is not set, the default algorithm preferences is used as defined in "Symmetric algorithm preference list" on page 9.

Configuration file option: CIPHER_NAME

Arguments

For *name*, the name of the cipher algorithm. The `-list-algo` command lists all the available algorithms.

Default is TRIPLE_DES.

-comment — Add a comment header to ASCII Armorized messages

Format

-comment *string*

Description

When generating an ASCII "armorized" message, this option adds the comment header with the specified string *string*.

Configuration file option: ARMOR_COMMENT

Arguments

For *string*, the comment header to add. No default.

-compress-name — Specify the algorithm to use for compression

Format

-compress-name *name*

Description

This option establishes the compression algorithm to use during message construction. This value takes precedence over preferences established in a partner’s OpenPGP certificate.

Note: If this command option is not set, the default algorithm preferences is used as defined in “Compression algorithm preference list” on page 10.

Configuration file option: COMPRESS_NAME

Arguments

For *name*, the name of compression algorithm. The **-list-algo** command lists all the available algorithms.

The default varies depending on the following conditions:

- If multiple recipients are specified, the preference contained in a partner’s OpenPGP certificate. (Encryption Facility for OpenPGP uses the preference of the first recipient.)
- Otherwise, ZIP.

-debug-level *level* — Specify a level for trace information to be sent to the log file

Format

```
-debug-level level
```

Description

This option establishes the granularity of debug information.

Configuration file option: “DEBUG_LEVEL” on page 30

Arguments

For *level*, an integer that sets the amount of trace information to be sent to the log file, the STDERR, or both. The default is 0. For a list of valid levels, see the DEBUG_LEVEL command's “Arguments” on page 30.

-debug *number*— Specify a bit mask value for logging

Format

```
-debug number
```

Description

This option activates logging for components.

Configuration file option: ACTIVE_LOGGERS

Arguments

For *number*, the bit mask value that turns on logging for components. The default is 0.

-debug-on — Activate debugging information

Format

-debug-on

Description

This option activates debugging information printed to STDERR while executing.

Configuration file option: CREATE_TRACE

Arguments

None.

-digest-name — Specify the algorithm for the message digest

Format

-digest-name *name*

Description

This option establishes the digest algorithm to use during message digest calculation. The value takes precedence over preferences established in a partner's OpenPGP certificate.

Note: If this command option is not set, the default algorithm preferences is used as defined in "Message digest algorithm preference list" on page 9.

Configuration file option: DIGEST_NAME

Arguments

For *name*, the name of the digest algorithm. The `-list-algo` command lists all the available algorithms.

The default varies depending on the following conditions:

- If multiple recipients are specified, the preference contained in a partner's OpenPGP certificate. (Encryption Facility for OpenPGP uses the preference of the first recipient.)
- Otherwise, SHA_1.

-dn-common-name — Specify the common name of a distinguished name

Format

-dn-common-name *string*

Description

This option is used to specify the common name of a distinguished name (DN) for an X.509 certificate during key generation. If you do not specify this value, you are prompted for a value during key generation.

Configuration file option: "DN_COMMON_NAME" on page 43

Arguments

For *string*, the common name of a distinguished name (DN) for an X.509 certificate. No default.

-dn-country-code— Specify the country code of a distinguished name

Format

-dn-country-code *string*

Description

This option is used to specify the country code of a distinguished name (DN) for an X.509 certificate during key generation. If you do not specify this value, you are prompted for a value during key generation.

Configuration file option: "DN_COUNTRY_CODE " on page 44

Arguments

For *string*, the valid country code of a distinguished name (DN) for an X.509 certificate. No default.

-dn-locality — Specify the locality of a distinguished name

Format

-dn-locality *string*

Description

This option is used to specify the locality of a distinguished name (DN) for an X.509 certificate during key generation. If you do not specify this value, you are prompted for a value during key generation.

Configuration file option: "DN_LOCALITY" on page 44

Arguments

For *string*, the locality of a distinguished name (DN) for an X.509 certificate. No default.

-dn-organization — Specify the organization of a distinguished name

Format

-dn-organization *string*

Description

This option is used to specify the organization of a distinguished name (DN) for an X.509 certificate during key generation. If you do not specify this value, you are prompted for a value during key generation.

Configuration file option: "DN_ORGANIZATION" on page 44

Arguments

For *string*, the organization of a distinguished name (DN) for an X.509 certificate. No default.

-dn-organizational-unit — Specify the organization unit of a distinguished name

Format

-dn-organizational-unit *string*

Description

This option is used to specify the organization unit of a distinguished name (DN) for an X.509 certificate during key generation. If you do not specify this value, you are prompted for a value during key generation.

Configuration file option: “DN_ORGANIZATIONAL_UNIT” on page 45

Arguments

For *string*, the organization unit of a distinguished name (DN) for an X.509 certificate. No default.

-dn-state — Specify the state of a distinguished name

Format

-dn-state *string*

Description

This option is used to specify the state of a distinguished name (DN) for an X.509 certificate during key generation. If you do not specify this value, you are prompted for a value during key generation.

Configuration file option: “DN_STATE” on page 45

Arguments

For *string*, the state of a distinguished name (DN) for an X.509 certificate. No default.

-export-secret-key — Enable exporting of secret key when exporting an OpenPGP Certificate

Format

-export-secret-key

Description

When specified as an option for one of the export commands, -eP, -eA, or -eK, the secret key corresponding to the exported OpenPGP Certificate is also exported.

Configuration file option: “EXPORT_SECRET_KEY” on page 48.

Arguments

None.

-hidden-key-id — Specify speculative key ID support

Format

-hidden-key-id

Description

The option is used to support speculative key IDs as described by RFC 4880 during public key encryption and decryption. To use this support during public key encryption (-e), you must either specify the option on the command line or in the IBM Encryption Facility configuration file *ibmef.config*.

No command options are required to use speculative key ID support on the decrypt command (-d). Encryption Facility automatically detects and performs speculative key ID support when decrypting OpenPGP messages that contain hidden key ID values. This value activates speculative key ID support when encrypting OpenPGP messages. Speculative key ID support hides the key ID values in encrypted messages by using a value of zero in the Public-Key Encrypted Session Key Packets for the key ID field. This option is not set by default, and encrypted messages contain the key IDs of the public keys that are used to wrap the session key. When key IDs are included in the encrypted message, an implementation can easily find the associated private key. The private key is then used to unwrap the session key and decrypt the message.

When you use speculative key ID support, the Public-Key Encrypted Session Key Packets in the Encrypted Message contain zeroed-out key IDs fields. This implementation removes the risk that the key IDs can be being intercepted by an unauthorized user. When the key IDs are zeroed out, an implementation must try to decrypt the message with all available private keys in order to determine the correct private key that is able to successfully decrypt the message.

When the decrypt command discovers a hidden Key ID with the value zero, Encryption Facility attempts to decrypt the session key by using all available private keys, until a valid session key is retrieved. If a valid session key is retrieved, Encryption Facility encrypts the encrypted data packet with the session key. If a valid session key is not found, Encryption Facility fails with the following message:

```
CSD0600I Could not find a valid session key to decrypt the message.
```

Arguments

None.

Configuration file option: “HIDDEN_KEY_ID” on page 45

-ignore-textmode — Prevent conversion of output data to the system's character set

Format

-ignore-textmode

Description

Prevents the conversion of output data to the system's character set.

Configuration file option: “IGNORE_TEXTMODE” on page 50.

Arguments

None.

-jce-providers — Specify JCE class names

Format

-jce-providers *provider1*, *provider2*,..., *providerN*

Description

This option specifies a comma-delimited list of JCE provider fully-qualified class names.

Arguments

For *providerN*, the comma-separated list of providers that is prefixed to the provider list specified in `$JAVA_HOME/lib/security/java.security`, where N is the sequence number in the list. For hardware cryptography, the value is as follows:

```
com.ibm.crypto.hdwrCCA.provider.IBMJCECCA
```

```

```

The default is the provider list specified in `$JAVA_HOME/lib/security/java.security`.

Configuration file option: JCE_PROVIDER_LIST

-key-alias — Specify the alias of a new key

Format

-key-alias *alias*

Description

This option specifies the alias for a newly generated key within the Java keystore.

Configuration file option: **KEY_ALIAS**

Arguments

For *alias*, the alias of the generated key. No default.

-key-password — Specify the password for a new key

Format

-key-password *password*

Description

This option specifies the password for a newly generated key within the Java keystore.

Configuration file option: KEY_PASSWORD

Arguments

For *password*, the password for the generated key. No default.

-key-size — Specify the key size to generate

Format

-key-size *value*

Description

This option specifies the size for a newly generated key within the Java keystore. Key sizes depend on the hardware and software you are using. See "Java algorithm support for Encryption Facility for OpenPGP" on page 6.

Configuration file option: KEY_SIZE

Arguments

For *value*, the key size to generate. The default is **1024**.

-keystore — Specify the name of the Java keystore

Format

-keystore *name*

Description

This option specifies the name of the Java keystore file or keyring.

Configuration file option: JAVA_KEY_STORE_NAME

Arguments

For *name*, if the value is a RACF-type keystore, this value must be the RACF keyring name. Otherwise, *name* must be a UNIX System Services file name. No default.

-keystore-password — Specify the keystore password

Format

-keystore-password *password*

Description

This option specifies the password for the keystore.

Configuration file option: KEYSTORE_PASSWORD

Arguments

For *password*, the key-store password. No default.

-keystore-type — Specify the keystore type

Format

-keystore *type*

Description

This option specifies the type of keystore.

Configuration file option: JAVA_KEY_STORE_TYPE

Arguments

For *type*, one of the following keystore types:

- JKS
- JCEKS
- JCECCAKS
- JCECCARACFKS
- JCERACFKS

No default. If you specify the type as JCEKS, Encryption Facility converts the keystore from a JKS to a JCEKS keystore. If you use the hardware provider to generate keys, you must use the JCECCAKS keystore type.

-log-file — Write trace information to a file

Format

-log-file *filename*

Description

This option specifies the a UNIX System Services file name *filename* where write trace information is to be written. The system writes traces in XML.

Configuration file option: LOG_FILE

Arguments

For *filename* the name of the file for output trace XML information. No default.

-no — Specify no to prompts

Format

-no

Description

This option specifies an answer of **no** to most interactive questions.

Configuration file option: ANSWER_NO

Arguments

None.

-no-save — Display data to STDOUT only

Format

-no-save

Description

When unpacking a message, this option displays the data only to STDOUT and does not save the data to a file. When packaging an OpenPGP message, the command labels the data as "confidential." Receiving clients should then display the contents to STDOUT only.

This option is mutually exclusive with the -o option.

Configuration file option: CONFIDENTIAL

Arguments

None.

-o — Specify an output location

Format

-o *file*

Description

This option indicates the file system destination of any command. This option is required when a command produces an OpenPGP message or extracts data from an OpenPGP message. The **-o** option is not required when **-no-save** or -**use-embeded-file** is specified. Rewriting to the existing file must be confirmed. You can also use this option to specify a pre-allocated sequential data set, PDS, or PDSE. (See the **–no-save** and -use-embeded-file options.)

Configuration file option: OUTPUT

Arguments

For *file*, filename of output. No default output file.

For data sets, you must preallocate the data set and specify a prefix of //. For example:

```
//U1.HIGHRISK.EF.OUTPUT
```

For PDSE data sets, you must enclose the name in single quotations. For example:

```
'//SYS.TEST.PDS(MEMBER1)’
```

When you use a DD statement in the JCL to allocate the data set, be sure to specify the DD name instead of the data set name and enclose it in quotations. For example, for a data set specified on the DD statement labeled DDDEF:

```
 -o '//DD:DDDEF'
```

See “Sample JCL and code” on page 179.

-openPGP-days-valid — Specify the number of days a newly generated OpenPGP certificate is to be valid

Format

-openPGP-days-valid *days*

Description

This option specifies how many days a newly generated OpenPGP certificate is to be valid. If you do not specify this value, you are prompted for a value during key generation, key export by alias, and key export by key ID. When you are performing batch key generation or batch export, this option is required, and the request fails if you do not specify this option.

Configuration file option: “OPENPGP_DAYS_VALID” on page 46

Arguments

For *days*, the number of days this OpenPGP certificate is to be valid. The range for *days* is a number from 0 to 2147483647, where 0 indicates that the OpenPGP certificate never expires. The minimum number of days is 1, and the maximum number of days is 2147483647. No default.

-preserve-reclen — Preserve dataset record length

Format

-preserve-reclen

Description

Used to preserve the record structure when encrypting and decrypting variable length formatted data sets.

Configuration file option: “PRESERVE_DATASET_RECORD_LENGTH” on page 49.

Arguments

None.

-rA — Encrypt using the public key from the Java keystore

Format

-rA *public_key_ alias1,public_key_ alias2,... public_key_ aliasN*

Description

This option performs encryption using the public key in the Java keystore whose alias is the argument value *public_key_ aliasN*.

If you specify multiple `-rK` command invocations, Encryption Facility collects all of the recipient entries into a final list.

Configuration file option: RECIPIENT_ALIAS

Arguments

For *public_key_ aliasN*, a comma-separated list of aliases of the public key to use for encryption, where N is the sequence number in the list. These values will be used in conjunction with `-rK` and `-rP` values. No default.

-racf-keyring-userid — Specify a RACF user ID

Format

-racf-keyring-userid *userid*

Description

This option specifies a valid RACF user ID.

Configuration file option: RACF_KEYRING_USERID

Arguments

For *userid*, the RACF user ID. No default.

-rK — Encrypt for a specified key ID

Format

-rK *recipients_Key_ID1, recipients_Key_ID2*,... *recipients_Key_IDN*

Description

This option performs encryption using one or more public keys whose key IDs match the option values. The command uses the key ID to search the OpenPGP keyring file and keystore for a match.

If you specify multiple `-rK` command invocations, Encryption Facility collects all of the recipient entries into a final list.

Configuration file option: RECIPIENT_KEY_ID

Arguments

For *recipients_Key_IDN*, a comma-separated list of each recipient user ID, where N is the sequence number in the list. These values are used in conjunction with `-rP` and `-rA` values. No default user ID.

-rP — Encrypt for a specified user ID

Format

-rP *recipients_user_ID1*, *recipients_user_ID2*,... *recipients_user_IDN*

Description

This option performs encryption for one or more specified user IDs. It uses the user ID to search the OpenPGP keyring file for a match.

If you specify multiple `-rP` command invocations, Encryption Facility collects all of the recipient entries into a final list.

Configuration file option: RECIPIENT_USER_ID

Arguments

For *recipients_user_IDN*, a comma-separated list of each recipient user ID, where N is the sequence number in the list. These values are used in conjunction with `-rK` and `-rA` values. No default user ID.

-s2k-cipher-name — Specify the algorithm to use for passphrase-based encryption (PBE)

Format

-s2k-cipher-name *name*

Description

This option establishes the cipher algorithm to use for password-based encryption of the session key.

Configuration file option: S2K_CIPHER_NAME

Arguments

For *name*, the name of cipher to use.

The **-list-algo** command lists all the available algorithms. The default is TRIPLE_DES.

-s2k-count — Specify the count for passphrase-based encryption (PBE)

Format

-s2k-count *count*

Description

Allows user to specify the count value in an Iterated and Salted S2K password-based encryption process.

Configuration file option: "S2K_COUNT" on page 49.

Arguments

1024 - 65011712 inclusive.

The default is 65536.

-s2k-digest-name — Specify the digest algorithm for passphrase-based encryption (PBE)

Format

-s2k-digest-name *name*

Description

This option establishes the digest algorithm to use for password based encryption of the session key.

Configuration file option: S2K_DIGEST_NAME

Arguments

For *name*, the name of the digest to use. The **-list-algo** command lists all the available algorithms. The default is SHA_1.

-s2k-mode — Specify the mode for passphrase-based encryption (PBE)

Format

-s2k-mode *mode*

Description

This option establishes the password-based encryption (PBE) mode to use.

Configuration file option: S2K_MODE

Arguments

For *mode*, one of the following PBE modes:

0
Plain PBE (not recommended)

1
Salt- based PBE.

3
Salt- based PBE that is iterated.

The default is 3.

-s2k-passphrase — Specify the passphrase to use for passphrase-based encryption (PBE) and decryption

Format

```
-s2k-passphrase value
```

Description

This option establishes the passphrase to use for passphrase-based encryption (PBE) and decryption.

Configuration file option: S2K_PASSPHRASE

Arguments

For *value,* the passphrase value. No default. The system prompts the user.

-s2k-subkey-passphrase – Specify the passphrase to use for passphrase-based encryption (PBE) and decryption of a subkey

Format

-s2k-subkey-passphrase *value*

Description

This option establishes the passphrase to use for passphrase-based encryption (PBE) and decryption of a subkey.

Configuration file option: “S2K_SUBKEY_PASSPHRASE” on page 39.

Arguments

For *value,* specify a passphrase.

-signers-key-alias — Specify an alias for the system key

Format

```
-signers-key-alias alias
```

Description

This option specifies the alias of the system key of this OpenPGP system. The system key is the key used when signing data.

Configuration file option: SIGNERS_KEY_ALIAS

Arguments

For *alias,* the Java keystore alias name for the working key pair of this system. No default.

-signers-key-password — Specify a password for the system key

Format

-signers-key-password *password*

Description

This option specifies the password to the system key of this OpenPGP system. The system key is the key used when signing data.

Configuration file option: SIGNERS_KEY_PASSWORD

Arguments

For *password*, the key password. No default.

