z/OS

Cryptographic Services
Integrated Cryptographic Service Facility
DK AES PIN Part 4 Support
APAR OA46466

Contents

Chapter 1. Overview

This document describes changes to the Integrated Cryptographic Service Facility (ICSF) product in support of the German Banking Industry Committee (DK) PIN methods.

New key derivation methods for the Diversified Key Generate2 (CSNBDKG2 and CSNEDKG2) callable service, key management support for new AES secure messaging keys, support for AES diversified key-generating keys to derive AES secure messaging keys, additional key-derivation sequence levels for AES diversified key-generating keys, and support for AES secure messaging and MAC keys for the DK PIN Change service is provided.

Enhancements to the Diversified Key Generate2 (CSNBDKG2 and CSNEDKG2) callable service include support for three types of AES key derivation (diversified key generation):

- Derivation of a bank specific Issuer Master Key from a banking association specific Master Key.
- Derivation of an Integrated Circuit Card (ICC) Master Key from a bank specific Issuer Master Key. This is used for Application Cryptogram generation or verification, issuer authentication, and secure messaging.
- Derivation of an ICC Session Key from an ICC Master Key. This is used for Application Cryptogram generation or verification, issuer authentication, and secure messaging.

  Note: The Diversified Key Generate2 (CSNBDKG2) service already has this method defined using diversification process keyword SESS-ENC.

The Diversified Key Generate2 (CSNBDKG2) service has new key diversification process keywords and bit length keywords. The AES DKYGENKY key type has new key-derivation sequence levels. Key Token Build2 and KGUP has new keywords for DKYGENKY keys.

These changes are available through the application of the PTF for APAR OA46466 and apply to FMID HCR77B0, HCR77A1, and HCR77A0. Application of the PTFs for ICSF APARs OA42246, OA43906, and OA44444 are prerequisites. This document contains alterations to information previously presented in the following books:

- *z/OS Cryptographic Services ICSF Application Programmer's Guide*, SC14-7508-03
- *z/OS Cryptographic Services ICSF Administrator's Guide*, SC14-7506-03

The technical changes made to the ICSF product by the application of the PTF for APAR OA46466 are indicated in this document by a vertical line to the left of the change.

Chapter 2. Update of z/OS Cryptographic Services ICSF Application Programmer's Guide, SC14-7508-03, information

This topic contains updates to the document *z/OS Cryptographic Services ICSF Application Programmer's Guide*, SC14-7508-03, for the DK AES PIN Part 4 support provided by this APAR. Refer to this source document if background information is needed.

Introducing Symmetric Key Cryptography and Using Symmetric Key Callable Services

AES key types

The AES keys are 128-bit, 192-bit, and 256-bit keys that use the AES algorithm to perform the cryptographic function.

Table 1. Descriptions of AES key types and service usage

AES key type	Usable with services
Fixed-length AES key-token, version X'04'	
DATA	Symmetric Algorithm Decipher, Symmetric Algorithm Encipher
Variable-length AES key-token, version X'05'	
Cipher class (data operation keys) These keys are used to cipher text.	
CIPHER	Symmetric Algorithm Decipher, Symmetric Algorithm Encipher, Ciphertext Translate2
Key-encrypting key class These keys are used to cipher other keys.	
EXPORTER	Key Generate2, Key Translate2, PKA Key Generate, Symmetric Key Export
IMPORTER	Key Generate2, PKA Key Generate, Key Test2, Key Translate2, Restrict Key Attribute, Secure Key Import2, Symmetric Key Import2
MAC class These keys are used to generate and verify a message authentication code (MAC).	
MAC	DK Deterministic PIN Generate, DK Migrate PIN, DK PIN Change, DK PAN Modify in Transaction, DK PAN Translate, DK PRW Card Number Update, DK PRW CMAC Generate, DK Random PIN Generate, DK Regenerate PRW, MAC Generate2, MAC Verify2
PIN class These keys are used in various financial-PIN processing services.	
PINCALC	DK Deterministic PIN Generate
PINPROT	DK Deterministic PIN Generate, DK Migrate PIN, DK PAN Translate, DK PIN Change, DK PRW Card Number Update, DK Random PIN Generate, DK Regenerate PRW

Table 1. Descriptions of AES key types and service usage (continued)

AES key type	Usable with services
PINPRW	DK Deterministic PIN Generate, DK Migrate PIN, DK PAN Modify in Transaction, DK PAN Translate, DK PIN Change, DK PIN Verify, DK PRW Card Number Update, DK Random PIN Generate, DK Regenerate PRW
Key generating class These keys are used to derive operational keys.	
DKYGENKY	Diversified Key Generate2
Secure-messaging class (data operation keys) These keys are used to encrypt keys or PINs in an EMV script.	
SECMSG	DK PIN Change

Diversifying keys

CCA supports diversifying DES and AES symmetric keys. Key-diversification is a technique often used in working with smart cards. In order to secure interactions with a population of cards, a "key-generating key" is used with some data unique to a card to derive ("diversify") keys for use with that card. The data is often the card serial number or other quantity stored on the card. The data is often public, and therefore, it is very important to handle the key-generating key with a high degree of security or else the interactions with the whole population of cards could be placed in jeopardy. CCA supports diversifying a DES key using the Diversified Key Generate callable service and diversifying an AES key using the Diversified Key Generate2 service.

Several methods of diversifying a DES key are supported. They are CLR8-ENC, TDES-ENC, TDES-DEC, SESS-XOR, TDESEMV2, TDESEMV4, and TDES-XOR:

- The CLR8-ENC and TDES-ENC methods triple-encrypt data using the *generating_key* to form the diversified key. The diversified key is then multiply-enciphered by the DES master-key modified by the control vector for the output key. The TDES-DEC method is similar except that the data is triple-decrypted.

  The TDES-ENC, TDES-CBC, and TDES-DEC methods permit the production of either another key-generating key or a final key. Control-vector bits 19 through 22 associated with the key-generating key specify the permissible type of the final key. (See DKYGENKY in Figure 21 in *ICSF Application Programmer's Guide*.) Control-vector bits 12 through 14 associated with the key-generating key specify if the diversified key is a final key or another in a series of key-generating keys. Bits 12 through 14 specify a counter that is decreased by one each time that the diversified key generate service is used to produce another key-generating key. For example, if the key-generating key that you specify has its counter set to B'010', you must specify the control vector for the *generated_key* with a DKYGENKY key type having the counter bits set to B'001' and specifying the same final key type in bits 19 through 22. Use of a *generating_key* with bits 12 through 14 set to B'000' results in the creation of the final key. Thus, you can control both the number of diversifications required to reach a final key and you can closely control the type of the final key.
- The SESS-XOR method provides a means for modifying an existing DATA, DATAC, MAC, DATAM, or MACVER, DATAMV single- or double-length key. The provided data is exclusive-ORed into the clear value of the key. This form of key diversification is specified by several of the credit card associations.

- The TDESEMV2, TDESEMV4, and TDES-XOR methods also derive a key by encrypting supplied data including a transaction counter value received from an EMV smart card. The processes are described in detail in the topic, 'Visa and EMV-related smart card formats and processes', in *ICSF Application Programmer's Guide*. See 'Working with Europay–MasterCard–Visa smart cards' in *ICSF Application Programmer's Guide* for information on the processing capabilities you can use with EMV smart cards.

Several methods of diversifying an AES key are supported. They are SESS-ENC, MK-OPTC, and KDFFM-DK:

- The SESS-ENC method of diversifying a key creates a session key by enciphering the 16 bytes of derivation data supplied with the *k*-bit AES key-generating key to produce a *k*-bit AES generated session key using the AES algorithm in ECB mode, where *k* is 128, 192, or 256.
- The MK-OPTC method creates a sequence level 0 key-generating key using the EMV Option C derivation method. This method uses AES in ECB mode to encipher the 16 bytes of derivation data with the *k*-bit diversified key generating key (Issuer Master Key) to produce a *k*-bit generated ICC master key, where *k* = 128, 192, or 256.
- The KDFFM-DK method uses a derivation method based on the NIST KDF in Feedback Mode. Note that this method is specific to the DK PIN methods. This method uses AES CMAC to generate the 16 to 40 bytes of derivation data with the *k*-bit diversified key generating key (banking association specific master key) to produce a *k*-bit generated Bank specific Issuer Master Key, where *k* = 128, 192, or 256.