-sub-key-alias — Specify the alias for a new subkey during key generation

Format

-sub-key-alias *alias*

Description

This option specifies the alias to be used when generating a new subkey during key generation. If you do not specify this value, you are prompted for a value during key generation when an ElGamal subkey has been requested. When you are performing batch key generation with an ElGamal subkey, this option is required, and the request fails if you do not specify this option.

Configuration file option: SUB_KEY_ALIAS

Arguments

For *alias*, the alias to use for the ElGamal subkey to be generated. No default.

-system-CA-key-alias — Specify an alias for a new key pair certificate

Format

-system-CA-key-alias *alias*

Description

This option specifies the alias to use to sign a newly generated key pair’s certificate. If an alias is not specified, the certificate of the generated key pair certificate is self-signed.

Configuration file option: SYSTEM_CA_KEY_ALIAS

Arguments

For *alias*, the Java keystore alias name for the certificate authority (CA) of this system. No default

-system-CA-key-password — Specify a password for the certificate authority key

Format

-system-CA-key-password *password*

Description

This option specifies the password for the certificate authority key within the keystore.

Configuration file option: SYSTEM_CA_KEY_PASSWORD

Arguments

For *password*, the key password. The default is the keystore password.

-t — Treat input as text

Format

-t *charset_name*

Description

This option specifies a character set. Encryption Facility performs character conversions as follows:

- When producing an Encryption Facility message (commands **-e**, **-s**, and **-c**), Encryption Facility converts the data from the system's character set to this value. In addition to the character conversions, Encryption Facility converts end-of-line characters to carriage return and line feed.
- When processing an RFC 4880 message or an encryption facility message (commands **-d** and**-v**), Encryption Facility converts the data from this value to the system's character set. In addition to the character conversions, Encryption Facility converts end-of-line characters to line feed.
- When creating a detached signature (command **-b**), Encryption Facility converts the data from the local code page to UTF-8 and uses the UTF-8 characters to calculate or verify the detached signature. (Note that in this instance, the specified value is ignored as the local code page is assumed for the text.) In addition to the character conversions, Encryption Facility converts end-of-line characters to carriage return and line feed.
- When verifying with detached signatures (command **-v**), Encryption Facility converts the data from this character set to UTF-8 and uses the UTF-8 characters to calculate or verify the detached signature. In addition to the character conversions, Encryption Facility converts end-of-line characters to carriage return and line feed.

Configuration file option: LITERAL_TEXT_CHARSET

Text mode is only supported for plain text data either in:

- a file, which can include end-of-line characters such as newline or carriage return.
- a dataset or PDS(E) member, which has no non-text characters present anywhere within the data (including end-of-line characters) and where the end of each record is treated as if there were a virtual end-of-line marker present.

During text mode processing, any end-of-line characters (such as newline or carriage return) are removed. If the target format is a file, end-of-line characters are included in the decrypted data. If the target format is a dataset or PDS(E) member, end-of-line characters are not included in the decrypted data.

Any other non-printable characters will be converted based on the source and target code pages.

Text mode should not be used with structured data that contains any non-text characters except end-of-line characters as described above.

Arguments

For *charset_name*, the name of the character set to use for character conversion. If you specify the string _LOCAL, the command uses the default system code page. If not specified, the data is processed as binary. If specified without a value, UTF-8 is the default. The **-list-algo** command lists all the available character sets.

-trust-value — Specify a trust value

Format

-trust-value *integer*

Description

This option specifies an *integer* value from 0 to 255. You input this value when you import and generate OpenPGP certificates.

Configuration file option: TRUST_VALUE

Arguments

For *integer*, a value between 0 and 255. No default.

-trusted-comment — Specify a trust comment

Format

-trusted-comment *string*

Description

This option specifies a string added to OpenPGP certificates when they are imported or generated.

Configuration file option: TRUSTED_COMMENT

Arguments

For *string*, the text of the comment.

-use-embedded-file — Write data to a file specified in the data packet

Format

-use-embedded-file

Description

When unpacking a message, this option writes the data to the filename specified in the data packet. This option is mutually exclusive with the **-o** option used during decrypt and verify processing. Rewriting to the existing file must be confirmed.

Configuration file option: USE_EMBEDDED_FILENAME

Arguments

None.

-use-mdc — Specify the use of modification detection code

Format

-use-mdc

Description

This option specifies to use modification detection code.

Configuration file option: USE_MDC

Arguments

None.

-userID-comment — Specify a user ID comment for an OpenPGP certificate during key generation and key export

Format

-userID-comment *string*

Description

This option specifies a user ID comment for an OpenPGP certificate during key generation and key export. If you do not specify this option, you are prompted for a value during key generation, key export by alias, and key export by key ID.

Configuration file option: "USERID_COMMENT" on page 47

Arguments

For *string*, a comment to use with the user ID for the OpenPGP certificate. The comment section of an OpenPGP certificate user ID is optional. No default.

-userID-email — Specify a user ID email address for an OpenPGP certificate during key generation and key export.

Format

-userID-email *string*

Description

This option specifies a user ID email address for an OpenPGP certificate during key generation and key export. If you do not specify this option, you are prompted for a value during key generation, key export by alias, and key export by key ID.

Configuration file option: "USERID_EMAIL" on page 47

Arguments

For *string*, the user ID email address to use with the OpenPGP certificate. No default.

-userID-name — Specify a user ID for an OpenPGP certificate during key generation and key export

Format

-userID-name *name*

Description

This option specifies a user ID for an OpenPGP certificate during key generation and key export. If you do not specify this option, you are prompted for a value during key generation, key export by alias, and key export by key ID. When you are performing batch key generation or batch export, this option is required, and the request fails if you do not specify this option.

Configuration file option: "USERID_NAME" on page 48

Arguments

For *name*, the user ID name to use with the OpenPGP certificate. No default.

-x509-days-valid — Specify the number of days an X509 certificate is to be valid

Format

-x509-days-valid *days*

Description

This option specifies how many days a newly generated X.509 certificate should be valid when you use the generate command *-g*. If you do not specify this value, you are prompted for a value during key generation. When you are performing batch key generation this option is required, and the request fails if you do not specify the option.

Configuration file option: "X509_DAYS_VALID" on page 48

Arguments

For *days*, the number of days this X.509 certificate is to be valid. The range for *days* is a number from 1 to 9999, where 1 is the minimum number of days and 9999 is the maximum number of days allowed for an X.509 certificate to be valid. No default.

-yes — Specify yes to prompts

Format

-yes

Description

This option specifies an answer of **yes** to most interactive questions.

Configuration file option: ANSWER_YES

Arguments

None.

-z — Compress data

Format

-z *n*

Description

This option compresses the data according to the generated OpenPGP message level.

Configuration file option: COMPRESSION

Arguments

Compression level for *n*:

0
Do not use compression.

1
Use best speed for compression.

9
Use best compression. **Setting this value can result in a considerable impact to performance.**

-1
Use default compression.

Encryption Facility for OpenPGP Commands

Encryption Facility for OpenPGP commands are summarized alphabetically and include format, description of the command, and an explanation of any command arguments. For help with commands, use the `-h` command that can list command menu information in STDOUT.

-b — Sign the contents of an OpenPGP message and create an output file with signature

Format

-b *file*

Description

This command signs the contents of the OpenPGP message and creates one OpenPGP output file with an OpenPGP message that contains the signature. If commands to encrypt are also specified (`-e` or `-c`), this command is equivalent to specifying `-s`.

Arguments

For *file*, a valid file name for the data to be signed.

-c — Encrypt the contents of the OpenPGP message using PBE

Format

-c *file*

Description

This command encrypts the contents of the OpenPGP message using PBE.

Arguments

For *file*, a valid file name for the data to be encrypted.

If you are using only CPACF hardware cryptography with AES or TDES symmetric encryption and are not using any cryptographic coprocessor, you cannot encrypt session keys for public-key cryptography. Instead, use this command to encrypt session keys with PBE to protect OpenPGP messages

-compress — Compress data in OpenPGP message format

Format

-compress *file*

Description

This command compresses data in OpenPGP message format. This command does not sign nor encrypt the data after it is compressed. Use the Encryption Facility **-d** command to decompress the output from this command.

Use the **-z** command option with this command to specify a compression level. If a compression level is not specified, the compression level of 9 is used in order to perform the best compression.

Use the **-compress-name** command option with this command to specify a compression algorithm. If a compression algorithm is not specified, the ZIP compression algorithm is used.

Encryption Facility for OpenPGP uses the zEDC feature for compression if available and running with the required level of Java (IBM 31-bit SDK for z/OS, Java Technology Edition, Version 7 Release 1 or later).

zEDC requires a minimum input buffer size for compression and decompression. If the input data is smaller than the minimum threshold, the data is processed using traditional software-based compression and decompression.

For additional information about zEDC, see *z/OS MVS Programming: Callable Services for High-Level Languages*.

Arguments

For *file*, a valid file name for the data to be compressed.

-d — Decrypt or decompress an OpenPGP message

Format

-d *file* [*file* . . .]

Description

This command decrypts one or more encrypted OpenPGP messages.

Consider using the configuration file option USE_EMBEDDED_FILENAME or the **-use-embeeded-filename** command option, or the configuration file option CONFIDENTIAL or **-no-save** command option with this command. Otherwise, the specified file or data set specified on OUTPUT_FILE or on the **-o** command option is overwritten with the last file that you specify as input on the **-d** command.

Use this command to decompress an OpenPGP message containing only compressed data in the OpenPGP message format. The input OpenPGP message is not required to be encrypted in order for this command to perform decompression.

Encryption Facility for OpenPGP uses the zEDC feature for compression if available and running with the required level of Java (IBM 31-bit SDK for z/OS, Java Technology Edition, Version 7 Release 1 or later).

zEDC requires a minimum input buffer size for compression and decompression. If the input data is smaller than the minimum threshold, the data is processed using traditional software-based compression and decompression.

For additional information about zEDC, see *z/OS MVS Programming: Callable Services for High-Level Languages*.

Arguments

For *file*, one or more valid file names for data to be decrypted, decompressed, or both.

-e — Encrypt the contents of the OpenPGP message

Format

-e *file*

Description

This command allows you to use public key encryption with the specified recipients (see `-rP`, `-rK`, and `-rA`).

If you are using CPACF hardware cryptography with AES or TDES symmetric encryption and are not using any cryptographic coprocessors, you cannot use this command to encrypt session keys for public-key cryptography to protect OpenPGP messages. Instead, see the `-c` command.

Arguments

For *file*, a valid file name for the data to be encrypted.

-eA — Export an OpenPGP certificate by using an x.509 certificate alias from the OpenPGP keyring file

Format

-eA *alias* [*alias* . . .]

Description

This command allows you to interactively export an OpenPGP certificate. It sets fields that the OpenPGP certificate requires but that are not available in an x.509 certificate. You specify an alias for the x.509 certificate that Encryption Facility can use as a basis for the OpenPGP certificate that you want to export from the keystore.

Arguments

For *alias*, a set of Java keystore aliases of x.509 certificates that serve as the basis for the OpenPGP certificates to be exported.

-eK — Export an OpenPGP certificate by key ID from the OpenPGP keyring file

Format

-eK *key_ID* [*key_ID* . . .]

Description

This command exports OpenPGP certificates using key IDs.

Arguments

For *key_ID*, a set of key IDs of the certificate to be exported.

-eP — Export an OpenPGP certificate by user ID from the OpenPGP keyring file

Format

-eP *user_ID* [*user_ID* . . .]

Description

This command exports OpenPGP certificates by user IDs. User IDs match if the argument is a substring of the actual user ID.

Arguments

For *user_ID*, set of OpenPGP user IDs of the certificate to be exported.

-g — Generate a key pair as the system key for signatures

Format

-g

Description

This command interactively generates a key pair that serves as the system key for signatures. Invoke this command from the UNIX System Services environment.

This command updates the Java keystore and the OpenPGP keyring.

Arguments

None.

-h — Prints the Help menu to STDOUT

Format

-h

Description

This command prints the Help menu to STDOUT.

Arguments

None.

-i — Import an OpenPGP certificate into the OpenPGP keyring or keystore file

Format

-i *file* [*file* . . .]

Description

This command imports an OpenPGP certificate into the OpenPGP keyring or keystore file. Importing an OpenPGP certificate that contain secret keys or secret subkeys is written as an X.509 certificate and saved to the keystore file. The file can have more than one key in it.

Arguments

For *file*, one or more files as input.

-list-algo — Prints a list of algorithms to STDOUT

Format

-list-algo

Description

This command prints a list of all supported algorithms and character sets to STDOUT.

Arguments

None.

-pA — List information about public keys in the keyring file or as specified by alias

Format

-pA [*alias* . . .]

Description

This command lists information about all the public keys in the Java keystore or those public keys identified by the alias argument.

Arguments

Optional: For *alias*, alias of the certificate to display.

-pK — List information about the public keys in the keystore or those specified by key ID

Format

-pK [*key_ID* . . .]

Description

This command lists information about all the public keys in the keystore or keyring or those public keys identified by the key ID argument.

Arguments

Optional: For *key_ID*, key ID of the certificate to display.

-pP — List information about public keys in the keyring file or those specified by user ID

Format

-pP [*user_ID* . . .]

Description

This command lists information about all public keys in the keyring file or those public keys identified by the user ID argument.

User IDs are not case sensitive and match if the argument is a substring of the actual user ID.

Arguments

Optional: For *user_ID*, user ID of the OpenPGP certificate to display.

-prepare — Prepare the Java keystore to use existing keys in ICSF

Format

-prepare *key_label* [*key_label* . . .]

Description

This command prepares the Java keystore to use existing keys in ICSF.

Arguments

For *key_label*, the ICSF key labels of keys from the Java keystore.

-rebuild-key-index — Rebuild the indexes for the keyring file

Format

-rebuild-key-index

Description

This command rebuilds the indexes of the OpenPGP keyring file. Run this command if the indexes for the keyring have been compromised or lost.

Arguments

None.

-s — Sign the contents of an OpenPGP message using a key

Format

-s *file*

Description

This command signs the contents of the OpenPGP message using the key specified in the signers_KEY_ALIAS of the configuration file or **-signers-key-alias** command line option.

Arguments

For *file,* a valid file name for the data to be signed.

-v — Verify a signed OpenPGP message

Format

-v[*detached_signature*]*signed-file*

Description

This command verifies a signed OpenPGP message.

Arguments

For [*detached_signature*]*signed file,* both a detached signature AND a signed data file.

-version — Encryption Facility's current version and current APAR level

Format

-version

Description

Displays information about Encryption Facility's current version and current APAR level.

Arguments

None.

-xA — Delete key material associated with an alias

Format

-xA *alias* [*alias* . . .] *alias* [*alias* . . .]

Description

This command deletes key material from the Java keystore that is associated with one or more aliases. Aliases are used to access key material stored in the Java keystore.

Arguments

For *alias,* a set of aliases to be deleted.

-xK — Delete key material based on the key ID value

Format

-xK *key_ID* [*key_ID* . . .]

Description

This command deletes key material from the Java keystore and OpenPGP certificates in the OpenPGP keyring that have the specified key ID values. Key IDs are used to identify a specific key or key pair.

Arguments

For *key_ID*, a set of key IDs of certificates to be deleted.

-xP — Delete OpenPGP certificates associated with a user ID

Format

-xP *user_ID* [*user_ID* . . .]

Description

This command deletes OpenPGP certificates by user ID from the OpenPGP keyring. User IDs match if the argument is a substring of the actual user ID.

Arguments

For *user_ID*, a set of OpenPGP user IDs of the certificates to be deleted

Chapter 5. Encryption Facility for OpenPGP messages

This topic describes Encryption Facility for OpenPGP messages.

Encryption Facility for OpenPGP messages

Messages for Encryption Facility for OpenPGP:

CSD0000A **Confirm passphrase:**

Explanation

The passphrase needs to be confirmed.

System action

Waits for user to enter.

User response

Enter to confirm passphrase.

CSD0001A **Enter key password for *key_alias* .**

Explanation

The key password has not been specified in the configuration file or on the command line.

In the message text:

key_alias
Key alias.

System action

Waits for user to enter a password.

User response

Enter a key password.

CSD0002A **Enter keystore password for *keystore***

Explanation

The keystore password has not been specified in the configuration file or on the command line.

In the message text:

keystore
Name of the keystore.

System action

Waits for user to enter password.

User response

Enter key password.

CSD0003A **Please select what kind of key you want: (1) RSA (2) DSA (3) DSA and ElGamal**

Explanation

This message will appear during key generation when you attempt to generate a key in software.

System action

Waits for user to specify the type of key to generate.

User response

Specify a key type.

CSD0004A **Enter key size:**

Explanation

This message will appear during key generation. You must enter a valid key size. Default value is 1024.

System action

Waits for response.

User response

Specify the key size.

CSD0005I **Alias must be at least 1 character long.**

Explanation

You have not specified the required alias. An alias must be at least one character in length.

System action

Waits for response.

User response

Specify a valid alias.

CSD0006I **Passphrase not valid.**

Explanation

The passphrase is not valid.

System action

Waits for response.

User response

Enter a correct passphrase.

CSD0007I **The passphrase was retrieved from the system setting *value1*/*value2***

Explanation

In the message text:

value1
System setting value for the command

value2
System setting value for the configuration file

System action

Processing continues.

User response

None.

CSD0008I **Keystore already contains alias *value*. Do you want to continue? (yes/no)**

Explanation

When you attempted to generate a key using an alias, the alias already exists in your keystore.

In the message text:

value
Alias that exists in the keystore.

System action

Waits for response.

User response

Enter yes or no.

CSD0009A **Enter input file.**

Explanation

You have specified one of the commands that require an input file but have not specified the input file option.

System action

Waits for user input.

User response

Enter input file name.

CSD0010A **What is your first and last name?**

Explanation

This message will appear during key generation. This information is used to create the Distinguished Name (DN).

System action

Waits for response.

User response

Enter first and last name.

CSD0011A **What is the name of your organizational unit?**

Explanation

This message will appear during key generation. This information is used to create the Distinguished Name (DN).

System action

Waits for response.

User response

Enter your organizational unit.

CSD0012A **What is the name of your organization?**

Explanation

This message will appear during key generation. This information is used to create the Distinguished Name (DN).

System action

Waits for response.

User response

Enter your organization.

CSD0013A **What is the name of your city or locality?**

Explanation

This message will appear during key generation. This information is used to create the Distinguished Name (DN).

System action

Waits for response.

User response

Enter your city or locality.

CSD0014A **What is the name of your state or province?**

Explanation

This message will appear during key generation. This information is used to create the Distinguished Name (DN).

System action

Waits for response.

User response

Enter your state or province.

CSD0015A **What is the two-letter country code for this unit?**

Explanation

This message will appear during key generation. This information is used to create the Distinguished Name (DN).

System action

Waits for response.

User response

Enter a valid two-letter country code.

CSD0016A **Is *DN_info* correct? (yes/no)?**

Explanation

This message will appear during key generation and asks you to validate the information about the Distinguished Name (DN) value.

In the message text:

DN_info
Information about the Distinguished Name (DN)

System action

Waits for response.

User response

Enter yes or no.

CSD0017A **Confirm password:**

Explanation

This message will appear during key generation. After entering a key password, you are asked to confirm the password.

System action

Waits for response.

User response

Confirm the key password.

CSD0018A **Enter alias for *alias_name*.**

Explanation

This message will appear during key generation. The alias is used to identify the key in all future processing.

In the message text:

alias_name
Alias name

System action

Waits for response.

User response

Enter an alias to be used with the newly created key.

CSD0019A **For how many days should this OpenPGP certificate be valid (0 for always valid): [*number*]**

Explanation

This message will appear when importing an OpenPGP certificate. You can set OpenPGP certificates to always remain valid by entering a value of 0. The default is 0. Negative numbers are not allowed.

In the message text:

number
Number of days

System action

Waits for response.

User response

Enter a valid number_of_days.

CSD0020A **Real name.**

Explanation

This message will appear when importing an OpenPGP certificate. This information will be used to create the OpenPGP certificate user ID.

System action

Waits for response.

User response

Enter the real name.

CSD0021I **Name must be at least 5 characters long.**

Explanation

This message will appear when importing an OpenPGP certificate. The name must be at least 5 characters long.

System action

Processing continues.

User response

Enter a valid name.

CSD0022A **Email address**

Explanation

This message will appear when importing an OpenPGP certificate. This information will be used to create the OpenPGP certificate user ID.

System action

Waits for response.

User response

Enter email address.

CSD0023A **Passphrases do not match. Please try again.**

Explanation

You must enter a valid passphrase. Passphrases are case sensitive.

System action

Waits for response.

User response

Enter a valid passphrase.

CSD0024A **Comment:**

Explanation

This message will appear when importing an OpenPGP certificate. At least one user ID is required for an OpenPGP certificate. A user ID consists of three parts:

- Name
- Comment (optional)
- email address (optional)

System action

Waits for response.

User response

Enter a comment for this OpenPGP certificate.

CSD0025A **You specified user ID: "*value*". Change (N)ame, (C)omment, (E)mail, (X)Cancel or (O)kay to accept?**

Explanation

This message will appear when importing an OpenPGP certificate. The name must be at least 5 characters long.

In the message text:

value
user ID

System action

Waits for response.

User response

Accept, reject, or change the user ID.

CSD0026I **At least one user ID is required for an OpenPGP Certificate. A user ID consists of three parts: a name, a comment (optional), and an email address (optional).**

Explanation

This message will appear when importing an OpenPGP certificate. All OpenPGP certificates must have at least one user ID associated with them.

System action

Continuous processing.

User response

None.

CSD0027A **Add another user ID? (yes/no)**

Explanation

This message will appear when importing an OpenPGP certificate. You may specify multiple user ID's per OpenPGP certificate.

System action

Waits for response.

User response

Enter yes or no.

CSD0028I **Selection not valid: "*user_id*". Try again .**

Explanation

This message will appear when importing an OpenPGP certificate. You made an invalid selection when you confirmed the user ID from the prompt.

In the message text:

user_id
User ID

System action

Waits for response.

User response

Select a valid user ID.

CSD0029I **Exporting an OpenPGP certificate for *key_id*.**

Explanation

This message will appear when exporting an OpenPGP Certificate for informational purposes. The output file will contain an OpenPGP certificate for the ID specified.

In the message text:

key_id
Key ID, user ID, or alias

System action

Continues to export the certificate.

User response

None.

CSD0030A **ElGamal is an encrypt-only key. Encrypt-only keys must be exported as a subkey to a primary key that is capable of performing signatures. Specify the alias for the primary key.**

Explanation

This message will appear when you attempt to export an ElGamal key as the primary key. ElGamal keys may only be exported as subkeys.

System action

Processing continues.

User response

Specify another key at the next prompt.

CSD0031I **Alias *value* refers to an ElGamal key. ElGamal keys cannot be primary keys.**

Explanation

You have attempted to export an ElGamal key as the primary key. You can export ElGamal keys only as subkeys.

In the message text:

value
Key alias

System action

Processing continues.

User response

Enter a valid key.

CSD0032A **Do you want to add the exported OpenPGP Certificate to your OpenPGP key ring? (yes/no)**

Explanation

You have exported an OpenPGP Certificate. You have the option of adding a certificate to the keyring after exporting it to a file.

System action

Waits for response.

User response

Enter yes or no.

CSD0033I **This key will be signed with system CA alias: *certificate_authority_value*.**

Explanation

This message will appear during key generation if you have specified a certificate authority (CA). This key will be signed with the system CA alias.

In the message text:

certificate_authority_value
CA alias

System action

Signs key with the CA.

User response

None.

CSD0034A **Keystore already contains a system CA alias *certificate_authority_value*. Do you want to continue? (yes/no)**

Explanation

You have attempted to generate a key using an existing certificate authority (CA) alias.

In the message text:

certificate_authority_value
CA alias

System action

Waits for response.

User response

Enter yes or no.

CSD0035I **The specified value <*address*> is not a valid email address.**

Explanation

When importing an OpenPGP certificate, you have entered an invalid email address. Email addresses must be in the following format:

```
text@domainName
```

In the message text:

address
email address

System action

Processing continues.

User response

Enter a valid address.

CSD0036I **Password not valid.**

Explanation

You did not successfully enter and confirm a key password.

System action

Exits.

User response

Attempt to generate another key, enter, and confirm a key password.

CSD0037A **Passwords do not match. Please try again.**

Explanation

Confirmation of key password failed. You have 3 attempts before Encryption Facility exits with an error message.

System action

Waits for user to confirm key password.

User response

Confirm the key password.

CSD0038I **Importing OpenPGP certificate with primary *key_cipher_name* public key id: *alias_user_key*.**

Explanation

Message that displays that the key is being imported.

In the message text:

key_cipher_name
Key cipher name (either RSA or DSA)

alias_user_key
Alias, user ID, or key ID

System action

Imports key.

User response

None.

CSD0039I **Total number of commands processed successfully: *value***

Explanation

This message is displayed after a command has completed and indicates the number of commands that were successful after multiple attempts.

In the message text:

value
Number of successful operations

System action

Exits.

User response

None.

CSD0040A **Importing key into the OpenPGP key ring.**

Explanation

Informational message for importing a key into the keyring.

System action

Waits for response.

User response

None.

CSD0041A **Enter trust packet integer value (0 - 255).**

Explanation

This prompt allows you to specify a trust packet integer value.

System action

Waits for response.

User response

Enter integer for trust packet value.

CSD0042A **Enter trust packet comment:**

Explanation

This prompt allows you to specify a comment for the trust packet.

System action

Waits for response.

User response

Enter comment for the trust packet.

CSD0043I **Input not recognized.**

Explanation

Encryption Facility does not recognize the input.

System action

Prompts again for valid input.

User response

Enter valid input.

CSD0044A **For how many days should the X.509 certificate be valid (Maximum value 9999)?**

Explanation

This message prompts for the number_of_days to make valid a newly generated x.509 certificate.

System action

Waits for response.

User response

Enter valid number_of_days (1 - 9999).

CSD0045I **You requested to generate a key pair but did not specify a keystore type.**

Explanation

You must specify a keystore type for the generate command (-g).

System action

Exit.

User response

Run with the keystore type specified on the command line or in the configuration file.

CSD0046A **You specified a keystore type *type*. Generating RSA key. Do you want to continue? (yes/no).**

Explanation

When a hardware keystore type is specified, only RSA keys are allowed for generation.

In the message text:

type
Hardware keystore type

System action

Waits for response.

User response

Enter yes or no.

CSD0047I **The specified value *number* is not a valid number of days.**

Explanation

When prompted for the number of days for which a certificate key can be valid, you have entered a value that is not valid. X.509 can be valid for up to 9999 days. OpenPGP certificates can be valid indefinitely.

In the message text:

number
Number of days

System action

Prompts for a valid number of days

User response

Enter a valid number at the next prompt.

CSD0048A **Please select what type of hardware key you want: (1) PKDS (2) CLEAR**

Explanation

When you generate hardware RSA keys, only PKDS and Clear keys are valid. If you are using hardware cryptography and ICSF, understand the requirements. See "ICSF considerations" on page 15.

System action

Waits for response.

User response

Enter with type of key.

CSD0050I **Command processing ended abnormally:*text***

Explanation

Encryption Facility issues this message every time an invocation has not completed successfully. *text* records the message text.

In the message text:

text
Text of the message with details about the error. This message can display Encryption Facility messages as well as Java Software Developers Kit (SDK) failure messages. For more information about Java SDK messages, see IBM SDK, Java Technology Edition (www.ibm.com/docs/en/sdk-java-technology).

System action

Encryption Facility ends with a non-zero return code.

User response

Examine the message text, correct the error, and reinvoke the service.

CSD0051I **Command processing has completed successfully.**

Explanation

Encryption Facility issues this message every time an invocation has completed successfully.

System action

Encryption Facility ends with a zero return code.

User response

None.

CSD0052I **User *id* has ended the operation.**

Explanation

Encryption Facility issues this message when the userid under which the application is running ends the operation.

In the message text:

id
User ID under which the application is running

System action

Encryption Facility ends with a zero return code.

User response

None.

CSD0053I **At least one command flag or argument is required.**

Explanation

Encryption Facility requires at least one command option, or the command requires an argument.

System action

Ends processing.

User response

Provide a valid command.

CSD0054I **'Yes' was set to be the answer to any yes/no questions.**

Explanation

You indicated a command response of yes (ANSWER_YES in the configuration file) when you invoked Encryption Facility.

System action

Continues processing yes for answers.

User response

None.

CSD0055I **'No' was set to be the answer to any yes/no questions.**

Explanation

You indicated a command response of no (ANSWER_NO in the configuration file) when you invoked Encryption Facility.

System action

Continues processing no for answers.

User response

None.

CSD0056I **The combination of command arguments, option arguments, and/or configuration settings is not valid.**

Explanation

Encryption Facility requires a valid command option or argument, or a valid configuration option.

System action

Encryption Facility ends with a non-zero return code.

User response

Provide a valid command, and run the command again.

CSD0057I **You specified an unknown command flag or option: *command*.**

Explanation

The command or command option that you entered is not valid.

In the message text:

command
Command that is not valid

System action

Encryption Facility ends with a non-zero return code.

User response

Provide a valid command and run the command again.

CSD0058I **Ignoring arguments: *arguments*.**

Explanation

Encryption Facility ignores additional arguments that are specified on the command.

In the message text:

arguments
The arguments that are ignored

System action

Processing continues.

User response

Remove the additional argument from the command.

CSD0059I **One or more specified command flags require a final argument.**

Explanation

Some commands or command options require a final argument.

System action

Encryption Facility ends with a non-zero return code.

User response

Provide a valid final command argument and run the command again.

CSD0060I **The specified command flag *value* requires a final argument.**

Explanation

Some commands or command options require a final argument.

In the message text:

value
Command or command option

System action

Encryption Facility ends with a non-zero return code.

User response

Provide one or more valid final command arguments and run the command again.

CSD0061I **The program is ending because no command flags were specified.**

Explanation

Encryption Facility requires at least one command, command option, or file name.

System action

Encryption Facility ends with a non-zero return code.

User response

Provide a valid command, command option, or file name and run again.

CSD0062I **The specified option *option* requires a value.**

Explanation

The option requires at least one value.

In the message text:

option
Value for option

System action

Encryption Facility ends with a non-zero return code.

User response

Provide a valid value for the option and run again.

CSD0063I **The option *first_command_option* /*first_configuration_file_option* contradicts another specified option *second_command_option*/*second configuration file option*.**

Explanation

Some commands when specified together are mutually exclusive (for example, **-b** for detached signatures and **-s** for regular signatures).

In the message text:

first_command_option* /*first_configuration_file_option
First command or command option or configuration option

second_command_option* /*second_configuration_ file_option
Second command or command option or configuration option

System action

Encryption Facility ends with a non-zero return code.

User response

Be sure that the specified options are not mutually exclusive, and run the command again.

CSD0064I **The command option *first_option* contradicts another specified command option *second_option*.**

Explanation

Some options when specified together are mutually exclusive (for example, ANSWER_YES and ANSWER_NO in the configuration file).

In the message text:

first_option
First option

second_option
Second option

System action

Encryption Facility ends with a non-zero return code.

User response

Be sure that the specified commands are not mutually exclusive, and run the command again.

CSD0065I **No error message is available. Exception: *text***

Explanation

Encryption Facility issues this message text as part of message CSD0050E when it encounters an internal error and no error message is available.

In the message text:

text
Text of the exception message

System action

Encryption Facility ends with a non-zero return code.

User response

Review the exception text, activate the trace, and try again. Use trace records and the correct Encryption Facility invocation statements as well as the configuration file and invoke Encryption Facility again. If the problem persists, contact your IBM Service representative.

CSD0066I **One or more specified command flags require an output value on the command line or in the configuration file.**

Explanation

Some Encryption Facility commands require an option that specifies an output location (for example, -o command option and configuration file option OUTPUT_NAME). Encryption Facility issues this message when it encounters such commands without a specified location.

System action

Encryption Facility ends with a non-zero return code.

User response

Provide a location and run the command again.

CSD0067I **Alias *alias* does not point to a public/private key pair.**

Explanation

Encryption Facility issues this message text as part of message CSD0050E when it encounters an alias that does not refer to a public/private key pair.

In the message text:

alias
Alias

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the problem and run the command again.

CSD0068I **The value *value* was retrieved from the system setting *key*/*key_value*.**

Explanation

Trace record is written when key values are set with the system settings.

In the message text:

value
Value

key
Key name on the command

key_value
Key value setting in the configuration file

System action

yes or no.

User response

None.

CSD0069I **The password was retrieved from the system setting *key*/*key_value*.**

Explanation

Encryption Facility writes a trace record when the passwords are set with the system settings.

In the message text:

key
Key name on the command

key_value
Key value in the configuration file

System action

yes or no.

User response

None.

CSD0070A **The list of supported character sets is long. Do you wish to continue? (yes/no)**

Explanation

When running the `-list-algo` command, system returned output that is very long. You might not want to read all of the command output

System action

Waits for a response. If no, Encryption facility ends with a non-zero return code. If yes, processing continues.

User response

Enter yes or no.

CSD0071I **File *name* does not contain a valid OpenPGP certificate.**

Explanation

Input file did not contain a valid OpenPGP certificate.

In the message text:

name
Name of file

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0072I **Error encountered while attempting to import *name*. Error Message: *text*.**

Explanation

Import command failed; error message will display more information.

In the message text:

name
Name of file or data set that contains the OpenPGP certificate

text
Error message explanation

System action

Encryption Facility ends with a non-zero return code.

User response

Examine the message text, correct the error, and reinvoke the service.

CSD0073I **Total number of certificates imported successfully: *number*.**

Explanation

Message is displayed after all attempts to import and indicates the successful number of imports.

In the message text:

number
Number of certificates imported.

System action

yes or no.

User response

None.

CSD0074I **Total number of certificates not imported successfully: *number*.**

Explanation

Message is displayed after all attempts to import certificates and indicates the failed number of imports.

In the message text:

number
Number of certificates that failed to import

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the error, and reinvoke the service.

CSD0075A **Certificate <*certificate*> already exists in the key ring. Would you like to replace it? (yes/no)**

Explanation

User is attempting to import a certificate that already exists in the keyring.

In the message text:

certificate
Key ID

System action

Waits for response.

User response

Enter yes or no.

CSD0076I **Trust value <*value*> not valid. Values must be between 0 and 255.**

Explanation

Trust value must be between 0 and 255.

In the message text:

value
Trust value that is not valid

System action

Prompts again for valid input.

User response

Enter a valid value.

CSD0077A **Preparing RSA key for keystore type *type*. Do you want to continue? (yes/no)**

Explanation

Keys already generated using ICSF may be used with the application after they are successfully processed into a local keystore.

In the message text:

type
keystore type

System action

Waits for response.

User response

Enter yes or no.

CSD0078A **Deleting *identifier* from the keystore. Do you want to continue? (yes/no).**

Explanation

This message confirms that you really want this key deleted from the keystore.

In the message text:

identifier
String consisting of either the alias key ID or the user's alias key ID

System action

Waits for response.

User response

Enter yes or no.

CSD0079A **Deleting *identifier* from the OpenPGP key ring. Do you want to continue? (yes/no)**

Explanation

This message confirms that you want this key deleted from the OpenPGP keyring.