Diversifying keys for DK financial applications

Derive depth (key-derivation sequence level)

The following describes how DK is expected to use the CSNBDKG2 verb to derive their keys. The finance application needs three levels of key derivation: DKYL2, DKLY1, and DKYL0. The necessary derive depths are subtypes 2, 1, and 0. First, a DKYGENKY DKYL2 level key will be used to derive a DKYGENKY DKYL1 key, and the DKYGENKY DKYL1 key will be used to derive a DKYGENKY DKYL0 key. Finally, a DKYGENKY DKYL0 key will be used to derive the final ICC session key. The key type of the ICC session key is determined by the type of key to diversify defined in the high-order byte of KUF 1 of the DKYGENKY. For DK, this key type will be MAC.

DKYL2 derives DKYL1

Banking association specific Master Key (DKYGENKY DKYL2) derives bank specific Issuer Master Key (DKYGENKY DKYL1).

1. CSNBKTB2 will be called with the new rule array keyword DKYL2 to create an internal AES DKYGENKY skeleton key token that will ultimatelyly be used to derive an AES MAC key.
2. CSNBKPI2, CSNBSKI2, or CSNBKGN2 will be called to use the AES DKYGENKY DKYL2 skeleton to import or generate the Banking association specific Master Key.
3. CSNBDKG2 will be called with the completed AES DKYGENKY DKYL2 Banking association specific Master Key as the generating key and with new diversification process rule array keyword KDFFM-DK. This keyword requires a DKYL2 key and uses the derivation method defined by DK that is based on NIST KDF in Feedback Mode to derive the DKYL1 Bank specific Master Key.

DKYL1 derives DKYL0

Bank specific Issuer Master Key (DKYGENKY DKYL1) derives ICC Master Key (DKYGENKY DKYL0).

CSNBDKG2 will be called with the derived DKYL1 Bank specific Master Key as the generating key and with new diversification process rule array keyword MK-OPTC. This keyword requires a DKYL1 key, and uses the derivation method defined by EMV called Option C.

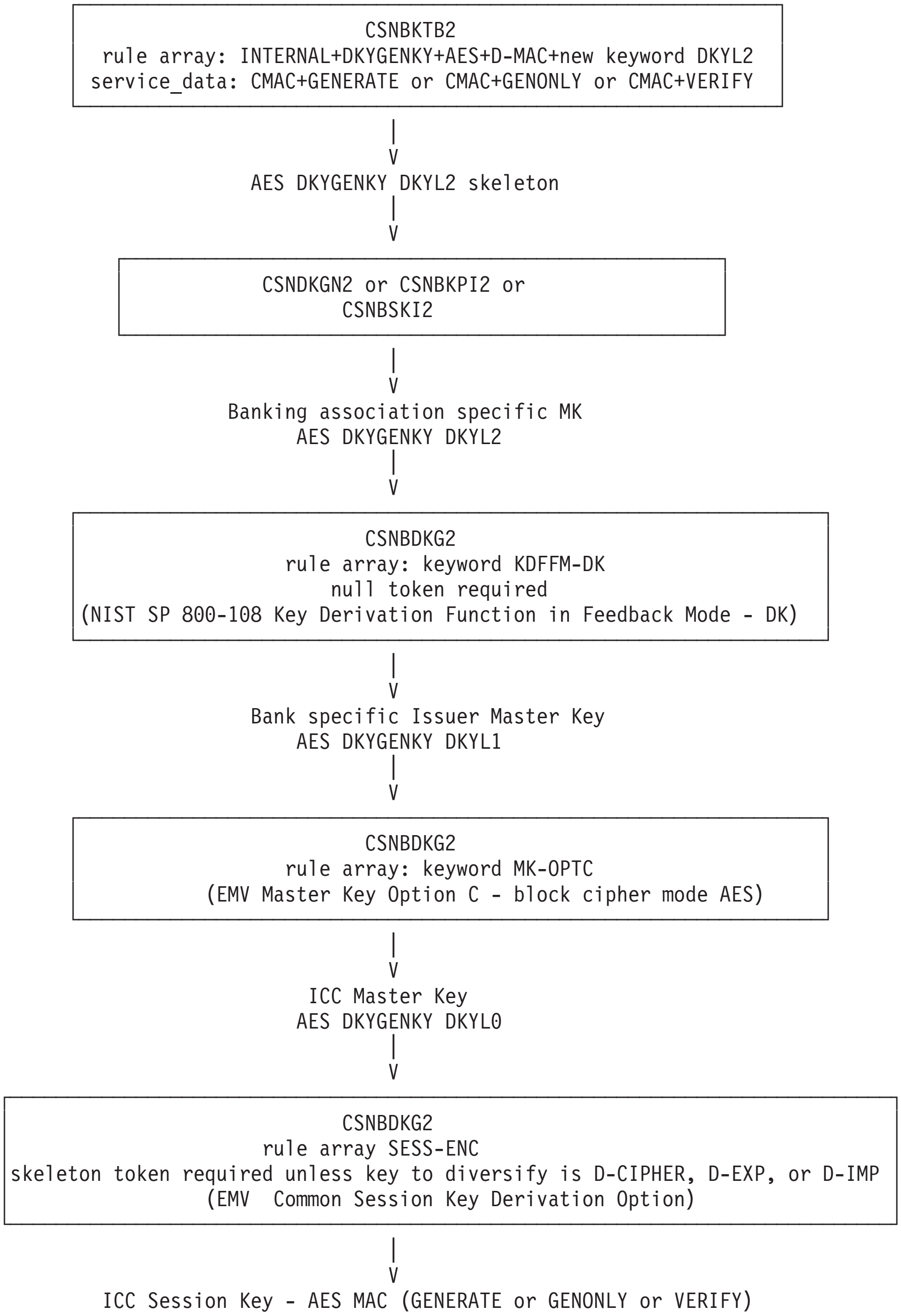
DKYL0 derives final ICC Session Key

ICC Master Key (DKYGENKY DKYL0) derives final operational ICC Session Key (AES MAC) used for Application Cryptogram generation or verification, issuer authentication, and secure messaging.

CSNBDKG2 will be called with the derived DKLY0 ICC Master Key as the generating key and with existing diversification process rule array keyword SESS-ENC. This keyword requires a DKYL0 key and uses the AES derivation method defined by EMV called Common Session Key Derivation Option.

Summary of key-derivation process

The following figure summarizes the process that the banks are expected to use to perform the three required key-derivation processes:


CSNBKTB2
rule array: INTERNAL+DKYGENKY+AES+D-MAC+new keyword DKYL2
service_data: CMAC+GENERATE or CMAC+GENONLY or CMAC+VERIFY
AES DKYGENKY DKYL2 skeleton
CSNDKGN2 or CSNBKPI2 or
CSNBSKI2
Banking association specific MK
AES DKYGENKY DKYL2
CSNBDKG2
rule array: keyword KDFFM-DK
null token required
(NIST SP 800-108 Key Derivation Function in Feedback Mode - DK)
Bank specific Issuer Master Key
AES DKYGENKY DKYL1
CSNBDKG2
rule array: keyword MK-OPTC
(EMV Master Key Option C - block cipher mode AES)
ICC Master Key
AES DKYGENKY DKYL0
CSNBDKG2
rule array SESS-ENC
skeleton token required unless key to diversify is D-CIPHER, D-EXP, or D-IMP
(EMV Common Session Key Derivation Option)
ICC Session Key - AES MAC (GENERATE or GENONLY or VERIFY)

Managing Symmetric Cryptographic Keys

Diversified Key Generate2 Callable Service (CSNBDKG2 and CSNEDKG2)

The Diversified Key Generate2 callable service generates an AES key based on a function of a key-generating key, the process rule, and data that you supply.