In the message text:

identifier
String consisting of either the alias key ID or the user's alias key ID

System action

Waits for response.

User response

Enter yes or no.

CSD0080I ***number_key* key(s) deleted successfully.**

Explanation

You can delete multiple keys at a time. This message informs you of how many keys were deleted successfully.

In the message text:

number_key
Number of keys deleted

System action

yes or no.

User response

None.

CSD0081I ***number_key*** **deletion(s) failed.**

Explanation

This message informs you how many keys failed deletion.

In the message text:

number_key
Number of keys that the system failed to delete

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the error, and reinvoke the service.

CSD0082I **Defaulting to *number* days valid.**

Explanation

When prompted for the number of valid days for a certificate, you entered without specifying a value, which indicates a default value of 90 days.

In the message text:

number
Default of 90 days

System action

Sets the certificate valid for 90 days.

User response

None.

CSD0083A **Input <*value*> not recognized. Please respond Yes or No.**

Explanation

You entered a value other than yes or no to a yes/no question.

In the message text:

value
Response text other than yes or no

System action

Waits for user response.

User response

Enter yes or no.

CSD0084I **Keystore type <*keystore type*> not valid.**

Explanation

You entered an invalid value for keystore type.

In the message text:

keystore type
keystore type that is not valid

System action

Issues an exception and exits.

User response

Specify a valid keystore type and rerun.

CSD0085I **OpenPGP Version 3 keys must be RSA.**

Explanation

For Version 3 OpenPGP certificates the keys must be type RSA.

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the error, and reinvoke the service.

CSD0086I **Unsupported algorithm specified <*algorithm*>.**

Explanation

You specified an unsupported algorithm.

In the message text:

algorithm
Unsupported algorithm

System action

Issues an exception and exits.

User response

Specify a supported algorithm and rerun.

CSD0087I **Modification detection code will be added when building the encrypted message.**

Explanation

Informational message indicating that modification detection code has been added.

System action

Processing continues.

User response

None.

CSD0088I **When building the encrypted message, modification detection code will not be added.**

Explanation

Informational message indicating that modification detection code has not been added.

System action

Processing continues.

User response

None.

CSD0089I **Data to package has format: *text*.**

Explanation

This is an informational message about the data package format.

In the message text:

text
Explanation of data package format

System action

Processing continues.

User response

None.

CSD0090I **No valid command flags were specified. Defaulting to decrypt.**

Explanation

When no valid commands or command options are specified, the default is to decrypt.

System action

Processing continues.

User response

None.

CSD0091I **Input has ASCII Armor.**

Explanation

This is an informational message indicating input has ASCII Armor.

System action

Processing continues.

User response

None.

CSD0092I **Message generated: *text*.**

Explanation

This is an informational message has been generated.

In the message text:

text
Text of the informational message

System action

Processing continues.

User response

None.

CSD0093I **Attempting to decrypt multiple inputs to the same output location.**

Explanation

You specified multiple inputs for decryption.

System action

Prompts for the user to continue processing.

User response

Consider using the `-use-embedded-file` option on the command or `USE_EMBEDDED_FILENAME` in the configuration file.

CSD0094I **Deleting certificate for: *alias_userid_key_id*.**

Explanation

This is an informational that indicates which key is being deleted.

In the message text:

alias_userid_key_id
Alias, user ID, or key ID

System action

Deletes key.

User response

None.

CSD0095I **Using ASCII armor.**

Explanation

Encryption Facility writes a trace record that indicates the use of ASCII Armor.

System action

Processing continues.

User response

None.

CSD0096I **Setting trust value *trust_value* and trust comment *trust_comment*.**

Explanation

Encryption Facility creates a log record when it sets the trust value and comment.

In the message text:

trust_value
Trust value

trust_comment
Trust comment

System action

Sets trust value and trust comment and creates a log record.

User response

None.

CSD0097I **Importing certificate: *certificate*.**

Explanation

Encryption Facility creates a trace record when importing a certificate.

In the message text:

certificate
Certificate to be imported

System action

Imports certificate.

User response

None.

CSD0098I **Total number of failed commands: *number*.**

Explanation

This message is displayed after a command has completed and shows how many commands failed after multiple attempts processed.

In the message text:

number
Number of failed commands

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0099I **Preparing ICSF hardware key label *label*.**

Explanation

An ICSF hardware key is being prepared for use with Encryption Facility .

In the message text:

label
Label of the ICSF key

System action

Prepares ICSF key for future use with Encryption Facility .

User response

None.

CSD0200I **IV size *iv* not valid for cipher algorithm:**

Explanation

In the message text:

iv
Size of the initialization vector (IV) for the cipher algorithm

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0400I **Message digests do not match.**

Explanation

The data in the message to be processed might have been modified.

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the error, and reinvoke the service.

CSD0401I **Could not find the key with key ID *key_id* to verify a signature.**

Explanation

The system is unable to find the key in the repository to verify the signature.

In the message text:

key_id
Key ID

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the error, and reinvoke the service.

CSD0500I **Insufficient bytes read for a partial data block.**

Explanation

While processing, Encryption Facility expected more input than that which was received.

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the error, and reinvoke the service.

CSD0501I **Insufficient bytes read. Expecting *value1* byte(s), read *value2* byte(s).**

Explanation

While processing, Encryption Facility expected more input than that which was received.

In the message text:

value1
The expected number of bytes to be processed

value2
The actual number of bytes received

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the error, and reinvoke the service.

CSD0600I **Could not find a valid session key to decrypt the message.**

Explanation

Encryption Facility issues this message as message text for CSD0050E. Encryption Facility supports multiple keys as well as passphrase-based encryption (PBE) to encrypt a message. If Encryption Facility is unable to find a valid key or passphrase for decryption, it issues this message.

System action

Encryption Facility exits with a non-zero return code.

User response

Correct the error and run again.

CSD0601I **Could not find any encrypted data in the encrypted message.**

Explanation

Encryption Facility issues this message as message text for CSD0050E when it tries to decrypt an OpenPGP message but does not encounter data.

System action

Encryption Facility exits with a non-zero return code.

User response

Correct the error and run again.

CSD0602I **Could not find a valid packet type in the input file.**

Explanation

Encryption Facility issues this message as message text for CSD0050E when it tries to decrypt an OpenPGP message, but does not encounter a valid packet type. It is possible that this message is not a valid OpenPGP message.

System action

Encryption Facility exits with a non-zero return code.

User response

Correct the error and run again. Ensure that the message is a valid OpenPGP message.

CSD0603I **Packet type *type* cannot be the first packet in a valid OpenPGP message.**

Explanation

Encryption Facility issues this message as message text for CSD0050E when it tries to decrypt an OpenPGP message but encounters an unexpected OpenPGP packet.

In the message text:

type
Packet type

System action

Encryption Facility exits with a non-zero return code.

User response

Correct the error and run again.

CSD0604I **Unsupported version number: *version***

Explanation

Unsupported version number:

In the message text:

version
Version that is not supported

System action

If the response is no, ends processing with a non-zero return code. If the response is yes, continues.

User response

Enter yes or no.

CSD0700A **File *file_name* already exists. Do you want to overwrite it?**

Explanation

Encryption Facility wants permission to over-write an existing file in the HFS/zFS.

In the message text:

file_name
Name of the file

System action

If the response is no, ends processing with a non-zero return code. If the response is yes, continues.

User response

Enter yes or no.

CSD0701A **Confidential processing was requested. Do you want to display data to STDOUT? (yes/no)**

Explanation

According to OpenPGP standards, OpenPGP messages cannot be unpackaged to persistent storage. Instead, you can print the contents to the STDOUT; Encryption Facility wants permission to send the output to the console.

System action

If the response is no, ends processing with a non-zero return code. If the response is yes, continues.

User response

Enter yes or no.

CSD0702A **Confidential processing was requested for a message labeled as binary. Do you want to display binary data to STDOUT? (yes/no)**

Explanation

According to OpenPGP standards, OpenPGP messages cannot be unpackaged to persistent storage. Instead, you can print the contents to the STDOUT; Encryption Facility wants permission to send the output to the console.

System action

If the response is no, ends processing with a non-zero return code. If the response is yes, continues.

User response

Enter yes or no.

CSD0704I **Hash algorithm *value* not valid or not supported.**

Explanation

Encryption Facility issues this message text as part of CSD0050E. The value of the hash algorithm is not supported or is not valid.

In the message text:

value
Hash value that is not valid

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the problem and run again.

CSD0705I **Cipher algorithm *value* not valid or not supported.**

Explanation

Encryption Facility issues this message text as part of CSD0050E. The value of the cipher algorithm is not supported or is not valid.

In the message text:

value
Cipher algorithm that is not valid

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the problem and run again.

CSD0706I **Compression algorithm *value* not valid or not supported.**

Explanation

Encryption Facility issues this message text as part of CSD0050E. The value of the compression algorithm is not supported or is not valid.

In the message text:

value
Compression algorithm that is not valid

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the problem and run again.

CSD0707I **Keys for OpenPGP Version 2 or 3 must be RSA; not for *algorithm_name***

Explanation

Encryption Facility issues this message text as part of CSD0050E. This version of the OpenPGP certificate only supports RSA keys.

In the message text:

algorithm_name
Algorithm name other than RSA

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the problem and run again.

CSD0708A **Could not find the key with key ID *key_id* to verify user ID "*user_id*". Do you want to continue? (yes/no)**

Explanation

OpenPGP Certificates bind user IDs to the keys that are contained in the certificate using signatures.

Other signatures can exist, where other parties have signed the user ID and primary key. Encryption Facility attempts to verify all signatures and issues this message when "no trust" has been established.

In the message text:

key_id
Key ID

user_id
User ID

System action

Encryption Facility waits for a response. You must respond yes or no. If the response is no, Encryption Facility terminates with a non-zero return code. Otherwise, it continues.

User response

Enter yes or no. If no, acquire a valid OpenPGP certificate and invoke Encryption Facility again.

CSD0709A **Could not find the key with key ID *key_id* to verify user ID attribute. Do you want to continue? (yes/no)**

Explanation

OpenPGP certificates uses signatures to bind user IDs to the keys that are contained in the certificate.

Other signatures can exist, where other parties have signed the user ID and primary key. Encryption Facility attempts to verify all signatures and issues this message when "no trust" has been established.

In the message text:

key_id
Key ID

System action

Encryption Facility waits for a response. You must respond yes or no. If the response is no, Encryption Facility terminates with a non-zero return code. Otherwise, it continues.

User response

Enter yes or no. If no, acquire a valid OpenPGP certificate and invoke Encryption Facility again.

CSD0710I **Verifying a self-signed certificate for user ID "*user_id*". Signing key ID: *key_id***

Explanation

OpenPGP Certificates bind user attributes to the keys encapsulated in the certificate using signatures.

A self-signed signature is one in which the primary key is used to produce a signature over the user ID and the primary key. Also, any number of other signatures can exist, where other parties have signed the user ID and primary key. Encryption Facility attempts to verify all signatures. It will issue this log message when it detects the self-signed signature.

In the message text:

user_id
User ID

key_id
Key ID

System action

Processing continues.

User response

None.

CSD0711I **Verifying a self-signed certificate for user attribute. Signing key ID: *key_id***

Explanation

OpenPGP Certificates bind user attributes to the keys encapsulated in the certificate using signatures.

A self-signed signature is one in which the primary key is used to produce a signature over the user attribute and the primary key. Also, any number of other signatures can exist, where other parties have signed the user attribute and primary key. Encryption Facility attempts to verify all signatures. It will issue this log message when it detects the self-signed signature.

In the message text:

key_id
Key ID

System action

Processing continues.

User response

None.

CSD07121I **Certificate with primary key ID *key_id* not valid. No self-signed user ID signature was found for user ID "*user_id*".**

Explanation

OpenPGP certificates use signatures to bind user IDs to the keys that are contained in the certificate. At least one self-signed signature must exist.

A self-signed signature is one in which the primary key is used to produce a signature over the user ID and the primary key. Also, any number of other signatures can exist, where other parties have signed the user ID and primary key. Encryption Facility attempts to verify all signatures. It will issue this error message when it does not encounter a self-signed signature.

In the message text:

key_id
Key ID

user_id
User ID

System action

Encryption Facility terminates with a non-zero return code.

User response

Acquire a valid OpenPGP certificate and invoke Encryption Facility again.

CSD0713A **Could not find the key with key ID *key_id* to verify a direct signature. Do you want to continue? (yes/no)**

Explanation

The system cannot find the key to verify a signature.

In the message text:

key_id
Key ID

System action

Waits for response.

User response

Enter yes or no.

CSD0714A **Could not find the key with key ID *key_id* to verify a revocation signature. Do you want to continue? (yes/no)**

Explanation

The system cannot find the key to verify a signature.

In the message text:

key_id
Key ID

System action

Waits for user response.

User response

Enter yes or no.

CSD0715I **Digital signature algorithm: *algorithm* not valid or not supported.**

Explanation

The system has found a combination of cipher and hash IDs that are not valid.

In the message text:

algorithm
Algorithm with the digital signature

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0716I **Public key encrypted session keys do not generate session keys when decrypting.**

Explanation

The system has found that you are attempting to decrypt with an encrypted session key.

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0717I **The checksum of the session key is not valid.**

Explanation

The system has found a session key checksum that is not valid.

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0718I **Input length <*value*> is not valid.**

Explanation

The system has found that the length of the input is not valid.

In the message text:

value
length value

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0719I **Input not valid.**

Explanation

The input is not valid.

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0720I **Unsupported asymmetric algorithm ID for Version 4 signatures: *algorithm*.**

Explanation

The system does not support the algorithm.

In the message text:

algorithm
Algorithm value

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0721I **Unsupported hash algorithm ID for Version 4 signatures: *algorithm*.**

Explanation

The system does not support the algorithm.

In the message text:

algorithm
Algorithm value

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0722I **Unsupported asymmetric algorithm ID for Version 3 signatures: *algorithm*.**

Explanation

The system does not support the algorithm.

In the message text:

algorithm
Algorithm value

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0723I **Unsupported hash algorithm ID for Version 3 signatures: *algorithm*.**

Explanation

The system does not support the algorithm.

algorithm
Algorithm value

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0724I **Revocation value: *value* not valid.**

Explanation

The revocation value for the certificate is not valid.

In the message text:

value
Value for revocation of certificate

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0725I **Boolean value *value* not valid.**

Explanation

A value that is not valid has been detected in the certificate.

In the message text:

value
Value of string

System action

Encryption Facility ends with a non-zero return code.

User response

Acquire a valid OpenPGP certificate and invoke Encryption Facility again.

CSD0726I **Unsupported cipher ID: *cipher_id*.**

Explanation

The system does not support the cipher ID.

In the message text:

cipher_id
Cipher ID

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0727I **Checksum failure. SecretKeyPacket data was manipulated.**

Explanation

The checksum has failed.

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0728I **OpenPGP Version 2 or 3 keys must be RSA keys.**

Explanation

The keys must be RSA key pairs.

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0729I **The keystore does not allow you to access private key data.**

Explanation

keystore does not allow private key data access.

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0730I **A modification detection code packet is required.**

Explanation

You must specify a modification detection code packet.

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0731I **Modification detection code verification failed. Packets might have been modified.**

Explanation

Verification of the modification detection code has failed. OpenPGP standards allow for a modification detection code to be appended to encrypted data. This code allows for the verification that the encrypted data has not been altered. The validity of the decryption results might be corrupted.

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0732I **Exportable value *value* not valid.**

Explanation

The system has found that a value on the export function is not valid.

In the message text:

value
  Value on the export

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0733I **Expecting *correct_value* octet(s) in input. Received input of length *incorrect_value*.**

Explanation

Expected length of the input value does not match what was specified.

In the message text:

correct_value
Expected value

incorrect_value
Received value

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0734I **Encountered unknown packet type.**

Explanation

The system does not recognize the packet.

Encryption Facility ends with a non-zero return code.

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0735I **Encountered unknown packet type. The content tag is: *content***

Explanation

The system does not recognize the packet.

In the message text:

content
Content tag of packet

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0736I **The leftmost bit of the content tag must be set.**

Explanation

The system does not recognize the packet.

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0737I **New format packet headers must have the second bit from the left set in the content tag.**

Explanation

The system does not recognize the packet.

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0738I **Hash value for hash algorithm not valid: *value*.**

Explanation

The system encountered a hash value that is not valid.

In the message text:

value
Hash value of the algorithm

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0739I **Key class for revocation subpacket not valid. The leftmost bit of the class must be set.**

Explanation

The system does not recognize the packet.

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0740I Packet header not valid. The leftmost bit is not set.

Explanation

The system does not recognize the packet.

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0741I New format packet headers must have the second bit from the left set.

Explanation

The system does not recognize the packet.

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0742I Old format packet headers must have the second bit from the left off.

Explanation

The system does not recognize the packet.

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0743I Size for the specified notation data flags not valid.

Explanation

The system does not recognize the packet.

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0744I **Image size is too large: *size*.**

Explanation

The system does not recognize the packet.

In the message text:

size
Size of the image

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0745I **Compression level not valid or not supported: *level*.**

Explanation

System indicates that an error has occurred about the specified level of compression.

In the message text:

level
Compression level that is not valid

System action

Returns a non-zero return code

User response

Select a supported level of compression and run the command again.

CSD0746I **Only one key server preference is supported. Any additional preferences are ignored.**

Explanation

Encryption Facility supports only one server preference and ignores any extra specifications.

System action

Processing continues.

User response

None.

CSD0747I **Only one key flag is supported. Any additional flags are ignored.**

Explanation

Encryption Facility supports only one key flag and ignores any extra specifications.

System action

Processing continues.

User response

None.

CSD0748I **Unsupported signature packet version: *number*.**

Explanation

Encryption Facility does not support the signature packet for this version.

In the message text:

number
Version number

System action

Encryption Facility ends with a non-zero return code.

User response

Acquire a valid OpenPGP certificate and invoke Encryption Facility again.

CSD0749I **Unsupported public key packet version: *number*.**

Explanation

Encryption Facility does not support the public key packet for this version.

In the message text:

number
Version number

System action

Encryption Facility ends with a non-zero return code.

User response

Acquire a valid OpenPGP certificate and invoke Encryption Facility again.

CSD0750I **Unsupported image encoding: *number*.**

Explanation

Encryption Facility does not support the image encoding for this version.

In the message text:

number
Version number

System action

Encryption Facility ends with a non-zero return code.

User response

Acquire a valid OpenPGP certificate and invoke Encryption Facility again.

CSD0751I **Private key for key id *key_id* was not found.**

Explanation

Encryption Facility cannot find the key ID.

In the message text:

key_id
Key ID

System action

Encryption Facility ends with a non-zero return code.

User response

Acquire a valid OpenPGP certificate and invoke Encryption Facility again.

CSD0752I **Encountered data format that is not valid or not supported.**

Explanation

Encryption Facility cannot find the key ID.

System action

Encryption Facility ends with a non-zero return code.

User response

Acquire a valid OpenPGP certificate and invoke Encryption Facility again.

CSD0753I **System CA keystore alias *value* was not found in keystore *keystore* .**

Explanation

Encryption Facility cannot find the certificate authority (CA) keystore alias in the specified keystore.

In the message text:

value
Certificate authority (CA) keystore alias

keystore
Name of the keystore

System action

Processing continues.

User response

Enter a valid CA key keystore alias on the command or in the configuration file.

CSD0754I **No private key found for signer's keystore alias *value*.**

Explanation

Encryption Facility cannot find the private key to sign the data in the signer's keystore alias.

In the message text:

value
Signer's keystore alias

System action

Encryption Facility ends with a non-zero return code.

User response

Acquire a valid OpenPGP certificate and invoke Encryption Facility again.

CSD0755I **ElGamal key generation can take several minutes.**

Explanation

This message is issued when ElGamal key generation occurs.

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD0756I **The specified key alias is ignored during ElGamal key generation: *alias*.**

Explanation

The key alias is not valid for the ElGamal key but processing continues.

In the message text:

alias
Key alias

System action

Processing continues.

User response

None.

CSD0757I **Encountered data format that is not valid or not supported in key ring: *keyring* .**

Explanation

Encryption Facility encountered a format error or does not support keyring data. Data is probably corrupted.

In the message text:

keyring
keyring name

System action

Encryption Facility ends with a non-zero return code.

User response

Try to rebuild the keystore indexes.

CSD0758I **Encountered data format that is not valid or not supported in the certificate.**

Explanation

Encryption Facility encountered a format error or does not support certificate data.

System action

Encryption Facility ends with a non-zero return code.

User response

Verify that the input location contains the correct data.

CSD0759I **Certificate not found.**

Explanation

One or more input values contain an OpenPGP certificated that is not valid.

System action

Encryption Facility ends with a non-zero return code.

User response

Verify that the input location contains the correct data.

CSD0760I **Could not find the key with key ID *key_id* to verify a direct signature.**

Explanation

Encryption Facility cannot find the key associated with the key ID to verify the signature.

In the message text:

key_id
Key ID

System action

Processing continues.

User response

Correct the error, and reinvoke the service.

CSD0761I **Could not find the key with key ID *key_id* to verify a revocation signature.**

Explanation

Encryption Facility cannot find the key associated with the key ID to verify a revocation signature.

In the message text:

key_id
Key ID

System action

Processing continues.

User response

Correct the error, and reinvoke the service.

CSD0762I **Could not find the key with key ID *key_id* to verify user ID "*user_id*".**

Explanation

Encryption Facility cannot find the key associated with the key ID to verify the user ID.

In the message text:

key_id
Key ID

user_id
User ID

System action

Processing continues.

User response

Correct the error, and reinvoke the service.

CSD0763I **Could not find the key with key ID *key_id* to verify user attribute.**

Explanation

Encryption Facility cannot find the key associated with the key ID to verify the user attribute.

In the message text:

key_id
Key ID

System action

Processing continues.

User response

Correct the error, and reinvoke the service.

CSD0764I **System CA *name* will be used to sign generated certificate.**

Explanation

Encryption Facility cannot find the key associated with the key ID to verify the user attribute.

In the message text:

name
Name of certificate authority (CA) system

System action

Processing continues.

User response

None.

CSD0765I **Enter system CA key password for *alias*.**

Explanation

Enter the key password associated with this system certificate authority (CA) alias.

In the message text:

alias
Alias

System action

Processing continues.

User response

Prompts again for valid input.

User response

Enter a correct alias.

CSD0766I **Creation time for primary key with key id *key_id* is greater than the current time. Do you want to continue? (yes/no).**

Explanation

The creation time for the primary key id specified exceeds the current time.

In the message text:

key_id
Primary key ID

System action

If yes, continues processing; if no, Encryption Facility ends with a non-zero return code.

User response

Enter yes or no.

CSD0767I **Creation time for subkey with key ID *key_id* is greater than the current time. Do you want to continue? (yes/no).**

Explanation

The creation time for the subkey ID specified exceeds the current time.

In the message text:

key_id
Subkey ID

System action

If yes, continues processing; if no, Encryption Facility ends with a non-zero return code.

User response

Enter yes or no.

CSD0768I **Output data can be exchanged with the owner of the key with key ID *key_id*.**

Explanation

This informational message identifies the trusted partner who can receive the data.

In the message text:

key_id
Key ID

System action

Processing continues.

User response

None.

CSD0769I **You cannot use alias *value*. Specify a new alias.**

Explanation

The specified alias cannot be reused.

In the message text:

value
Alias

System action

Prompts again for valid input.

User response

Enter a new alias.

CSD0770I **Key label *label* is not valid.**

Explanation

The length of the specified key label is not valid.

In the message text:

label
Key label

System action

Encryption Facility continues processing remaining labels, then ends with a non-zero return code.

User response

Correct the error, and reinvoke the service.

CSD0771I **Could not lock file *file*.**

Explanation

An I/O error occurred when Encryption Facility tried to write to an output file. Some other process already has the output file locked.

In the message text:

file
Name of the output file

System action

Encryption Facility ends with a non-zero return code.

User response

Ensure that the file is unlocked, and reinvoke the service.

CSD0772I **The keystore type *keystore_type* is not valid for ICSF key preparation.**

Explanation

The keystore type that is specified on the command or in the configuration file is not valid for ICSF.

In the message text:

keystore_type
Keystore type

System action

Encryption Facility ends with a non-zero return code.

User response

Select one of the following valid keystores for ICSF key preparation:

- JCECCAKS
- JCECCARACFKS
- JCERACFKS

CSD0773I **Character set name *name* not valid or not supported.**

Explanation

The character set name that is specified on the command or in the configuration file is not valid or supported.

In the message text:

name
Character set name

System action

Encryption Facility ends with a non-zero return code.

User response

Specify a valid character set and reinvoke the service.

CSD0774I **Keystore *keystore_name* with type *type* is read only. Deletions based on key ids will be attempted against the OpenPGP key ring only.**

Explanation

You specified keystore type JCECCARACFKS or JCERACFKS that are read only. Key material can not be deleted from these keystore types. Encryption Facility tries to find key information that matches the specified key ID in the keyring and deletes it from the ring.

In the message text:

keystore_name
Name of the keystore

type
Type of keystore.

System action

Processing continues.

User response

None.

CSD0775I **Product registration failed. See return code from IFAEDJReg: *return_code*.**

Explanation

Product registration has failed.

In the message text:

return_code
Return code from IFAEDJReg

System action

Encryption Facility ends with a non-zero return code.

User response

Examine the return code, correct the error, and reinvoke the service.

CSD0776I **Product deregistration failed. See return code from IFAEDJReg: *return_code*.**

Explanation

Product deregistration has failed

In the message text:

return_code
Return code from IFAEDJReg

System action

Encryption Facility ends with a non-zero return code.

User response

Examine the return code, correct the error, and reinvoke the service.

CSD0777I **User ID *id* will be used to load RACF keyring *keyring*.**

Explanation

You did not specify a RACF user ID, so the system ID will be used to load the RACF keyring.

In the message text:

id
User ID

keyring
Name of the keyring

System action

yes or no.

User response

None.

CSD0778I **Data set *name* has fixed record lengths. Fixed record lengths are not allowed for OpenPGP message output.**

Explanation

Encryption Facility cannot write OpenPGP messages to fixed block data sets.

In the message text:

name
Name of the output data set with fixed blocks

System action

Encryption Facility ends with a non-zero return code.

User response

Specify a variable block data set as the output for OpenPGP messages.

CSD0779I **Signature by key ID *key_id* is valid.**

Explanation

This message informs the user that digital signature verification occurred without any errors.

In the message text:

key_id
Key ID of signing key

System action

Processing continues.

User response

None.

CSD0780I **I/O exception encountered while saving updates to OpenPGP keyring: *keyring_name*.**

Explanation

An I/O error was encountered while attempting to save the OpenPGP keyring or any of its two index files.

In the message text:

keyring_name
Name of the keyring

System action

Encryption Facility ends with a non-zero return code.

User response

Correct any problems with the file system and try again.

CSD0781I **I/O exception encountered while saving updates to keystore: *keystore_name*.**

Explanation

An I/O error was encountered while attempting to save the keystore.

In the message text:

keystore_name
Name of the keystore

System action

Encryption Facility ends with a non-zero return code.

User response

Correct any problems with the file system and try again.

CSD0782I **ASCII Armor output requires a data set with a record length of at least 76 bytes. *data_set_name* has usable record length of *record_length* bytes.**

Explanation

ASCII Armor output consists of records of 76 bytes. The length of the specified data set is not valid.

In the message text:

data_set_name
Name of the data set

record_length
Length in bytes of the specified record length

System action

Encryption Facility ends with a non-zero return code.

User response

Specify a new output data set and try again.

CSD0783I **Flag *flag_value* is not a valid option or command.**

Explanation

The invocation for the option or command is not valid.

In the message text:

flag_value
Option or command value that is in error

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the command or option error.

CSD0784I **Flag *value* is a valid option or command but did not appear in the proper order on the command line.**

Explanation

The command invocation syntax is not correct.

In the message text:

value
Option or command value

System action

Encryption Facility ends with a non-zero return code.

User response

Use the following syntax to correct the command invocation:

```
[-homedir name] [options ....] [commands ....]
```

```
 [last args....]
```

where:

- *-homedir name* is the name of the home directory that contains the configuration file
- *options* is one or more valid configuration option
- *commands* is one or more valid command
- *last args* is one or more command argument

CSD0785I **Unsupported preferred symmetric algorithm ID *algorithm* found in certificate.**

Explanation

The system does not support the algorithm.

In the message text:

algorithm
Algorithm value

System action

Encryption Facility ends with a non-zero return code.

User response

Regenerate the OpenPGP certificate with the identified algorithm removed from the preference settings.

CSD0786I **Unsupported preferred hash algorithm ID *algorithm* found in certificate.**

Explanation

The system does not support the algorithm.

In the message text:

algorithm
Algorithm value

System action

Encryption Facility ends with a non-zero return code.

User response

Regenerate the OpenPGP certificate with the identified algorithm removed from the preference settings.

CSD0787I **Unsupported preferred compression algorithm ID *algorithm* found in certificate.**

Explanation

The system does not support the algorithm.

In the message text:

algorithm
Algorithm value

System action

Encryption Facility ends with a non-zero return code.

User response

Regenerate the OpenPGP certificate with the identified algorithm removed from the preference settings.

CSD0788A **Unsupported preferred hash algorithm *algorithm* found in certificate. Do you want to continue? (yes/no)**

Explanation

The system does not support the algorithm.

In the message text:

algorithm
Algorithm value

System action

Encryption Facility waits for a response.

User response

Enter yes to ignore the unsupported preference setting. If you enter no, you must regenerate the OpenPGP certificate with the identified algorithm removed from the preference settings.

CSD0800I **ASCII Armor header record not valid: *text*.**

Explanation

The ASCII Armor heading is not valid.

In the message text:

text
Explanation of the error

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the error.

CSD0801I **ASCII Armor format not valid. A record is too long. Length = *value*.**

Explanation

The ASCII Armor format is not valid.

In the message text:

value
Record length value that is incorrect

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the error.

CSD0802I **ASCII Armor header record in a multi-part header not valid.**

Explanation

The ASCII Armor heading is not valid.

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the error.

CSD0803I **The defined character encodings do not match across a multi-part ASCII Armor message.**

Explanation

The ASCII Armor message is not valid.

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the error.

CSD0804I **The CRC check for the ASCII Armor message failed.**

Explanation

The ASCII Armor message is not valid.

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the error.

CSD0805I **CRC line was not found in the ASCII Armor stream.**

Explanation

The ASCII Armor message is not valid.

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the error.

CSD0806I **End-of-input stream was reached before encountering an end ASCII Armor header record.**

Explanation

The ASCII Armor message is not valid.

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the error.

CSD0900I **Input is a negative number. Value is not valid.**

Explanation

Negative numbers are not valid input.

System action

Encryption Facility ends with a non-zero return code.

User response

Enter a valid number.

CSD0901I **Input is a too large to be a Scalar number. Value is not valid.**

Explanation

The data is not valid.

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the error, and reinvoke the service.

CSD0902I **The specified value is too large.**

Explanation

The data is not valid.

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the error, and reinvoke the service.

CSD0903I **Insufficient bytes were retrieved by the IOFacility.**

Explanation

The data is not valid.

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD1000I **String-to-key mode not valid: *value*.**

Explanation

Encryption Facility issues this message as part of the message text for CSD0050E. For passphrase-based encryption, you have specified a value for string-to-key mode that is not valid. For valid values, see "-s2k-mode — Specify the mode for passphrase-based encryption (PBE)" on page 68.

In the message text:

value
String-to-key mode that is not valid

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the error and run again.

CSD1001A **Enter passphrase for passphrase-based encryption:**

Explanation

If you do not specify the `-s2-passphrase` command line option or S2K_PASSPHRASE configuration option, Encryption Facility prompts you for a passphrase for passphrase-based encryption (PBE) of the data.

System action

Waits for response.

User response

Enter the passphrase.

CSD1002A **Enter passphrase for passphrase-based decryption:**

Explanation

If you do not specify the `-s2-passphrase` command line option or S2K_PASSPHRASE configuration option, Encryption Facility prompts you for a password for decrypting the data.

System action

Waits for response.

User response

Enter the passphrase.

CSD1003I **Specifier ID for passphrase-based decryption is not valid.**

Explanation

You specified a passphrase for decryption that is not valid.

System action

Processing continues.

User response

Enter a valid passphrase.

CSD1004I **Specifier ID not valid for passphrase-based encryption: *id*.**

Explanation

The data is not valid or is not supported.

In the message text:

id
Specifier ID that is not valid

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the error, and reinvoke the service.

CSD1005I **Could not peek first byte.**

Explanation

The data is not valid or is not supported.

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the error, and reinvoke the service.

CSD1007I **String-to-key count value must be must be between 1,024 and 65,011,712 inclusive. Setting value to 65,536.**

Explanation

String-to-key count value must be must be between 1,024 and 65,011,712 inclusive. Setting value to 65,536.

System action

Displays error, sets default value, and continues.

User response

None.

CSD1100I **Encountered unsupported type of certificate class from keystore: *name*.**

Explanation

In the message text:

name
Java class name of the certificate object

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD1101I **Attempting to update a READ only keystore (name: *name* type: *type*).**

Explanation

The system cannot generate a key to a read-only keystore like that for RACF.

In the message text:

name
Name

type
Type

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD1102I **Cannot self sign an ElGamal key pair.**

Explanation

ElGamal keys cannot be self signed.

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD1103I **Could not rename temporary file to key ring file *name*.**

Explanation

An error occurred while committing changes to OpenPGP keyring field,

In the message text:

name
File name

System action

Encryption Facility ends with a non-zero return code.

User response

Check the file system data and retry.

CSD1104I **Could not delete existing key ring file: *keyring*.**

Explanation

A problem occurred when saving updates to the key ring.

In the message text:

keyring
Filename of the keyring

System action

Displays error and exits

User response

None.

CSD1105I **Unable to add certificate with key ID *key_id* to key ring: *keyring*.**

Explanation

A problem occurred when saving updates to the keyring.

In the message text:

key_id
Key ID

keyring
Filename of the keyring

System action

Displays error and exits

User response

None.

CSD1106I **Data set *name* was not found or was unavailable for use.**

Explanation

Encryption Facility cannot find the z/OS-type data set, or the data set is not available to use.

In the message text:

name
Data set name

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the error and run again.

CSD1107I **I/O exception encountered while opening data set *name* .**

Explanation

Encryption Facility encountered an I/O error while trying to open the z/OS-type data set.

In the message text:

name
Data set name

System action

Encryption Facility ends with a non-zero return code.

User response

Ensure that the data set is valid.

CSD1200I **Log file records are not produced because an exception occurred during debug facility initialization. Exception message: *text* .**

Explanation

Encryption Facility cannot write log records. See the exception message.

In the message text:

text
Exception message that is issued during debug initialization

System action

Processing continues.

User response

Examine the message text, correct the error, and reinvoke the service.

CSD1201I **OpenPGP certificates require at least one user ID packet.**

Explanation

You must provide at least one user ID packet for the self-signed certificate.