To use this service, specify:

- The rule array keyword to select the diversification process.
- The operational AES key-generating key from which the diversified keys are generated.

  For a key-generating key with a key-derivation sequence level of 1 or 2:
  The type of key created will be a DKYGENKY key with a sequence level one lower than the key-generating key with the same key usage fields.

  For a key-generating key with a key-derivation sequence level of 0:
  Key usage field 1 determines the type of key that is generated and restricts the use of this key to the key-diversification process. If the generating key has related key usage fields 3 through field 6 defined, these key usage attributes are used to control the permitted key usage attributes for the key to be generated.

  Note: Key usage field 2 of the generating DKYGENKY key contains a flag in its high-order byte. This flag byte determines how key usage fields 3 and beyond (called the related generated key usage fields) are used to control the values of the key usage fields of the generated key:

  - When the type of key to diversify is D-ALL, the flag is undefined because there are no key usage restrictions on the generated key. The generating key has no related generated key usage fields.
  - When the type of key to diversify is not D-ALL and the flag byte has KUF-MBE usage, the key usage fields of the key to be generated must be equal to the related generated key usage fields that start with key usage field 3 of the generating key.
  - When the type of key to diversify is not D-ALL and the flag byte has KUF-MBP usage, the key usage fields of the key to be generated must be permissible. In other words, a key to be diversified is only permitted to have a level of usage less than or equal to the related key usage fields (key usage fields starting with key usage field 3). One exception is that the UDX-only setting of the generated key always must be equal to the UDX-ONLY setting of the generating key.
- The diversification data and length of data used in the diversification process.
- The variable-length AES symmetric-key generated token with a suitable key type and key usage fields for receiving the diversified key, or a null key token if the type of key to diversify supports default key usage and a default key is desired.

The callable service name for AMODE(64) invocation is CSNEDKG2.

Format

```
CALL CSNBDKG2(
            return_code,
            reason_code,
            exit_data_length,
            exit_data,
            rule_array_count,
            rule_array,
            generating_key_identifier_length,
            generating_key_identifier,
            derivation_data_length,
            derivation_data,
            reserved1_length,
            reserved1,
            reserved2_length,
            reserved2,
               generated_key_identifier1_length,
               generated_key_identifier1,
               generated_key_identifier2_length,
               generated_key_identifier2)
```

Parameters

return_code

Direction	Type
Output	Integer

The return code specifies the general result of the callable service.

reason_code

Direction	Type
Output	Integer

The reason code specifies the result of the callable service that is returned to the application program. Each return code has different reason codes assigned to it that indicate specific processing problems.

exit_data_length

Direction	Type
Input/Output	Integer

The length of the data that is passed to the installation exit. The data is identified in the *exit_data* parameter.

exit_data

Direction	Type
Input/Output	String

The data that is passed to the installation exit.

rule_array_count

Direction	Type
Input	Integer

The number of keywords you supplied in the *rule_array* parameter. The value must be 1 or 2.

rule_array

Direction	Type
Input	String

Keywords that provide control information to the callable service. The keywords must be in contiguous storage with each of the keywords left-justified in its own 8-byte location and padded on the right with blanks.

Table 2. Rule array keywords for diversified key generate2

Keyword	Meaning
Diversification Process (required)	
KDFFM-DK	Specifies to use the DK version of key derivation function in feedback mode. This method uses AES CMAC to encipher the 16 to 40 bytes of derivation data with the *k*-bit diversified key generating key (banking association specific master key) to produce a *k*-bit generated bank specific Issuer Master Key, where *k* = 128, 192, or 256.
MK-OPTC	Specifies to use the EMV master key derivation option C specified in *EMV Integrated Circuit Card Specifications for Payments Systems*. This method uses AES in ECB mode to encipher the 16 bytes of derivation data with the *k*-bit diversified key generating key (Issuer Master Key) to produce a *k*-bit generated ICC master key, where *k* = 128, 192, or 256.
SESS-ENC	Specifies to use the EMV common session key derivation option specified in *EMV Integrated Circuit Card Specifications for Payments Systems*. This method uses AES in ECB mode to encipher the 16 bytes of derivation data with the *k*-bit diversified key generating key (ICC master key) to produce a *k*-bit generated key (ICC session key), where *k* = 128, 192, or 256.
Bit length of generated key (one, optional). Valid only with the KDFFM-DK keyword. Default is to use the bit length of the generating key as the bit length of the generated key.	
KLEN128	Specifies the bit length of the generated key to be 128.
KLEN192	Specifies the bit length of the generated key to be 192, allowed if and only if the bit length of the generating key is greater than or equal to 192.
KLEN256	Specifies the bit length of the generated key to be 256, allowed if and only if the bit length of the generating key is 256.

generating_key_identifier_length

Direction	Type
Input	Integer

Specifies the length in bytes of the *generating_key_identifier* parameter. If the *generating_key_identifier* contains a label, the value must be 64. Otherwise, the value must be between the actual length of the token and 725.

generating_key_identifier

Direction	Type
Input/Output	String

The identifier of the key-generating key. The key identifier is an operational token or the key label of an operational token in key storage. The key algorithm of this key must be AES and the key type must be DKYGENKY. The key usage field indicates the key type of the generated key. The key length determines the length of the generated key.

If SESS-ENC is specified, the clear length of the generated key is equal to the clear length of the generating key. If SESS-ENC is specified, the key-derivation sequence level must be set to DKYL0 in the key usage field 2.

If the token supplied was encrypted under the old master key, the token is returned encrypted under the current master key.

derivation_data_length

Direction	Type
Input	Integer

Specifies the length in bytes of the *derivation_data* parameter. If SESS-ENC or MK-OPTC is specified, the value must be 16. If KDFFM-DK is specified, the value may be between 16 to 40 inclusive.

derivation_data

Direction	Type
Input	String

The derivation data to be used in the key generation process. This data is often referred to as the diversification data. For SESS-ENC, the derivation data is 16-bytes long.

Note that if SESS-ENC is specified and the length of the key generating key is 192 bits or 256 bits, the data is manipulated in conformance with the EMV Common Session Key Derivation Option.

reserved1_length

Direction	Type
Input	Integer

Length in bytes of the *reserved1* parameter. The value must be 0.

reserved1

Direction	Type
Input	String

This field is ignored.

reserved2_length

Direction	Type
Input	Integer

Length in bytes of the *reserved2* parameter. The value must be 0.

reserved2

Direction	Type
Input	String

This field is ignored.

generated_key_identifier1_length

Direction	Type
Input/Output	Integer

On input, the length of the buffer for the *generated_key_identifier1* parameter in bytes. The maximum value is 725 bytes.

On output, the parameter holds the actual length of the *generated_key_identifier1* parameter.

generated_key_identifier1

Direction	Type
Input/Output	String

The buffer for the generated key token.

On input, the buffer contains a null token or a valid internal skeleton token containing the desired key-usage fields and key-management fields you want to generate. The key token must be left justified in the buffer.

The generating key (*generating_key_identifier* parameter) determines whether on input the *generated_key_identifier1* parameter can identify a null key token or a skeleton key token.

Table 3. Summary of input generating key tokens, input generated key tokens, and output generated key tokens

Input generating key token	Input generated key token	Output generated key token
DKYL0, type of key to diversify D-ALL	Skeleton key token required.	Key type same as skeleton, diversified key final.
DKYL0, type of key to diversify not D-ALL	Null or skeleton key token allowed.	Key type determined by input generated key token type of key to diversify. If null key token on input, the output key token will have attributes based on the related generated key usage fields of the input generating key token. Otherwise, the output key token will have attributes of input skeleton key token.
DKYL1, any type of key to diversify	Null key token required.	Same as input generating key token except DKYL0 and with new level of diversified key.
DKYL2, any type of key to diversify	Null key token required.	Same as input generating key token except DKYL1 and with new level of diversified key.