System action

Encryption Facility ends with a non-zero return code.

User response

Acquire a valid OpenPGP certificate and invoke Encryption Facility again.

CSD1202I **Error while attempting to create new file *name* . The directory *directory_name* does not exist and cannot be created.**

Explanation

Directory cannot be created.

In the message text:

name
Name of the file

directory_name
Name of the file

System action

Encryption Facility ends with a non-zero return code.

User response

Check the file system or the configuration file and retry.

CSD1300I **Attempting to consume message in file *source_file* .**

Explanation

This message will appear in the log file and indicates that Encryption Facility is processing the data from the source file to the output file.

In the message text:

source_file
Name of source file

System action

Processing continues.

User response

None.

CSD1301I **Attempting to produce message for file *source_file* to file *output_file*.**

Explanation

This message will appear in the log file and indicates that Encryption Facility is processing the data from the source file to the output file.

source_file
Name of source file

output_file
Name of output file

System action

Processing continues.

User response

None.

CSD1302I **ASCII Armor is only supported when working with OpenPGP certificates.**

Explanation

Encryption Facility issues this message as part of the message text for CSD0050E. Encryption Facility supports ASCII Armor only for OpenPGP certificates and for files that are not z/OS-type data sets.

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the problem and run again.

CSD1303A **Enter a recipient's OpenPGP certificate user ID (Enter to end):**

Explanation

If you request public key encryption but do not specify a recipient on an option, you are prompted for a recipient’s user ID. User IDs are not case sensitive but must match the string or substring that is in the OpenPGP certificate.

System action

Waits for response.

User response

Enter one or more user IDs. After you specify all user IDs, enter without any text to notify Encryption Facility that all recipients have been specified.

CSD1304A **Enter a recipient's public key keystore alias. (Enter to end):**

Explanation

If you request public key encryption but do not specify a recipient on an option, you are prompted for a keystore alias to use for public encryption.

System action

Waits for response.

User response

Enter one or more aliases. After you specify all aliases, enter without any text to notify Encryption Facility that all recipients have been specified.

CSD1305A **Enter a recipient's hexadecimal key ID (Enter to end):**

Explanation

If you request public key encryption but do not specify a recipient on an option, you are prompted for a key ID to use for public encryption.

System action

Waits for response.

User response

Enter one or more hexadecimal 8-byte key IDs of the public key. After you specify all key IDs, enter without any text to notify Encryption Facility that all recipients have been specified.

CSD1306I **No recipients were specified, yet public encryption was requested. Defaults to passphrase-based encryption.**

Explanation

When you specify public key cryptography and do not specify recipients, Encryption Facility automatically reverts to passphrase-based encryption (PBE) and issues the message.

System action

Processing continues.

User response

None.

CSD1307I **No public key or certificate found for alias *value*.**

Explanation

The keystore did not return a result for the alias.

In the message text:

value
Alias name that is not in the keystore

System action

Continues processing additional recipients and passphrase-based encryption (PBE) if specified.

User response

Correct the problem and run again.

CSD1308I **No session keys were established. Defaults to passphrase-based encryption.**

Explanation

Public-based encryption has been specified, but Encryption Facility cannot locate valid public keys. Encryption Facility defaults to passphrase-based encryption (PBE).

System action

Continues processing using passphrase-based encryption (PBE).

User response

Correct the problem and run again.

CSD1309I **No public key or certificate found for OpenPGP certificate user ID "*user_id*".**

Explanation

Public-key encryption has been specified, but Encryption Facility cannot locate valid public keys for the specified user. Encryption Facility defaults to passphrase-based encryption (PBE).

In the message text:

user_id
User ID that is not in the OpenPGP keyring

System action

Continues processing additional recipients and passphrase-based encryption (PBE) if specified.

User response

Correct the problem and run again.

CSD1310I **No public key or certificate found for key ID *key_id*.**

Explanation

Public-key encryption has been specified, but Encryption Facility cannot locate valid public keys for the specified key ID. Encryption Facility defaults to passphrase-based encryption (PBE).

In the message text:

key_id
Key ID that is not in the OpenPGP keyring or keystore

System action

Continues processing additional recipients and passphrase-based encryption (PBE) if specified.

User response

Correct the problem and run again.

CSD1311I **An acceptable symmetric cipher algorithm name was not found. Using *default*.**

Explanation

Encryption Facility retrieves the algorithm name from the configuration file, command line options, or in the case of user IDs and key IDs, the preferences that are defined in a recipient’s OpenPGP certificate. If the supported symmetric cipher is not found, Encryption Facility defaults to the algorithm name *default*.

In the message text:

default
Default symmetric cipher algorithm name for encryption

System action

Continues using the default algorithm.

User response

Correct the problem and run again.

CSD1312I **An acceptable compression algorithm name was not found. Using *default*.**

Explanation

Encryption Facility retrieves the algorithm name from the configuration file, command line options, or in the case of user IDs and key IDs, the preferences that are defined in a recipient’s OpenPGP certificate. If the supported compression algorithm is not found, Encryption Facility defaults to the algorithm name *default*.

In the message text:

default
Default compression algorithm name for encryption

System action

Continues using the default algorithm.

User response

Correct the problem and run again.

CSD1331I **An acceptable hash algorithm name was not found. Using *default*.**

Explanation

Encryption Facility retrieves the algorithm name from the configuration file, command line options, or in the case of user IDs and key IDs, the preferences that are defined in a recipient’s OpenPGP certificate. If the supported hash algorithm is not found, Encryption Facility defaults to the algorithm name *default*.

In the message text:

default
Default hash algorithm name for encryption

System action

Continues using the default algorithm.

User response

Correct the problem and run again.

CSD1332I **Signatures were requested, but no signer's key_alias was specified.**

Explanation

Encryption Facility issues this message as part of the message text for CSD0050E. In order for Encryption Facility to sign data, it must find an alias to a public/private key pair in the keystore specified in the configuration file or as a command option. The recipient of the signed data must be a trusted partner, that is, a holder of a copy of the public key from the system public/private key.

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the problem and run again.

CSD1333A **Alias *value* refers to an X.509 certificate that is not valid. Do you want to continue? (yes/no)**

Explanation

The certificate referred to by the alias did not pass the validity check; the certificate might have expired or is being used before it is valid.

In the message text:

value
Alias

System action

Waits for a response. If no, ends with a non-zero return code. Otherwise, processing continues.

User response

Enter yes or no. If no, correct the problem and run again.

CSD1334A **OpenPGP Certificate for user ID "*user_id*" contains one or more revocation signatures. Do you want to continue? (yes/no)**

Explanation

The certificate referred to by the user ID did not pass the validity check; the certificate contains a revocation signature that is used to invalidate the certificate.

In the message text:

user_id
User ID

System action

Waits for a response. If no, ends with a non-zero return code. Otherwise, processing continues.

User response

Enter yes or no. If no, correct the problem and run again.

CSD1335A **OpenPGP certificate with the key that has key ID *key_id* contains one or more revocation signatures. Do you want to continue? (yes/no)**

Explanation

The certificate referred to by the key did not pass the validity check; the certificate contains a revocation signature that is used to invalidate the certificate.

In the message text:

key_id
Key ID

System action

Waits for a response. If no, ends with a non-zero return code. Otherwise, processing continues.

User response

Enter yes or no. If no, correct the problem and run again.

CSD1337I **Generate key pair was requested, but no keystore was specified.**

Explanation

This message will appear when you try to generate a key pair without specifying a keystore.

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD1338I ***number* errors were encountered while attempting to export OpenPGP certificates.**

Explanation

This message will appear when an export fails.

In the message text:

number
Number of the errors

System action

Displays the message and exits.

User response

None.

CSD1339A **OpenPGP Certificate for user ID "*user_id*" contains the following expired keys: key ID Expiration Date*text*Do you want to continue? (yes/no)**

Explanation

This message indicates that the system is processing an OpenPGP certificate with expired keys.

In the message text:

user_id
User ID

text
Key ID and expiration date

System action

Waits for a response.

User response

Enter yes or no.

CSD1340A **OpenPGP Certificate for user ID "*user_id*" has one or more user IDs that are not bound to the primary key. Do you want to continue? (yes/no)**

Explanation

This message indicates that the system is processing a certificate with user ID that are not bound to the primary key.

In the message text:

user_id
User ID

System action

Waits for a response.

User response

Enter yes or no.

CSD1341A **OpenPGP certificate for user ID "*user_id*" has one or more subkeys that are not bound to the primary key. Do you want to continue? (yes/no)**

Explanation

This message indicates that the system is processing a certificate with subkeys that are not bound to the primary key.

In the message text:

user_id
User ID

System action

Waits for a response.

User response

Enter yes or no.

CSD1342A **OpenPGP certificate with key ID *key_id* contains the following expired keys: key ID Expiration Date*text*Do you want to continue? (yes/no)**

Explanation

In the message text:

key_id
Key ID

text
Key ID expiration date

System action

Waits for a response.

User response

Enter yes or no.

CSD1343A **OpenPGP certificate containing key with key ID *key_id* has one or more user IDs that are not bound to the primary key. Do you want to continue? (yes/no)**

Explanation

This message indicates that the system is processing a certificate with user IDs that are not bound to the primary key.

In the message text:

key_id
Key ID

System action

Waits for a response.

User response

Enter yes or no.

CSD1344A **OpenPGP certificate containing key with key ID *key_id* has one or more subkeys that are not bound to the primary key. Do you want to continue? (yes/no)**

Explanation

This message indicates that the system is processing a certificate with subkeys that are not bound to the primary key.

In the message text:

key_id
Key ID

System action

Waits for a response.

User response

Enter yes or no.

CSD1345I **A signature could not be validated. Processing continues.**

Explanation

This message indicates that the system attempted to verify signatures but that the signing key is null.

System action

yes or no.

User response

None.

CSD1346I **The subkey with key ID *key_id* is not bound to the primary key (key ID *primary_key_id*) in the OpenPGP certificate.**

Explanation

This message indicates that the system is processing a certificate with subkeys that are not bound to the primary key.

In the message text:

key_id
Key ID

primary key_id
Primary key ID

System action

Waits for a response.

User response

Enter yes or no.

CSD1347I ***number*** **OpenPGP certificate(s) were exported successfully to *file*.**

Explanation

This message indicates how many OpenPGP certificates the system has processed successfully.

In the message text:

number
Number of certificates

file
Output file

System action

Processing continues.

User response

None.

CSD1348I **Error encountered while attempting to export *certificate*. Error message: *text*.**

Explanation

This message indicates that an error occurred when you tried to export a certificate.

In the message text:

certificate
certificate

text
Text of the error message

System action

Encryption Facility ends with a non-zero return code.

User response

Examine the message text, correct the error, and reinvoke the service.

CSD1349I **A certificate was found for user ID "*user_id*", but it did not contain a key capable of encryption.**

Explanation

This message indicates that the certificate is not capable of doing encryption.

In the message text:

user_id
user ID

System action

If there are no other valid recipient user IDs specified, Encryption Facility fails over to passphrase-based encryption (PBE) and prompts the user for a passphrase to generate an encryption key for the message. If there are other valid user IDs specified, the failing user ID is ignored (for example, the message will not be enciphered for that user ID) and processing will continue as normal for those other valid user IDs.

User response

Verify that the proper user ID is being specified. If it is, ensure that the key specified by the user ID was generated with the correct properties.

CSD1350I **The key with key ID *key_id* is not a key capable of encryption.**

Explanation

This message indicates that the key is not capable of performing encryption operations.

In the message text:

key_id
Key ID

System action

If there are no other valid recipient key IDs specified, Encryption Facility fails over to passphrase-based encryption (PBE) and prompts the user for a passphrase to generate an encryption key for the message. If there are other valid key IDs specified, the failing key ID is ignored (for example, the message will not be enciphered for that key ID) and processing will continue as normal for those other valid key IDs.

User response

Verify that the proper key ID is being specified. If it is, ensure that the key specified by the key ID was generated with the correct properties.

CSD1351I **Alias *alias* refers to a key that is not capable of encryption.**

Explanation

This message indicates that the certificate is not capable of doing encryption.

In the message text:

alias
Alias ID

System action

If there are no other valid recipient aliases specified, Encryption Facility fails over to passphrase-based encryption (PBE) and prompts the user for a passphrase to generate an encryption key for the message. If there are other valid aliases specified, the failing alias is ignored (for example, the message will not be enciphered for that alias) and processing will continue as normal for those other valid aliases.

User response

Verify that the proper alias is being specified. If it is, ensure that the key specified by the alias was generated with the correct properties.

CSD1352I **Displaying certificate with alias: *value* dn: *certificate*[expired: *date*]**

Explanation

This message displays the certificate value for the specified alias.

In the message text:

value
Alias ID

certificate
Certificate

date
Expiration date

System action

Processing continues.

User response

None.

CSD1353I **Displaying OpenPGP certificate whose user IDs match: "*user_id*"*certificate***

Explanation

This message displays the certificate value for the matching IDs.

In the message text:

user_id
User ID

certificate
Certificate

System action

Processing continues.

User response

None.

CSD1354I **Displaying certificate for key ID: *key_id* dn: *certificate*[expired:*date*]**

Explanation

This message displays the certificate value for the key IDs.

In the message text:

key_id
Key ID

certificate
Certificate

date
Expiration date

System action

Processing continues.

User response

None.

CSD1355I **Displaying OpenPGP certificate for key ID: *key_idcertificate***

Explanation

This message displays the certificate value for the OpenPGP certificate with the specified key ID.

In the message text:

key_id
Key ID

certificate
Certificate

System action

Processing continues.

User response

None.

CSD1356I **No signature packet found in file *file*.**

Explanation

This message indicates that the signature packet has not been found.

In the message text:

file
Name of the file

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD1357I **Signature verification failed.**

Explanation

This message indicates that the signature verification has failed; however, Encryption Facility might have updated the output.

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD1358I **Error encountered while attempting to delete *certificate*. Error Message: *text***

Explanation

This message indicates an error occurred when trying to delete the certificate.

In the message text:

certificate
Certificate

text
Error message explanation

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD1359I **The specified provider *provider* is not valid.**

Explanation

This message will appear if the JCE cryptographic provider is not valid.

In the message text:

provider
Name of the JCE cryptographic provider

System action

Encryption Facility ends with a non-zero return code.

User response

None.

CSD1360I **Provider: *provider* inserted in the provider list: position: *position*.**

Explanation

This message indicates that the trace record contains the name of the JCE cryptographic provider and its position in the provider list.

In the message text:

provider
Name of the JCE cryptographic provider

position
Position of the provider name in the list

System action

Processing continues.

User response

None.

CSD1362I **User *id* sending message to: *output_file***

Explanation

This message indicates that the information about the system ID is written to the log file.

In the message text:

id
System ID under which the application is running

output_file
Name of the output file

System action

Processing continues.

User response

None.

CSD1363I **Generating *key_type* key.**

Explanation

This message indicates that the specified key type has been generated.

In the message text:

key_type
Type of key

System action

Processing continues.

User response

None.

CSD1364I **This key will be self-signed with alias: *value***

Explanation

This message indicates that the key will be self-signed with the specified alias.

In the message text:

value
Alias of signing key

System action

Processing continues.

User response

None.

CSD1365I **This key will be signed with alias: *value***

Explanation

This message indicates that the key will be signed with the specified alias.

In the message text:

value
Alias of signing key

System action

Processing continues.

User response

None.

CSD1366I **User: *user_id* has generated key: *key_id* in Java keystore: *Java_store*.**

Explanation

This message indicates that the specified user ID has generated the key ID in the named Java store.

In the message text:

user_id
User ID

key_id
Key ID

Java_store
Name of the Java keystore

System action

Processing continues.

User response

None.

CSD1367I **Hardware key type <*type*> not valid.**

Explanation

You specified an incorrect key type for hardware generation.

System action

Prompts again for valid input.

In the message text:

type
Hardware type that is not valid

User response

Enter a valid key type on the command.

CSD1368I **Keystore type not specified.**

Explanation

You must specify a keystore type for key generation.

System action

Encryption Facility issues an exception message and ends with a non-zero return code.

User response

Run again with the keystore type specified

CSD1369I **A problem was encountered loading the configuration file. Exception message: *text***

Explanation

The configuration file is not valid.

In the message text:

text
Text of error message

System action

Encryption Facility ends with a non-zero return code.

User response

Check the file system or the configuration file.

CSD1370I **The value <*value*> is not valid for the option: *option1* / *option2***

Explanation

The value on the command or in the configuration file is not valid.

In the message text:

value
Incorrect value

option1
Configuration file option

option2
Command line option

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the error, and reinvoke the service.

CSD1371I **Keystore name not specified.**

Explanation

You did not specify a keystore name.

System action

Exit with a non zero return code.

User response

Specify a valid keystore name and reinvoke the service.

CSD1372I **X.509 certificate summary for key_alias *value* Key ID: *key_id* Key Type: *key_type* Key Size: *key_size* Keyring User ID(s): *keyring_value*.**

Explanation

This message issues a summary for the X.509 certificate that is generated for the keyring.

In the message text:

value
Alias

key_id
Key ID

key_type
Key type

key_size
Size of the key

keyring_value
Keyring user ID

System action

Processing continues.

User response

None.

CSD1373I **Only 1024 bit ElGamal keys are supported. Setting key size to 1024.**

Explanation

This informational message indicates the ElGamal keys and key size.

System action

Processing continues.

User response

None.

CSD1374I **Only 1024 bit DSA keys are supported. Setting key size to 1024.**

Explanation

This informational message indicates the DSA keys and key size.

System action

Processing continues.

User response

None.