Table 3. Summary of input generating key tokens, input generated key tokens, and output generated key tokens (continued)

Input generating key token	Input generated key token	Output generated key token
Notes: 1. If the supplied generated key-token contains a key, the key value and length are ignored and overwritten. 2. If the *generating_key_identifier1* parameter identifies a DKYGENKY key token with a key-derivation sequence level of DKYL0 and it does not have a type of key to diversify of D-ALL, the key type must match what the generating key indicates can be created in the key generating key usage field at offset 45. 3. The key usage fields in the generated key must meet the requirements (KUF 'must be equal' or 'must be permitted') of the corresponding key usage fields in the generating key unless D-ALL is specified in the generating key. A flag bit in the DKYGENKY key-usage field 2 determines whether the key-usage field level of control is KUF-MBE or KUF-MBP. 4. If authorized by access control, D-ALL permits the derivation of several different keys.		

On output, the buffer contains the generated key token.

generated_key_identifier2_length

Direction	Type
Input/Output	Integer

Length in bytes of the *generated_key_identifier2* parameter. The value must be 0.

generated_key_identifier2

Direction	Type
Input/Output	String

This field is ignored.

Usage notes

SAF may be invoked to verify the caller is authorized to use this callable service, the key label, or internal secure key tokens that are stored in the CKDS.

Access control points

The following table shows the access control points in the domain role that control the function of this service:

Table 4. Required access control points for Diversified Key Generate2

Rule array keyword	Access control point
KDFFM-DK	Diversified Key Generate2 – KDFFM-DK
MK-OPTC	Diversified Key Generate2 - MK-OPTC
SESS-ENC	Diversified Key Generate2 – SESS-ENC

To use the KLEN192 and KLEN256 keywords, the **Diversified Key Generate2 - Allow length option with KDFFM-DK** access control point must be enabled.

If the key-generating key key-usage fields indicate that all key types may be derived, the **Diversified Key Generate2 – DALL** access control point must be enabled in the domain role.

Required hardware

This table lists the required cryptographic hardware for each server type and describes restrictions for this callable service.

Table 5. Diversified key generate2 required hardware

Server	Required cryptographic hardware	Restrictions
IBM eServer zSeries 990 IBM eServer zSeries 890		This service is not supported.
IBM System z9 EC IBM System z9 BC		This service is not supported.
IBM System z10 EC IBM System z10 BC		This service is not supported.
IBM zEnterprise 196 IBM zEnterprise 114	Crypto Express3 Coprocessor	Requires the November 2013 or later licensed internal code (LIC). Keywords KDFFM-DK, MK-OPTC, KLEN128, KLEN192, and KLEN256 are not supported.
IBM zEnterprise EC12 IBM zEnterprise BC12	Crypto Express3 Coprocessor Crypto Express4 CCA Coprocessor	Requires the September 2013 or later licensed internal code (LIC). Keywords KDFFM-DK, MK-OPTC, KLEN128, KLEN192, and KLEN256 require the June 2015 or later licensed internal code (LIC).
IBM z13	Crypto Express5 CCA Coprocessor	Keywords KDFFM-DK, MK-OPTC, KLEN128, KLEN192, and KLEN256 require the July 2015 or later licensed internal code (LIC).

Key Token Build2 (CSNBKTB2 and CSNEKTB2)

Use the Key Token Build2 callable service to build a variable-length CCA symmetric key token in application storage from information that you supply. A clear key token built by this service can be used as input for the Key Test2 callable service. A skeleton token built by this service can be used as input for the Diversified Key Generate2, Key Generate2, Key Part Import2, and Secure Key Import2 callable services.

This service will build internal or external HMAC and AES tokens, both as clear key tokens and as skeleton tokens containing no key.

The callable service name for AMODE(64) is CSNEKTB2.

Format

```
CALL CSNBKTB2(
            return_code,
            reason_code,
            exit_data_length,
            exit_data,
            rule_array_count,
            rule_array,
            clear_key_bit_length,
clear_key_value,
key_name_length,
key_name,
user_associated_data_length,
user_associated_data,
token_data_length,
token_data,
service_data_length,
service_data,
target_key_token_length,
target_key_token )
```

Parameters

return_code

Direction	Type
Output	Integer

The return code specifies the general result of the callable service.

reason_code

Direction	Type
Output	Integer

The reason code specifies the result of the callable service that is returned to the application program. Each return code has different reason codes that indicate specific processing problems.

exit_data_length

Direction	Type
Ignored	Integer

This field is ignored. It is recommended to specify 0 for this parameter.

exit_data

Direction	Type
Ignored	String

This field is ignored.

rule_array_count

Direction	Type
Input	Integer

The number of keywords you supplied in the *rule_array* parameter. The minimum value is 3, and the maximum value is 34.

rule_array

Direction	Type
Input	String

The *rule_array* contains keywords that provide control information to the callable service. The keywords must be in contiguous storage with each of the keywords left-justified in its own 8-byte location and padded on the right with blanks.

Table 6. Keywords for Key Token Build2 Control Information

Keyword	**Meaning**
Token type (one required)	
EXTERNAL	Specifies to build an external key token.
INTERNAL	Specifies to build an internal key token.
Token algorithm (one required)	
AES	Specifies to build an AES key token.
HMAC	Specifies to build an HMAC key token.
Key status (one, optional)	
KEY-CLR	Specifies to build the key token with a clear key value. This creates a key token that can be used with the Key Test2 service to generate a verification pattern for the key value.
NO-KEY	Specifies to build the key token without a key value. This creates a skeleton key token that can later be supplied to the Key Generate2 service. This is the default.
Payload version (one, optional)	
V0PYLD	Build a token with the old variable-length payload format for the target token. This is the default for AES CIPHER, EXPORTER, IMPORTER key types and is only valid with those key types.
V1PYLD	Build a token with the new fixed-length payload format for the target token. This is the default for AES MAC, PINPROT, PINCALC, PINPRW, DKYGENKY, and SECMSG key types. Not valid with the HMAC MAC key type.
Key type (one required)	
CIPHER	Specifies that this key is for data-encryption. Only valid for AES algorithm. See Figure 1 on page 19 and Table 7 on page 20 for all the valid keyword combinations and their defaults for AES key type CIPHER.
DKYGENKY	Specifies that this key is for key-generation. Only valid for AES algorithm. See Figure 6 on page 33 and Table 12 on page 34 for all the valid keyword combinations and their defaults for AES key type DKYGENKY.
EXPORTER	Specifies that this key is an EXPORTER key-encrypting key. Only valid for AES algorithm. See Figure 4 on page 27 and Table 10 on page 28 for all the valid keyword combinations and their defaults for AES key type EXPORTER.
IMPORTER	Specifies that this key is an IMPORTER key-encrypting key. Only valid for AES algorithm. See Figure 5 on page 30 and Table 11 on page 31 for all the valid keyword combinations and their defaults for AES key type IMPORTER.
MAC	Specifies that this key is for message authentication code operations. Valid for HMAC and AES algorithms. See Figure 2 on page 22 and Table 8 on page 22 for all the valid keyword combinations and their defaults for AES key type MAC. See Figure 3 on page 24 and Table 9 on page 25 for all the valid keyword combinations and their defaults for HMAC key type MAC.
PINCALC	Specifies that this key is for calculating PINs. Only valid for AES algorithm. See Figure 7 on page 38 and Table 14 on page 38 for all the valid keyword combinations and their defaults for AES key type PINCALC.

Table 6. Keywords for Key Token Build2 Control Information (continued)

Keyword	Meaning
PINPROT	Specifies that this key is for wrapping and unwrapping PIN blocks. Only valid for AES algorithm. See Figure 8 on page 40 and Table 15 on page 41 for all the valid keyword combinations and their defaults for AES key type PINPROT.
PINPRW	Specifies that this key is for generating and verifying PIN reference values. Only valid for AES algorithm. See Figure 9 on page 43 and Table 16 on page 43 for all the valid keyword combinations and their defaults for AES key type PINPRW.
SECMSG	Specifies that this key is for encrypting PINs in an EMV script. Only valid for AES algorithm. See Figure 10 on page 45 and Table 17 on page 46 for all the valid keyword combinations and their defaults for AES key type SECMSG.

clear_key_bit_length

Direction	Type
Input	Integer

The length of the clear key in bits. Specify 0 when no key value is supplied (Key status rule NO-KEY). Specify a valid key bit length when a key value is supplied (Key status rule KEY-CLR):

- For HMAC algorithm, MAC key type, this is a value between 80 and 2048.
- For AES algorithm, CIPHER/EXPORTER/IMPORTER key types, this is a value of 128, 192, or 256.

clear_key_value

Direction	Type
Input	String

This parameter is used when the KEY-CLR keyword is specified. This parameter is the clear key value to be put into the token being built.

key_name_length

Direction	Type
Input	Integer

The length of the *key_name* parameter. Valid values are 0 and 64.

key_name

Direction	Type
Input	String

A 64-byte key store label to be stored in the associated data structure of the token.

user_associated_data_length

Direction	Type
Input	Integer

The length of the user-associated data. The valid values are 0 to 255 bytes.

user_associated_data

Direction	Type
Input	String

User-associated data to be stored in the associated data structure.

token_data_length

Direction	Type
Input	Integer

This parameter is reserved. The value must be zero.

token_data

Direction	Type
Ignored	Integer

This parameter is ignored.

service_data_length

Direction	Type
Input	Integer

The length of the *service_data* parameters in bytes. For rule array keyword DKYUSAGE, the value must be a multiple of 8. Otherwise, the value must be 0. The maximum value is 280.

service_data

Direction	Type
Input	String

Data to be processed by this service when building the skeleton token. If the DKYUSAGE keyword is specified in the rule array, this parameter contains an array of key usage keywords for the type of key to be derived. The keywords are 8 bytes in length and must be left-aligned and padded on the right with blanks.

target_key_token_length

Direction	Type
Input/Output	Integer

On input, the length of the *target_key_token* parameter supplied to receive the token. On output, the actual length of the token returned to the caller. Maximum length is 725 bytes.

target_key_token

Direction	Type
Output	String

The key token built by this service.

Usage notes

The topic contains information for all key types detailing the key-usage and key-management keywords that are supported for each key type.

Figure 1 shows all the valid keyword combinations and their defaults for AES key type CIPHER. For a description of these keywords, see Table 7 on page 20.

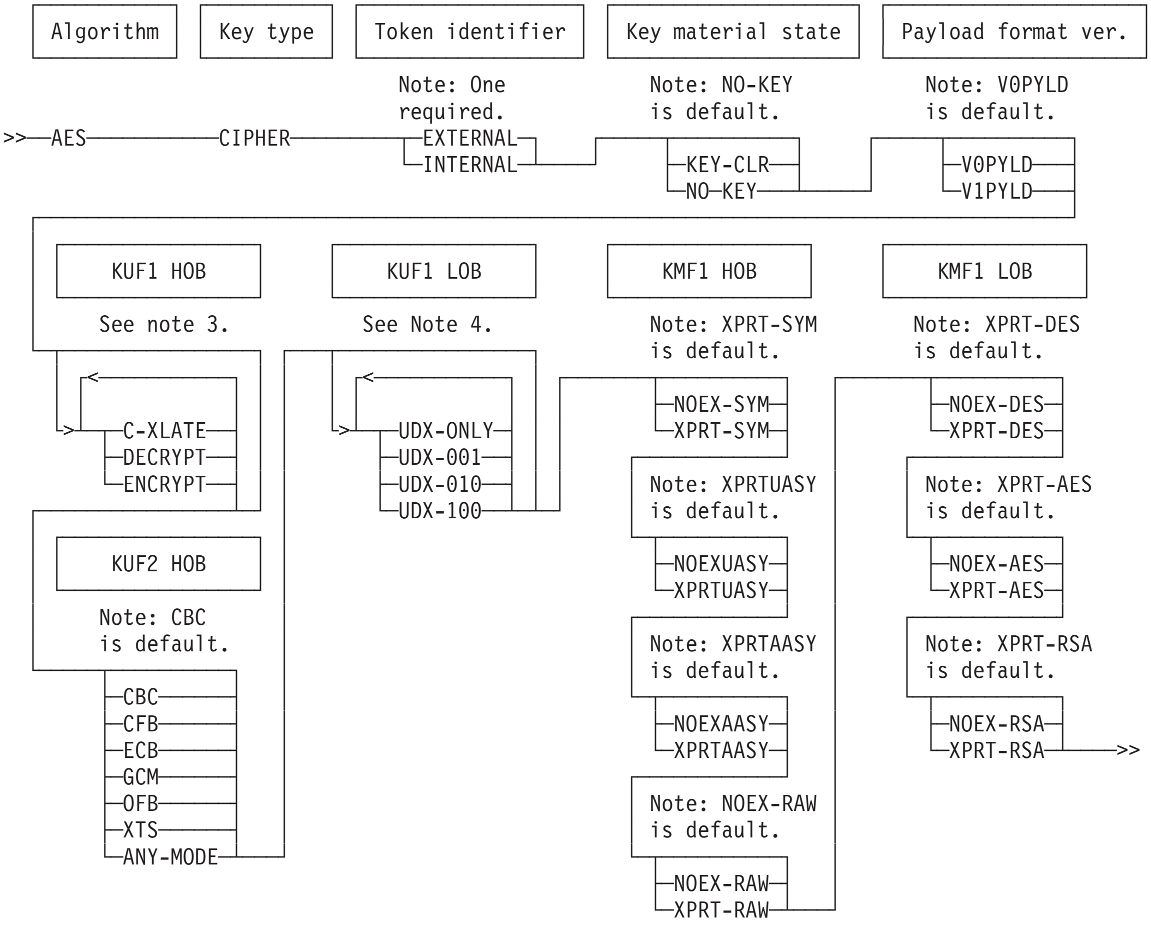


Figure 1. Key Token Build2 keyword combinations for AES CIPHER keys

Notes:

1. Keyword V0PYLD is the default for compatibility reasons. V1PYLD is recommended because it provides improved security.
2. Each key-usage field (KUF) and key-management field (KMF) of a version X'05' variable-length symmetric key-token consists of a high-order byte (HOB) and a low-order byte (LOB).
3. Keywords DECRYPT and ENCRYPT are defaults if neither of these keywords is specified, regardless of whether C-XLATE is specified or not.
4. Choose any number of keywords in this group. No keywords in the group are defaults.

5. NOEX-RAW and XPRT-RAW are defined for future use and their meanings are currently undefined. To avoid this export restriction in the future when the meaning is defined, specify XPRT-RAW.

Table 7. Rule array keywords for AES CIPHER keys

Keyword	**Meaning**
Key-token header section	
Token identifier (one required).	
EXTERNAL	Build a key token that is not to be used locally.
INTERNAL	Build a key token that is to be used locally.
Wrapping-information section	
Key status (one, optional).	
KEY-CLR	Build a key token that contains a clear key.
NO-KEY	Build a key token that does not contain a key value. This is the default.
Payload format version (one, optional). Identifies format of the payload.	
V0PYLD	Build a key token with a version 0 payload format. This format has a variable length and the key length can be inferred from the size of the payload. This format is compatible with all releases. This is the default.
V1PYLD	Build the key token with a version 1 payload format. This format has a fixed length and the key length cannot be inferred by the size of the payload. An obscured key length is considered more secure.
Associated data section	
Algorithm type (one required).	
AES	Key can be used for AES algorithm.
Key type (one required).	
CIPHER	Key can be used for encryption, decryption, and translation of data.
Encryption and translation control (one or more, optional). Key-usage field 1, high-order byte. Keywords DECRYPT and ENCRYPT are defaults unless one or more keywords in the group are specified.	
DECRYPT	Key can be used for decryption.
ENCRYPT	Key can be used for encryption.
Ciphertext Translate Control (optional). Key-usage field 1, high-order byte.	
C-XLATE	Key can be used for data translate.
User-defined extension (UDX) control (one or more, optional). Key-usage field 1, low-order byte. No keywords in the group are defaults.	
UDX-ONLY	Key can only be used in UDXs.
UDX-001	Specifies that the rightmost user-defined UDX bit is set.
UDX-010	Specifies that the middle user-defined UDX bit is set.
UDX-100	Specifies that the leftmost user-defined UDX bit is set.
Encryption mode (one, optional). Key-usage field 2, high-order byte.	
CBC	Specifies that this key can be used for cipher block chaining. This is the default.
CFB	Specifies that this key can be used for cipher feedback.
ECB	Specifies that this key can be used for electronic code book.
GCM	Specifies that this key can be used for Galois/counter mode.
OFB	Specifies that this key can be used for output feedback.
XTS	Specifies that this key can be used for Xor-Encrypt-Xor-based Tweaked Stealing.