CSD1375I **The specified value <*value*> is not a valid key size. RSA key size must be between 1024 and *RSA_value* and also divisible by 8.**

Explanation

The RSA key values are not valid.

In the message text:

value
Alias

RSA value
RSA key size value

System action

Prompts again for valid input.

User response

Enter a valid value.

CSD1376I **Option <*value*> specified in configuration file not valid.**

Explanation

An option in the configuration file is not valid.

In the message text:

value
Option that is not valid

System action

Encryption Facility ends with a non-zero return code.

User response

Correct the error, and reinvoke the service.

CSD1378I **Data set *data_set_name* has fixed record lengths. Fixed record lengths are not allowed for OpenPGP message output.**

Explanation

Fixed-record-length data sets cannot serve as the output data set for Encryption Facility encryption, signature command processing, or both.

In the message text:

data_set_name
Name of the data set

System action

Encryption Facility ends with a non-zero return code.

User response

Specify new output destination.

CSD1400I **Batch processing requires option *option command_option/ configuration_file_option* to be specified.**

Explanation

Either batch key generation was specified with the `-batch-generate`/BATCH_GENERATE command option or batch public key export was specified with the `-batch-export`/BATCH_EXPORT command option, but all required command options have not been specified in order to successfully complete batch processing.

See "-batch-generate — Specify batch key generation" on page 54 or "-batch-export — Specify batch public key export" on page 54 for a list of command options that are required for these batch processing functions.

In the message text:

command_option /configuration_file_option
The command option or configuration file option that is required for batch processing but was not specified

System action

Encryption Facility ends with a non-zero return code.

User response

Specify the *command_option /configuration_file_*option displayed in the message and run the command again.

CSD1401I **Batch processing requires either the *first_command_option/ first_configuration_file_option* option or the *second_command_option/ first_configuration_file_option* option to be specified.**

Explanation

Either batch key generation was specified with the `-batch-generate`/BATCH_GENERATE command option or batch public key export was specified with the `--batch-export`/BATCH_EXPORT command option, but all required command options have not been specified in order to successfully complete batch processing.

See "-batch-generate — Specify batch key generation" on page 54 or "-batch-export — Specify batch public key export" on page 54 for a list of command options that are required for these batch processing functions.

In the message text:

first_command_option /first_configuration_file_option
First command option or configuration file option

second_command_option /first_configuration_file_option
Second command option or configuration file option

One of the two command options or configuration file options is required for batch processing but neither was specified.

System action

Encryption Facility ends with a non-zero return code.

User response

Specify one of the two command options or configuration file options displayed in the message and run the command again.

CSD1402I **Batch option *command_option/configuration_file_option* requires at least one argument.**

Explanation

The `-batch-generate`/BATCH_GENERATE command option was specified without an argument.

In the message text:

command_option /configuration_file_option
The command option or configuration file option argument that is required for batch processing but was not specified

System action

Encryption Facility ends with a non-zero return code.

User response

A `-batch-generate`/BATCH_GENERATE command option requires at least one argument. See "-batch-generate — Specify batch key generation" on page 54 for a list of arguments that may be used with this command option and run the command again.

CSD1403A **What is your first and last name? [*response*]**

Explanation

This message will appear during key generation. This information is used to create the Distinguished Name (DN).

In the message text:

response
Valid online response

One of the two command options or configuration file options is required for batch processing but neither was specified.

System action

System waits for a response.

User response

Enter first and last name for *response*.

CSD1404A **What is the name of your organizational unit? [*response*]**

Explanation

This message will appear during key generation and is used to create the Distinguished Name (DN).

In the message text:

response
Valid online response

One of the two command options or configuration file options is required for batch processing but neither was specified.

System action

System waits for a response.

User response

Enter the name of the organizational unit for *response*.

CSD1405A **What is the name of your organization? [*response*]**

Explanation

This message will appear during key generation and is used to create the Distinguished Name (DN).

In the message text:

response
Valid online response

One of the two command options or configuration file options is required for batch processing but neither was specified.

System action

System waits for a response.

User response

Enter the name of the organization for *response*.

CSD1406A **What is the name of your city or locality? [*response*]**

Explanation

This message will appear during key generation and is used to create the Distinguished Name (DN).

In the message text:

response
Valid online response

One of the two command options or configuration file options is required for batch processing but neither was specified.

System action

System waits for a response.

User response

Enter the name of your city or locality for *response*.

CSD1407A **What is the name of your state or province? [*response*]**

Explanation

This message will appear during key generation and is used to create the Distinguished Name (DN).

In the message text:

response
Valid online response

One of the two command options or configuration file options is required for batch processing but neither was specified.

System action

System waits for a response.

User response

Enter the name of your state or province for *response*.

CSD1408A **What is the two-letter country code for this unit? [*response*]**

Explanation

This message will appear during key generation and is used to create the Distinguished Name (DN).

In the message text:

response
Valid online response

One of the two command options or configuration file options is required for batch processing but neither was specified.

System action

System waits for a response.

User response

Enter a valid two-character country code for *response*.

CSD1409I **Referencing a PDS or PDSE data set without a member name is not supported.**

Explanation

You cannot specify a PDS or PDSE data set without a member name.

System action

Encryption Facility ends with a non-zero return code.

User response

Specify a member name within the PDS or PDSE data set.

CSD1410I **The supplied input and output sources must be different.**

Explanation

You cannot specify the same source for input and output.

System action

Encryption Facility ends with a non-zero return code.

User response

Ensure that the command input and output specified by either the -o command option or the OUTPUT_FILE configuration option are different.

CSD1411I **The supplied input sources must be different.**

Explanation

You cannot specify the same source of input for a command that takes multiple arguments.

System action

Encryption Facility ends with a non-zero return code.

User response

Ensure that the source for each input that is supplied to the command are different.

CSD1412I **The distinguished name information supplied for the certificate was not valid. You must supply at least one field.**

Explanation

During certificate generation, you are required to supply at least one valid field for the distinguished name (DN).

System action

Encryption Facility waits for you to enter a valid DN. If batch processing is enabled, Encryption Facility ends with a non-zero return code.

User response

Ensure that you supply at least one valid field for the DN through either the command prompt or the -dn-* command options or DN_* configuration options.

CSD1413I **A certificate with the following alias *alias_name* has an empty subject distinguished name. Replace the certificate with one containing a valid subject distinguished name.**

Explanation

A certificate with an empty subject distinguished name was encountered during command processing.

In the message text:

alias_name
Alias name.

System action

Processing continues.

User response

Remove the certificate and replace the certificate with one containing a valid subject distinguished name. Any certificate with an empty subject distinguished name will not be usable with Java Technology Edition, Version 8 or later, and will prevent the entire keystore from being loaded.

CSD1414I **A certificate with the following alias *alias_name* has an empty issuer distinguished name. Replace the certificate with one containing a valid issuer distinguished name.**

Explanation

A certificate with an empty issuer distinguished name was encountered during command processing.

In the message text:

alias_name
Alias name.

System action

Processing continues.

User response

Remove the certificate and replace the certificate with one containing a valid issuer distinguished name. Any certificate with an empty issuer distinguished name will not be usable with Java Technology Edition, Version 8 or later, and will prevent the entire keystore from being loaded.

CSD1415I **A certificate within the keystore has a non-valid or missing subject or issuer distinguished name and was unable to load. Replace the certificate with one containing a valid subject or issuer distinguished name.**

Explanation

A certificate with an empty or non-valid distinguished name was encountered during command processing which prevented the entire keystore from being loaded when running with Java Technology Edition, Version 8 or later.

System action

Encryption Facility ends with a non-zero return code.

User response

Remove the certificates and replace the certificates with ones containing a valid distinguished name. Encryption Facility will only be able to process a keystore containing certificates with valid distinguished names while running Java Technology Edition, Version 8 or later.

CSD1416I **No embedded filename found in message.**

Explanation

The USE_EMBEDDED_FILENAME or -use-embedded-file option were specified, but no embedded filename was found in the OpenPGP message.

System action

Encryption Facility ends with a non-zero return code.

User response

Attempt to verify that the OpenPGP message contains a valid filename string. Alternatively, disable the -use-embedded-filename/USE_EMBEDDED_FILENAME options if they are not required.

CSD1417I **Importing OpenPGP certificate with primary *key_cipher_name* secret key ID: *Key_ID*...**

Explanation

User is attempting to import a Secret Key certificate.

In the message text:

key_cipher_name
Cipher ID associated with the OpenPGP certificate

Key_ID
Key ID associated with the OpenPGP certificate.

System action

Imports key.

CSD1418I **Creating X.509 certificate for secret key ID: *Key_ID*...**

Explanation

An X.509 certificate is being created for secret keys or secret subkeys during an import operation.

In the message text:

Key_ID
Key ID associated with the secret key or secret subkey.

System action

Creates X.509 certificate for specified key.

CSD1419A **Enter key password for encrypted *Cipher_ID Packet_Type*:**

Explanation

The key password for the encrypted key being imported.

In the message text:

Cipher_ID
Cipher ID of the key being decrypted.

Packet_Type
Packet type for packet being decrypted.

System action

Waits for user to enter a password.

User response

Enter a key password.

CSD1420I **Version 3 secret key packets unsupported for import.**

Explanation

Certificates containing V3 secret key packets are not supported for import in Encryption Facility.

System action

Informs user of error and exits.

User response

Provide a certificate with V4 secret key packets.

CSD1421A **Enter encrypting key password:**

Explanation

Enter the password that will be used for password-based encryption of the secret key material to be exported.

System action

Waits for user to enter a password.

User response

Enter the encrypting key password.

CSD1422A **Enter alias for key:**

Explanation

Enter the alias of the key that is to be exported. The alias is used to retrieve the secret key material from the keystore when generating the secret key packet to be exported.

System action

Waits for user to enter an alias.

User response

Enter an alias.

CSD1423I **Error encountered attempting to decrypt key material. Check to ensure correct password was supplied.**

Explanation

An error was encountered attempting to decrypt the encrypted secret key material. This could be caused by a malformed packet or supplying an incorrect decrypting password.

System action

Informs user of error and exits.

CSD1424A **Reuse X509 certificate info: (<*distinguished_name*> and validity: *days_valid*) for the next subkey? (yes/no)**

Explanation

If you are importing a secret key with one or more subkeys, you are asked if you would like to reuse the distinguished name (DN) and validity period for the next subkey to be imported. The DN and validity period are from the previously imported secret key or secret subkey from the same session.

In the message text:

distinguished_name
The DN set from the previously imported secret key.

days_valid
The validity period set from the previously imported secret key.

System action

Waits for user to enter yes or no.

User response

Enter yes or no.

CSD1425I **WARNING! This certificate already exists in the key ring. Overwriting it may result in a loss of data.**

Explanation

When you are processing an export by alias or keyID, you might be prompted to overwrite an OpenPGP certificate in the keyring with the OpenPGP certificate that is created by the processor for exporting. Because newly generated certificates might not contain all of the data that is found in the original certificate, overwriting the original certificate might result in the loss of data.

System action

Informs user of potential concerns with warning message and continues processing.

User response

None.

CSD1426I **No X.509 certificate corresponding to keyID {0} was found in the keystore.**

Explanation

An error was encountered attempting to fetch an X.509 Certificate from the keystore by key ID.

System action

Informs user of error and exits.

User response

Check that both the keystore and X.509 certificate exist and are not removed or changed during the execution of the Encryption Facility command.

CSD1427I **Expected exception caught: *ExceptionType*. Processing will continue.**

Explanation

The listed exception that was caught during processing was expected and processing does not need to be interrupted. This message only appears in trace files when debug options are specified in the command.

In the message text:

ExceptionType
The type of exception caught.

System action

Logs expected event in trace file and continues processing.

User response

None.

CSD1428I **Unrecognized signature subpacket detected: *UnsupportedSubpacket*. Accept Unrecognized Subpackets command specified. Subpacket will be ignored.**

Explanation

An unsupported signature subpacket was detected during the import of an OpenPGP certificate if the user specified the ACCEPT_UNRECOGNIZED_SUBPACKETS option.

In the message text:

UnsupportedSubpacket
The unsupported signature subpacket.

System action

Displays message and continues processing.

User response

None.

CSD1429I **Unrecognized signature subpacket detected: *UnsupportedSubpacket*. Accept Unrecognized Subpackets command not specified. Certificate will be rejected.**

Explanation

An unsupported signature subpacket was detected during the import of an OpenPGP certificate if the user did not specify the ACCEPT_UNRECOGNIZED_SUBPACKETS option.

In the message text:

UnsupportedSubpacket
The unsupported signature subpacket.

System action

Displays message and exits.

User response

User must specify the -accept-unrecognized command option or the ACCEPT_UNRECOGNIZED_SUBPACKETS ibmef.config option.

CSD1430I **IBM Encryption Facility for z/OS V1.2 - APAR: <*current-installed-APAR-number*>**

Explanation

Displays current APAR and Encryption Facility version.

In the message text:

current-installed-APAR-number
The currently installed APAR number.

System action

Displays information and exits.

User response

None.

Chapter 6. JCL, command examples, and reference

This topic provides JCL, user examples, and common error messages for Encryption Facility for OpenPGP:

- "Sample JCL and code" on page 179
- "Examples of commands for Encryption Facility for OpenPGP" on page 182
- "Common error messages" on page 186

Sample JCL and code

Sample JCL to invoke the Java batch program:

```
//*********************************************************************
//* Licensed Materials - Property of IBM
//* 5655-P97 Copyright IBM Corp. 2007
//* Status = HCF7740
//*
//* It is recommended to use IBM JZOS Batch Toolkit for z/OS to invoke
//* the OpenPGP support.
//* The JZOS invocation samples provided by Encryption Facility V1.2
//* consist of three different files:
//* 1. Procedure in PROCLIB
//* 2. Shell script to configure environment variables
//* 3. Batch job that calls the sample procedure in PROCLIB
//*
//* This is a sample procedure used by the sample batch job
//* to launch the Encryption Facility V1.2 OpenPGP support.
//*
//* To use this sample, tailor the procedure to your installation:
//* 1.) Replace '<high-level qualifier>.JZOS.LOADLIB' with the PDSE that contains the
//*     JVMLDMxx modules that were installed during installation
//* 2.) The ARGS parameter should not updated.  Instead update the
//*     MAINARGS DD in the calling DD.
//*
//*********************************************************************
//CSDJZSVM PROC JAVACLS='com.ibm.encryptionfacility.EFOpenPGP',
//   ARGS='',                            < Args to Java class
//   LIBRARY='<high-level qualifier>.JZOS.LOADLIB',    <STEPLIB FOR JVMLDM module
//   VERSION='50',                       < JVMLDM version: 50,56
//   LOGLVL='+I',                       <JZOS Dbg LVL: +I(info) +T(trc)
//   REGSIZE='0M',                       <EXECUTION REGION SIZE
//   LEPARM=''
//JAVAJVM  EXEC PGM=JVMLDM&VERSION,REGION=&REGSIZE,
//   PARM='&LEPARM/&LOGLVL &JAVACLS &ARGS'
//STEPLIB  DD DSN=&LIBRARY,DISP=SHR
//SYSPRINT DD SYSOUT=*                   <System stdout
//SYSOUT   DD SYSOUT=*                  <System stderr
//STDOUT   DD SYSOUT=*                  <Java System.out
//STDERR   DD SYSOUT=*                  <Java System.err
//CEEDUMP  DD SYSOUT=*
//ABNLIGNR DD DUMMY
//*
//*The following DDs can/should be present in the calling JCL
//*
//*STDIN   DD                       <OPTIONAL - Java System.in
//*STDENV  DD                         <REQUIRED - JVM Environment script
//*MAINARGS DD                          <Preferred method to supply args
// PEND
```

Sample code for the Java environment script to configure any environment variables for the Java JVM:

```
# Licensed Materials - Property of IBM
# 5655-P97 Copyright IBM Corp. 2007
# Status = HCF7740
#
```

```
# It is recommended to use IBM JZOS Batch Toolkit for z/OS to invoke
# the OpenPGP support.
# The JZOS invocation samples provided by Encryption Facility V1.2
# consist of three different files:
# 1. Procedure in PROCLIB
# 2. Shell script to configure environment variables
# 3. Batch job that calls the sample stored procedure
#
# This is a sample shell script which configures
# any environment variables for the Java JVM.
# Variables must be exported to be seen by the launcher.
#
# To use this sample, tailor the script to your installation:
# 1.) Replace <JAVA_HOME> to point the location of the 5.0 JDK

. /etc/profile
export JAVA_HOME=<JAVA_HOME>
export JZOS_HOME="${JAVA_HOME}"/lib/ext/

export PATH=/bin:"${JAVA_HOME}"/bin:

LIBPATH=/usr/lib/java_runtime:/lib:/usr/lib:"${JAVA_HOME}"/bin
LIBPATH="$LIBPATH":"${JAVA_HOME}"/bin/classic
LIBPATH="$LIBPATH":"${JZOS_HOME}"
export LIBPATH="$LIBPATH":

# Customize your CLASSPATH here
CLASSPATH=/usr/include/java_classes/ifaedjreg.jar
CLASSPATH=$CLASSPATH:/usr/lpp/encryptionfacility/CSDEncryptionFacility.jar

# Add JZOS required jars to end of CLASSPATH
for i in "${JZOS_HOME}"/*.jar; do
    CLASSPATH="$CLASSPATH":"$i"
    done
export CLASSPATH="$CLASSPATH":

# Set JZOS specific options
# Use this variable to specify encoding for DD STDOUT and STDERR
#export JZOS_OUTPUT_ENCODING=Cp1047
# Use this variable to prevent JZOS from handling MVS operator commands
#export JZOS_ENABLE_MVS_COMMANDS=false
# Use this variable to supply additional arguments to main
#export JZOS_MAIN_ARGS=""

# Configure JVM options
IJO="-Xms16m -Xmx128m"
# Configure the number of garbage collection treads during execution
IJO="$IJO -Xgcthreads4"
# Uncomment the following to aid in debugging "Class Not Found" problems
#IJO="$IJO -verbose:class"
IJO="$IJO -Djzos.home=${JZOS_HOME}"
# Uncomment the following if you want to run without JIT
#IJO="$IJO -Djava.compiler=NONE"
# Uncomment the following if you want to run with Ascii file encoding.
#IJO="$IJO -Dfile.encoding=ISO8859-1"
export IBM_JAVA_OPTIONS="$IJO -Dibm.DES.usehdwr.size=0"
# Uncomment the following if you want to run with trace from hardware crypto
# provider
#export IBM_JAVA_OPTIONS="$IBM_JAVA_OPTIONS -Djava.security.auth.debug=all"

# Uncomment the following if you want to run with trace from the JZOS
# component
#export IBM_JAVA_OPTIONS="$IBM_JAVA_OPTIONS -Djzos.logging=T"
#export IBM_JAVA_OPTIONS="$IBM_JAVA_OPTIONS -Djzos.merge.sysout=true"

export JAVA_DUMP_HEAP=false
export IBM_JAVA_ZOS_TDUMP=NO

# Required to correctly read ASCII armor data sets since ASCII armor data sets
# contain some 0 byte records
export _EDC_ZERO_RECLEN=Y
```

Sample JCL that uses the Java batch program and environment script (Note: Sample includes the following steps):

1. Encrypt a data set with a passphrase.
2. Decrypt a data set with a passphrase.
3. Encrypt a data set with public key.

In order for step 3 (//JAVA3) to run, you must use the `-g` command with the following options to make the key alias available. This sample is run from the shell script environment. Also, ensure that you set up the Java environment to use larger key sizes. See the z/OS Java home page (www.ibm.com/docs/en/semeru-runtime-ce-z).:

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar   \
-homedir /etc/encryptionfacility  \
-key-alias rsa_md2_4096 \
-keystore /var/encryptionfacility/keystores/encrdecr/keystore_jceks \
-keystore-type JCEKS \
-key-size 4096 \
-keystore-password password \
-key-password password \
-g
```

Sample code for encrypting and decrypting z/OS data sets:

```
//CSDSMJCL  JOB ()
//PROCLIB JCLLIB ORDER=<HLQ>.JZOS.JCL
//*
//*********************************************************************
//* Licensed Materials - Property of IBM
//* 5655-P97 Copyright IBM Corp. 2007
//* Status = HCF7740
//*
//* It is recommended to use IBM JZOS Batch Toolkit for z/OS to invoke
//* the OpenPGP support.
//* The JZOS invocation samples provided by Encryption Facility V1.2
//* consist of three different files:
//* 1. Procedure in PROCLIB
//* 2. Shell script to configure environment variables
//* 3. Batch job that calls the sample stored procedure
//*
//* This is a sample batch job to launch the Encryption Facility V1.2
//* OpenPGP support.
//* Tailor the job for your installation:
//* 1.) Modify the job card per your installation's requirements
//* 2.) Replace '<HLQ>.JZOS.JCL(CSDSMPEN )' with the PDS that contains
//*     the shell script to update the JVM's environment variables
//* 3.) Replace '<HLQ>.JZOS.JCL' with the PDS that contains the
//*     sample procedure CSDJZSVM
//* 4.) Update the MAINARGS DD to specify options and commands for the
//*     OpenPGP support invocation.  Refer to the user's guide for the
//*     correct syntax for specifying the options and commands for an
//*     invocation.
//*
//* This sample job contains example invocations across three steps:
//* JAVA1-Encrypt a data set with password
//* JAVA2-Decrypt a data set with password
//* JAVA3-Encrypt a text data set with public key
//* The sample steps use the following data sets:
//*  HLQ.EFR2.ENC.OUT - allocated in DD
//*  HLQ.EFR2.ENC.OUT2 - allocated in DD
//*  HLQ.EFR2.INPUT(CLRTXT) - assumed to exist
//*  HLQ.EFR2.DEC.OUT - allocated in DD
//*
//* JAVA3 assumes the existence of a keystore that contains an X.509
//* certificate for alias rsa_md2_4096.
//*
//* The -s2k-passphrase option is shown here for simplicity. It is not
//* recommended to include your passphrase in the JCL.  Instead, update
//* your ibmef.config file to include the passphrase (keyword
//* S2K_PASSPHRASE) and maintain proper access control on the file.
//*
//JAVA1 EXEC PROC=CSDJZSVM,VERSION='50'
//STDENV DD DSN=<HLQ>.JZOS.JCL(CSDSMPEN ),DISP=SHR
//*
//DDDEF DD DSN=HLQ.EFR2.ENC.OUT,
//        DISP=(NEW,KEEP),
//        DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760),
//        UNIT=SYSALLDA,
//        SPACE=(CYL,(5,1))
//*
//MAINARGS  DD *
-homedir /etc/encryptionfacility/
-o '//DD:DDDEF'
-s2k-passphrase PASSW0RD
-c '//HLQ.EFR2.INPUT(CLRTXT)'
/*
```

```
//JAVA2 EXEC PROC=CSDJZSVM,VERSION='50'
//STDENV DD DSN=<HLQ>.JZOS.JCL(CSDSMPEN),DISP=SHR
//DDDEF DD DSN=HLQ.EFR2.DEC.OUT,
//         DISP=(NEW,CATLG),
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160),
//         UNIT=SYSALLDA,
//         SPACE=(CYL,(5,1))
//*
//MAINARGS  DD *
-homedir /etc/encryptionfacility/
-o '//DD:DDDEF'
-s2k-passphrase PASSW0RD //HLQ.EFR2.ENC.OUT
/*
//JAVA3 EXEC PROC=CSDJZSVM,VERSION='50'
//STDENV DD DSN=<HLQ>.JZOS.JCL(CSDSMPEN),DISP=SHR
//*
//DDDEF DD DSN=HLQ.EFR2.ENC.OUT2,
//         DISP=(NEW,CATLG),
//         DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760),
//         UNIT=SYSALLDA,
//         SPACE=(CYL,(5,1))
//*
//MAINARGS  DD *
-homedir /etc/encryptionfacility/
-o 'DD:DDDEF'
-rA rsa_md2_4096
-keystore /var/encryptionfacility/keystores/encrdecr/keystore_jceks
-keystore-type JCEKS
-keystore-password password
-key-password password
-t 'UTF-8'
-e '//HLQ.EFR2.INPUT(CLRTXT)'
/*
```

Examples of commands for Encryption Facility for OpenPGP

In each of these command examples, the jar file for Encryption Facility for OpenPGP is `CSDEncryptionFacility.jar`. Commands are issued from UNIX System Services.

If you are using the triple DES algorithm with hardware, you must include the following line as the first line of code:

```
java -Dibm.DES.usehdwr.size=0
```

Obtaining help

Obtain help information for all commands:

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-h
```

Listing algorithms

List all available algorithms for encryption:

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-list-algo
```

Deleting a certificate by user ID

Delete a PGP certificate with user ID `test_user`:

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-keystore-type JKS
-keystore /var/encryptionfacility/keystores/example.jks
-keystore-password abcd1234
-xP test_user
```

Deleting a certificate by key ID

Delete OpenPGP certificates and X.509 certificates with key ID `DBDE74D86844CF02`:

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-keystore-type JKS
-keystore /var/encryptionfacility/keystores/example.jks
-keystore-password abcd1234
-xK DBDE74D86844CF02
```

Encrypting a PDSE with PBE using the triple DES cryptographic algorithm

Use password-based encryption (PBE) to encrypt PDSE member `EFV2.ENCRDECR.DATA.PDSEVB(INPUT)` with the triple DES algorithm:

```
java -Dibm.DES.usehdwr.size=0
-jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-homedir /var/encryptionfacility/configs/encrdecr/symmetric
-o //EFV2.OUTPUT.ENCRDECR.TDES.ENC(ZLIB0)
-z 1
-cipher-name TRIPLE_DES
-compress-name ZLIB
-s2k-passphrase password
-c
//EFV2.ENCRDECR.DATA.PDSEVB(INPUT)
```

Encrypting a PDSE using multiple aliases

Encrypt PDSE member `EFV2.ENCRDECR.DATA.PDSEVB(INPUT)` using multiple recipient aliases and store in PDSE `EFV2.OUTPUT.ENCRDECR.JKS.ENC(LOTS)`:

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-homedir /var/encryptionfacility/configs/encrdecr/publickey
-o //EFV2.OUTPUT.ENCRDECR.JKS.ENC(LOTS)
-rA rsa_md5_1024,rsa_md5_2048,rsa_sha1_1024,rsa_sha1_2048,
-keystore /var/encryptionfacility/keystores/encrdecr/keystore_jks
-keystore-type JKS
-keystore-password password
-key-password password
-e
//EFV2.ENCRDECR.DATA.PDSEVB(INPUT)
```

Decrypting a PDSE member

Decrypt PDSE member `EFV2.OUTPUT.ENCRDECR.JKS.ENC(LOTS)`:

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-homedir /var/encryptionfacility/configs/encrdecr/publickey
-o //EFV2.OUTPUT.ENCRDECR.JKS(LOTS)
-keystore /var/encryptionfacility/keystores/encrdecr/keystore_jks
-keystore-type JKS
-keystore-password password
-key-password password
-d
//EFV2.OUTPUT.ENCRDECR.JKS.ENC(LOTS)
```

Exporting an alias from the Java keystore

Export alias `dsa1024gen558` from the Java keystore:

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-homedir /etc/encryptionfacility
-eA dsa1024gen558
```

Exporting a key ID from the Java keystore or OpenPGP keyring

Export key ID `011601CE36231FF2` from the Java keystore or OpenPGP keyring:

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-homedir /etc/encryptionfacility
-eK 011601CE36231FF2
```

Exporting a user ID from the OpenPGP keyring to an output file

Export user ID `dsa1024gen310` from the OpenPGP keyring to the output file `/var/encryptionfacility/output/exptemp.out`:

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-homedir /etc/encryptionfacility
-o /var/encryptionfacility/output/exptemp.out
-eP dsa1024gen310
```

Generating a key

Use key-store type JCECCAKS to generate a key. It is advisable to issue this command from a UNIX System Services platform:

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-homedir /etc/encryptionfacility
-keystore /var/encryptionfacility/keystores/hardware/JCECCAKS
-keystore-password password
-keystore-type JCECCAKS
-yes
-g
```

Importing a certificate

Import certificate `rsa2048.bin` from the Java keystore:

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-homedir /etc/encryptionfacility
-i /var/encryptionfacility/input/import/rsa2048.bin
```

Displaying aliases in the keystore

Display all of the aliases in keystore `/var/encryptionfacility/keystores/pkds/JCECCAKS`:

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-homedir /etc/encryptionfacility
-keystore /var/encryptionfacility/keystores/pkds/JCECCAKS
-keystore-type JCECCAKS
-keystore-password password
-yes
-pA
```

Displaying information about a user ID

Display information about user ID `jks_rsa_1024` in the keyring:

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-homedir /etc/encryptionfacility
-pP jks_rsa_1024
```

Displaying certificates by key ID

Display all of the certificates by key ID in both the keystore and the keyring:

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-homedir /etc/encryptionfacility
-pK
```

Preparing an existing ICSF key to use the keystore

Prepare an already existing ICSF key with label `ICSF1024NUMBER1` to use with the keystore:

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-homedir /etc/encryptionfacility
-prepare ICSF1024NUMBER1
```

Rebuilding the key-ring index

Rebuild the key-ring index files with the option **debug** activated:

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-homedir /etc/encryptionfacility
-log-file /var/encryptionfacility/logs/rebldidx/ef2ri_log.xml
-no
-debug-on
-debug -1
-debug-level 1000
-rebuild-key-index
```

Creating a signature using a signature key alias

Create a signature using the signature key alias `jks_test` and place it in the PDSE member `EFV2.OUTPUT.SIGN.OTXTPDS(EF2S0501)`:

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-homedir /etc/encryptionfacility
-keystore /var/encryptionfacility/keystores/sign/keystore_jks
-keystore-type JKS
-keystore-password abcd1234
-signers-key-alias jks_test
-signers-key-password abc01234
-digest-name SHA_1
-o //EFV2.OUTPUT.SIGN.OTXTPDS(EF2S0501)
-s //EFV2.INPUT.DSIGN.INPDS(TEXT)
```

Verifying a signature using a signature key alias

Verify the signature using the signature key alias `jks_test` in PDSE member `EFV2.OUTPUT.SIGN.OTXTPDS(EF2S0501)`

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-homedir /etc/encryptionfacility
-keystore /var/encryptionfacility/keystores/dsign/keystore_jks
-keystore-type JKS
-keystore-password abcd1234
-signers-key-alias jks_test
-signers-key-password abc01234
-digest-name SHA_1
-v //EFV2.OUTPUT.SIGN.OTXTPDS(EF2S0501)
```

Exporting an X.509 alias

Export the alias dsa1024gen558 for an X.509 certificate from the keystore and place it in the file `/var/encryptionfacility/dsa1024gen558.bin`:

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-keystore /var/encryptionfacility/sample.JKS.ks
-keystore-type JKS
-keystore-password keystorePassword
-key-password aliasPassword
```

```
-o /var/encryptionfacility/dsa1024gen558.bin
-eA dsa1024gen558
```

Exporting a key ID using ASCII Armor

Export the key ID 011601CE36231FF2 from the OpenPGP keyring and specify ASCII Armor for the output that is placed in the file /var/encryptionfacility/011601CE36231FF2.asc:

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-a
-o /var/encryptionfacility/011601CE36231FF2.asc
-eK 011601CE36231FF2
```

Exporting a user ID using ASCII Armor

Export the user ID "IBM User" from the OpenPGP keyring and specify ASCII Armor for the output that is placed in the file /var/encryptionfacility/output/ibmuser.asc:

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-a
-o /var/encryptionfacility/output/ibmuser.asc
-eP "IBM User"
```

Creating a detached signature for a z/OS partitioned data set member

Create a detached signature for the z/OS partitioned data set member EF.INPUT.DSIGN.INPDS(TEXT) and specify sequential data set EF.OUTPUT.DSIGN.OTEXSEQ as output.

The specified keystore where the system key that is to sign the data is located is keystore_jks. The keystore type is JKS. You must specify a password to access the keystore (abcd1234) and provide both an alias to the system key of this OpenPGP system (jks_alias) and a password to the system key of this OpenPGP system (12345678). The digest algorithm to use is SHA_1, and the JCE provider name is the default JCE software provider:

```
java -jar /usr/lpp/encryptionfacility/CSDEncryptionFacility.jar
-homedir /etc/encryptionfacility
-keystore /var/encryptionfacility/keystores/keystore_jks
-keystore-type JKS
-keystore-password abcd1234
-signers-key-alias jks_alias
-signers-key-password 12345678
-digest-name SHA_1
-o //EF.OUTPUT.DSIGN.OTXTSEQ
-b //EF.INPUT.DSIGN.INPDS(TEXT)
```

Common error messages

Consider the following common error messages that can occur for Encryption Facility for OpenPGP:

- **Exception in thread "main" java.lang.UnsupportedClassVersionError**: Indicates that you are not using the correct Java version.
- **Exception in thread "main" java.lang.UnsatisfiedLinkError: ifaedjreg (Not found in java.library.path)**: Indicates that you have not defined the following in your shell:

  ```
  export LIBPATH=$LIBPATH:/usr/lib/java_runtime
  ```

- The following Encryption Facility message indicates that your policy files have not been updated:

  ```
  CSD0050I Command processing ended abnormally.
  text CSD0065I No error message is available.
  Exception: java.lang.UnsupportedOperationException
  ```

- The following message is returned by Java. When this message is issued, it usually indicates that an incorrect Java keystore password has been specified or an incorrect key password has been specified. Additionally, if configured with a RACF Keyring, a keystore and key password must be specified even though they are not used by RACF. The password prompts can safely be disabled by using the DISABLE_RACF_PASSWORD_PROMPTS configuration option listed in "DISABLE_RACF_PASSWORD_PROMPTS" on page 33. If these passwords do not match, this message might be issued:

```
Given final block not properly padded
```

- Message CSD0050I Illegal Key Size usually indicates that you are using restricted Java policy files. Due to export restrictions, IBM Java Virtual Machine (JVM) is shipped with restricted policy files that limit the size of the cryptographic keys that are supported. The security files are local_policy.jar and US_export_policy.jar. To correct this, replace the /lib/security/local_policy.jar and /lib/security/US_export_policy.jar files with the unrestricted files that are found in ${java-home}/demo/jce/policy.files/unrestricted and retry the operation.

Appendix A. Accessibility

Accessible publications for this product are offered through IBM Documentation for z/OS (www.ibm.com/docs/en/zos).

If you experience difficulty with the accessibility of any z/OS documentation see How to Send Feedback to IBM to leave documentation feedback.

Notices

This information was developed for products and services that are offered in the USA or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for IBM Documentation. Use of hyperlinks in other output formats of this information is at your own risk.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Site Counsel
2455 South Road

Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at ibm.com®/privacy and IBM's Online Privacy Statement at ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at ibm.com/software/info/product-privacy.

Policy for unsupported hardware

Various z/OS elements, such as DFSMSdfp, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: IBM Lifecycle Support for z/OS (www.ibm.com/software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and Trademark information (www.ibm.com/legal/copytrade.shtml).

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

The registered trademark Linux®® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Index

Product Number: 5655-P97

SA23-2230-60