Table 7. Rule array keywords for AES CIPHER keys (continued)

Keyword	**Meaning**
ANY-MODE	Specifies that this key can be used for any mode of encryption.
Symmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEX-SYM	Prohibits the export of the key with a symmetric key.
XPRT-SYM	Permits the export of the key with a symmetric key. This is the default.
Unauthenticated asymmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEXUASY	Prohibits the export of the key with an unauthenticated asymmetric key.
XPRTUASY	Permits the export of the key with an unauthenticated asymmetric key. This is the default.
Authenticated asymmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEXAASY	Prohibits the export of the key with an authenticated asymmetric key.
XPRTAASY	Permits the export of the key with an authenticated asymmetric key. This is the default.
RAW-key export control (one, optional). Key-management field 1, high-order byte.	
NOEX-RAW	Prohibits the export of the key in RAW format. This is the default.
XPRT-RAW	Permits the export of the key in RAW format.
DES-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-DES	Prohibits the export of the key using DES key.
XPRT-DES	Permits the export of the key using DES key. This is the default.
AES-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-AES	Prohibits the export of the key using AES key.
XPRT-AES	Permits the export of the key using AES key. This is the default.
RSA-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-RSA	Prohibits the export of the key using RSA key.
XPRT-RSA	Permits the export of the key using RSA key. This is the default.

Figure 2 on page 22 shows all the valid keyword combinations and their defaults for AES key type MAC. For a description of these keywords, see Table 8 on page 22.

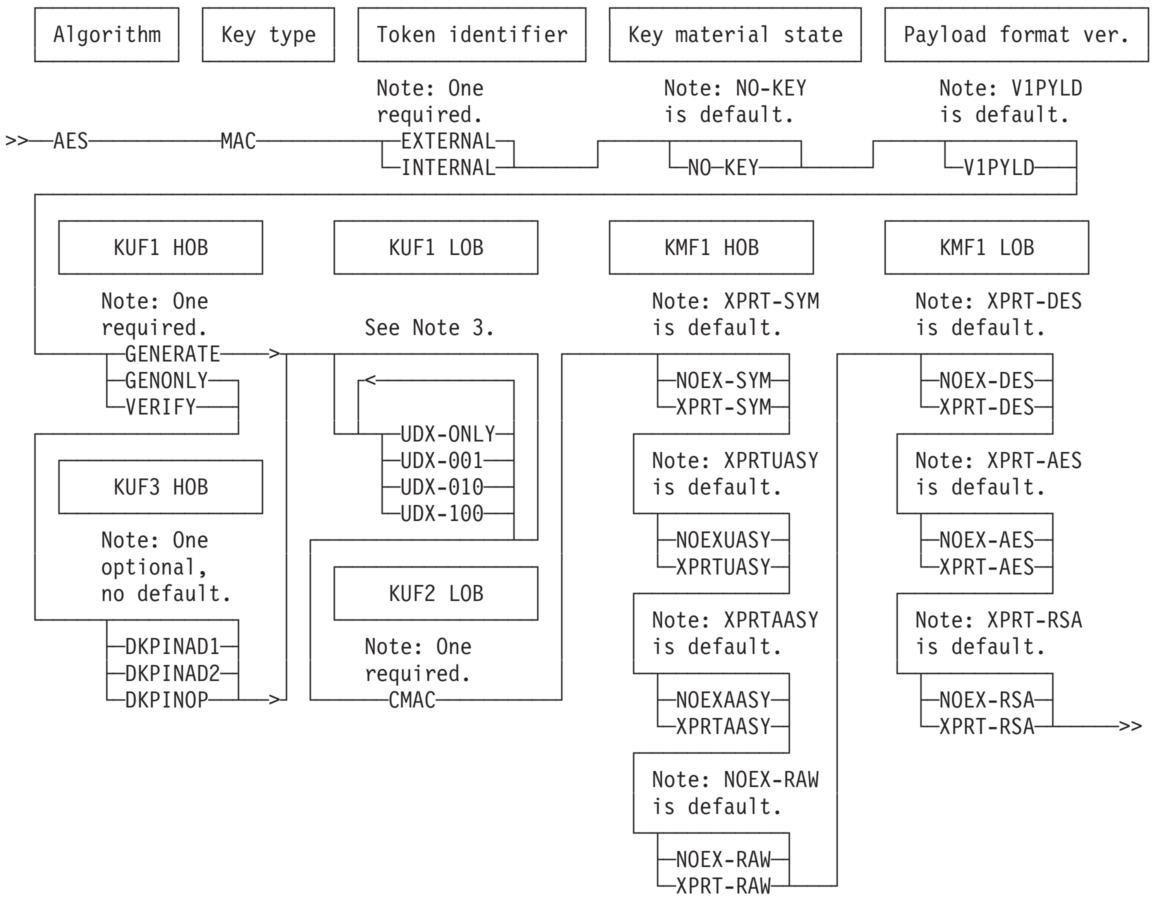


Figure 2. Key Token Build2 keyword combinations for AES MAC keys

Notes:

1. Each key-usage field (KUF) and key-management field (KMF) of a version X'05' variable-length symmetric key-token consists of two bytes: a high-order byte (HOB) and a low-order byte (LOB).
2. NOEX-RAW and XPRT-RAW are defined for future use and their meanings are currently undefined. To avoid this export restriction in the future when the meaning is defined, specify XPRT-RAW.
3. Choose any number of keywords in this group. No keywords in the group are defaults.

Table 8. Rule array keywords for AES MAC keys

Keyword	**Meaning**
Key-token header section	
Token identifier (one required).	
EXTERNAL	Build a key token that is not to be used locally.
INTERNAL	Build a key token that is to be used locally.
Wrapping-information section	
Key status (one, optional).	
NO-KEY	Build a key token that does not contain a key value. This is the default.
Payload format version (one, optional). Identifies format of the payload.	

Table 8. Rule array keywords for AES MAC keys (continued)

Keyword	Meaning
V1PYLD	Build the key token with a version 1 payload format. This format has a fixed length and the key length cannot be inferred by the size of the payload. An obscured key length is considered more secure.
Associated data section	
Algorithm type (one required).	
AES	Key can be used for AES algorithm.
Key type (one required).	
MAC	Key can be used for generation and verification of message authentication codes.
MAC control (one, required). Key-usage field 1, high-order byte.	
GENERATE	Specifies that this key can be used to generate a MAC. A key that can generate a MAC can also verify a MAC.
GENONLY	Specifies that this key can only be used to generate a MAC. It cannot be used to verify a MAC. Not valid with keywords DKPINOP, DKPINAD1, and DKPINAD2.
VERIFY	Specifies that this key cannot be used to generate a MAC. It can only be used to verify a MAC.
User-defined extension (UDX) control (one or more, optional). Key-usage field 1, low-order byte. No keywords in the group are defaults.	
UDX-ONLY	Key can only be used in UDXs.
UDX-001	Specifies that the rightmost user-defined UDX bit is set.
UDX-010	Specifies that the middle user-defined UDX bit is set.
UDX-100	Specifies that the leftmost user-defined UDX bit is set.
MAC mode (one, required). Key-usage field 2, high-order byte.	
CMAC	Key can be used for block cipher-based MAC algorithm, called CMAC (NIST SP 800-38B).
Common control (one, optional). Key-usage field 3, high-order byte. Use of a common control keyword causes key-usage field 3, low-order byte (field format identifier at token offset 050) to be set to X'01' (DK enabled).	
DKPINAD1	Specifies that this key may be used to create or verify a pin block to allow the changing of the account number associate with a PIN.
DKPINAD2	Specifies that this key may be used to create or verify an account change string to allow the changing of the account number associated with a PIN.
DKPINOP	Specifies that this key may be used as a general-purpose key. It may not be used as a special-purpose key.
Symmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEX-SYM	Prohibits the export of the key with a symmetric key.
XPRT-SYM	Permits the export of the key with a symmetric key. This is the default.
Unauthenticated asymmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEXUASY	Prohibits the export of the key with an unauthenticated asymmetric key.
XPRTUASY	Permits the export of the key with an unauthenticated asymmetric key. This is the default.
Authenticated asymmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEXAASY	Prohibits the export of the key with an authenticated asymmetric key.
XPRTAASY	Permits the export of the key with an authenticated asymmetric key. This is the default.
RAW-key export control (one, optional). Key-management field 1, high-order byte.	

Table 8. Rule array keywords for AES MAC keys (continued)

Keyword	Meaning
NOEX-RAW	Prohibits the export of the key in RAW format. This is the default.
XPRT-RAW	Permits the export of the key in RAW format.
DES-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-DES	Prohibits the export of the key using DES key.
XPRT-DES	Permits the export of the key using DES key. This is the default.
AES-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-AES	Prohibits the export of the key using AES key.
XPRT-AES	Permits the export of the key using AES key. This is the default.
RSA-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-RSA	Prohibits the export of the key using RSA key.
XPRT-RSA	Permits the export of the key using RSA key. This is the default.

Figure 3 shows all the valid keyword combinations and their defaults for HMAC key type MAC. For a description of these keywords, see Table 9 on page 25.

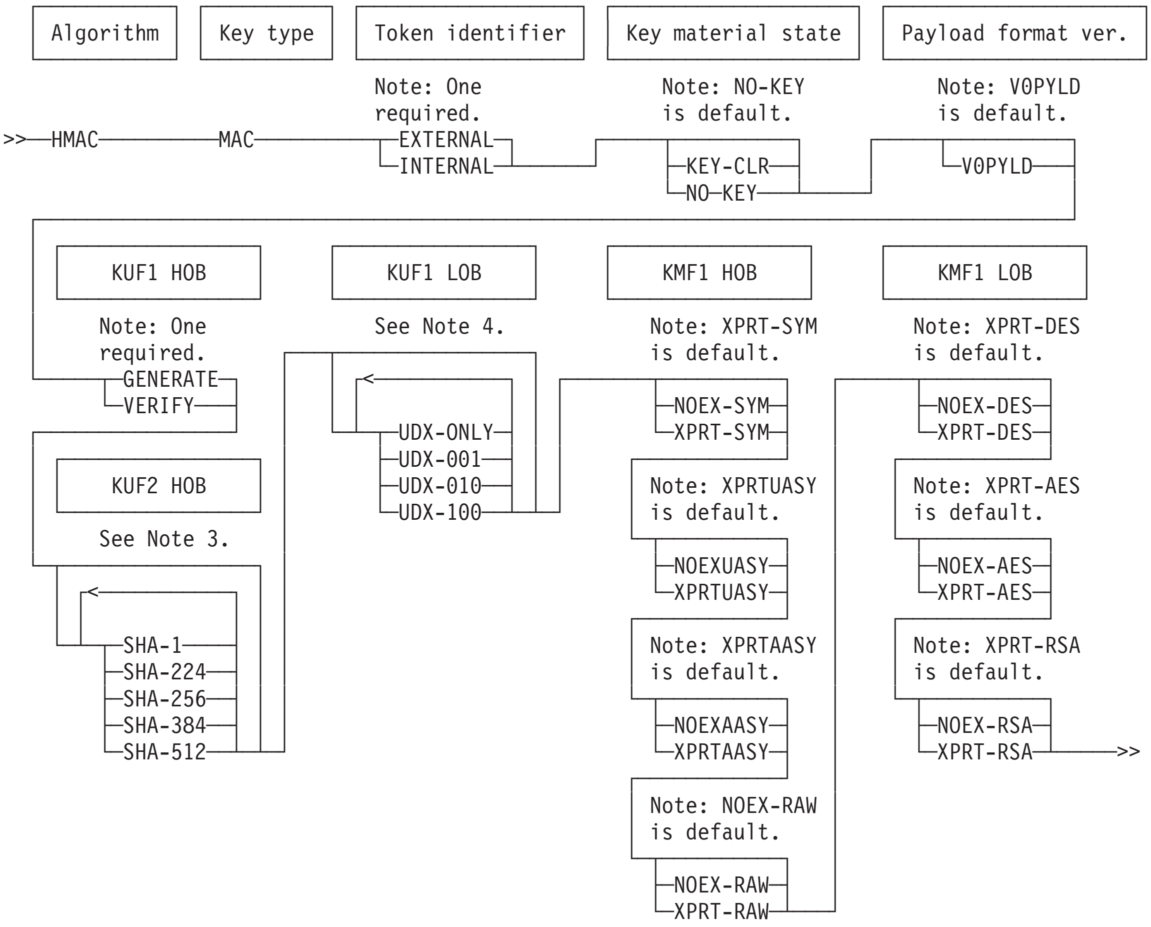


Figure 3. Key_Token_Build2 keyword combinations for HMAC MAC keys

Notes:

1. Each key-usage field (KUF) and key-management field (KMF) of a version X'05' variable-length symmetric key-token consists of two bytes: a high-order byte (HOB) and a low-order byte (LOB).
2. NOEX-RAW and XPRT-RAW are defined for future use and their meanings are currently undefined. To avoid this export restriction in the future when the meaning is defined, specify XPRT-RAW.
3. All keywords in the group are defaults unless one or more keywords in the group are specified.
4. Choose any number of keywords in this group. No keywords in the group are defaults.

Table 9. Rule array keywords for HMAC MAC keys

Keyword	**Meaning**
Key-token header section	
Token identifier (one required).	
EXTERNAL	Build a key token that is not to be used locally.
INTERNAL	Build a key token that is to be used locally.
Wrapping-information section	
Key status (one, optional).	
KEY-CLR	Build a key token that contains a clear key.
NO-KEY	Build a key token that does not contain a key value. This is the default.
Payload format version (one, optional). Identifies format of the payload.	
V0PYLD	Build a key token with a version 0 payload format. This format has a variable length and the key length can be inferred from the size of the payload. This is the default.
Associated data section	
Algorithm type (one required).	
HMAC	Key can be used for HMAC algorithm.
Key type (one required).	
MAC	Key can be used for generation and verification of message authentication codes.
MAC control (one, required). Key-usage field 1, high-order byte.	
GENERATE	Specifies that this key can be used to generate a MAC. A key that can generate a MAC can also verify a MAC.
VERIFY	Specifies that this key cannot be used to generate a MAC. It can only be used to verify a MAC.
User-defined extension (UDX) control (one or more, optional). Key-usage field 1, low-order byte. No keywords in the group are defaults.	
UDX-ONLY	Key can only be used in UDXs.
UDX-001	Specifies that the rightmost user-defined UDX bit is set.
UDX-010	Specifies that the middle user-defined UDX bit is set.
UDX-100	Specifies that the leftmost user-defined UDX bit is set.
Hash method control (any combination, optional). Key-usage field 2, high-order byte. Note: All keywords in the list below are defaults unless one or more keywords in the list are specified.	
SHA-1	Specifies that the SHA-1 hash method is allowed for the key.
SHA-224	Specifies that the SHA-224 hash method is allowed for the key.
SHA-256	Specifies that the SHA-256 hash method is allowed for the key.

Table 9. Rule array keywords for HMAC MAC keys (continued)

Keyword	**Meaning**
SHA-384	Specifies that the SHA-384 hash method is allowed for the key.
SHA-512	Specifies that the SHA-512 hash method is allowed for the key.
Symmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEX-SYM	Prohibits the export of the key with a symmetric key.
XPRT-SYM	Permits the export of the key with a symmetric key. This is the default.
Unauthenticated asymmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEXUASY	Prohibits the export of the key with an unauthenticated asymmetric key.
XPRTUASY	Permits the export of the key with an unauthenticated asymmetric key. This is the default.
Authenticated asymmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEXAASY	Prohibits the export of the key with an authenticated asymmetric key.
XPRTAASY	Permits the export of the key with an authenticated asymmetric key. This is the default.
RAW-key export control (one, optional). Key-management field 1, high-order byte.	
NOEX-RAW	Prohibits the export of the key in RAW format. This is the default.
XPRT-RAW	Permits the export of the key in RAW format.
DES-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-DES	Prohibits the export of the key using DES key.
XPRT-DES	Permits the export of the key using DES key. This is the default.
AES-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-AES	Prohibits the export of the key using AES key.
XPRT-AES	Permits the export of the key using AES key. This is the default.
RSA-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-RSA	Prohibits the export of the key using RSA key.
XPRT-RSA	Permits the export of the key using RSA key. This is the default.

Figure 4 on page 27 shows all the valid keyword combinations and their defaults for AES key type EXPORTER. For a description of these keywords, see Table 10 on page 28.

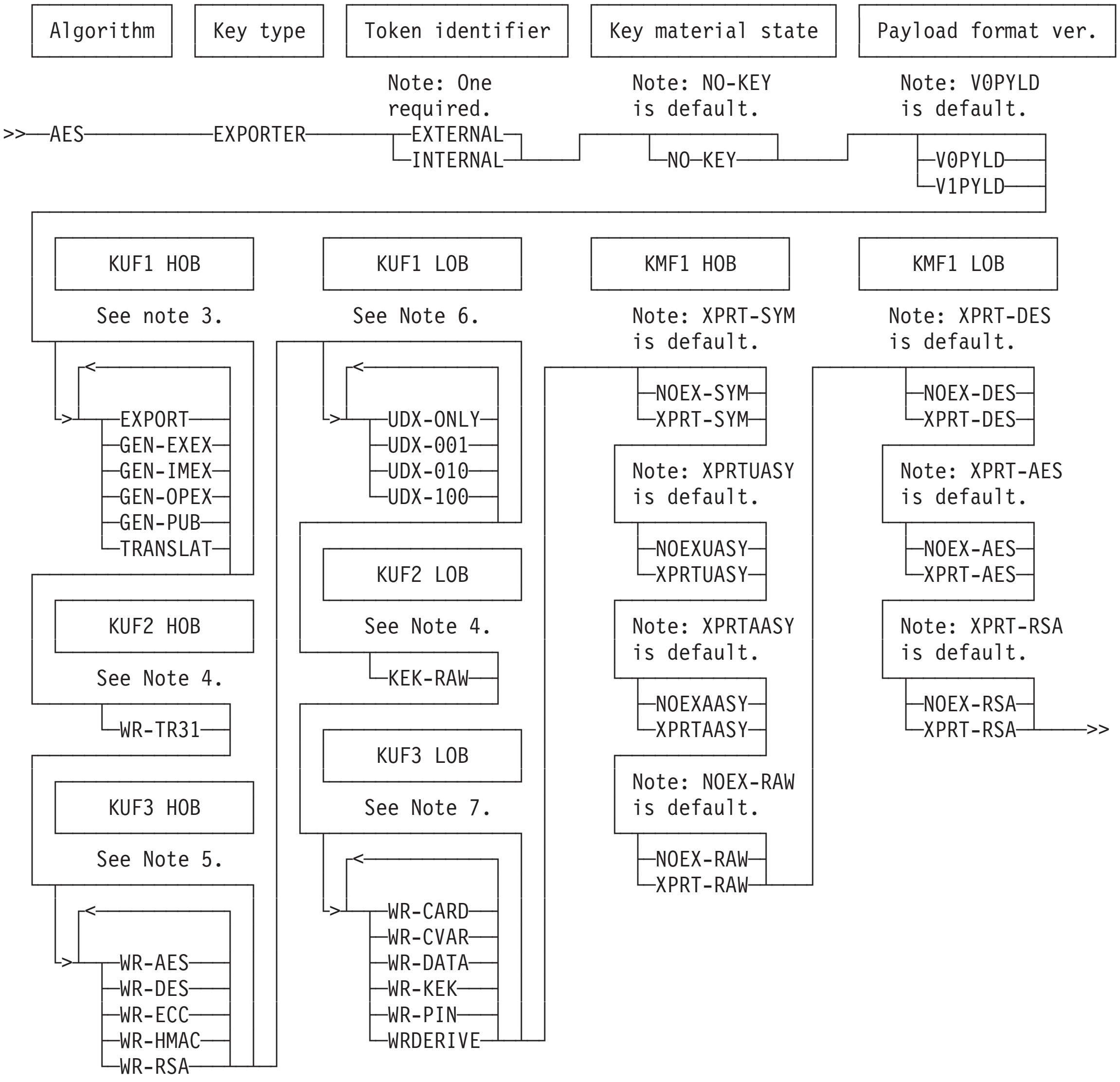


Figure 4. Key Token Build2 keyword combinations for AES EXPORTER keys

Notes:

1. Keyword V0PYLD is the default for compatibility reasons. V1PYLD is recommended because it provides improved security.
2. Each key-usage field (KUF) and key-management field (KMF) of a version X'05' variable-length symmetric key-token consists of a high-order byte (HOB) and a low-order byte (LOB).
3. All keywords in the group are defaults unless one or more keywords in the group are specified.
4. There is no default. This keyword is defined for future use and its meaning is currently undefined. To avoid this restriction in the future when the meaning is defined, specify this keyword.
5. Keywords WR-AES, WR-DES, and WR-HMAC are defaults unless one or more keywords in the group are specified.
6. Choose any number of keywords in this group. No keywords in the group are defaults.
7. Keywords WR-CARD, WR-DATA, WR-KEK, WR-PIN, and WRDERIVE are defaults unless one or more keywords in this group are specified.

8. NOEX-RAW and XPRT-RAW are defined for future use and their meanings are currently undefined. To avoid this export restriction in the future when the meaning is defined, specify XPRT-RAW.

Table 10. Rule array keywords for AES EXPORTER keys

Keyword	Meaning
Key-token header section	
Token identifier (one required).	
EXTERNAL	Build a key token that is not to be used locally.
INTERNAL	Build a key token that is to be used locally.
Wrapping-information section	
Key status (one, optional).	
NO-KEY	Build a key token that does not contain a key value. This is the default.
Payload format version (one, optional). Identifies format of the payload.	
V0PYLD	Build a key token with a version 0 payload format. This format has a variable length and the key length can be inferred from the size of the payload. This is the default.
V1PYLD	Build the key token with a version 1 payload format. This format has a fixed length and the key length cannot be inferred by the size of the payload. An obscured key length is considered more secure.
Associated data section	
Algorithm type (one required).	
AES	Key can be used for AES algorithm.
Key type (one required).	
EXPORTER	Key can be used for wrap a key to be exported.
Exporter control (any combination, optional). Key-usage field 1, high-order byte. Note: All keywords in the list below are defaults unless one or more keywords in the list are specified.	
EXPORT	Specifies that this key can be used for export.
TRANSLAT	Specifies that this key can be used for translate.
GEN-OPEX	Specifies that this key can be used for generate OPEX.
GEN-IMEX	Specifies that this key can be used for generate IMEX.
GEN-EXEX	Specifies that this key can be used for generate EXEX.
GEN-PUB	Specifies that this key can be used for generate PUB.
User-defined extension (UDX) control (one or more, optional). Key-usage field 1, low-order byte. No keywords in the group are defaults.	
UDX-ONLY	Key can only be used in UDXs.
UDX-001	Specifies that the rightmost user-defined UDX bit is set.
UDX-010	Specifies that the middle user-defined UDX bit is set.
UDX-100	Specifies that the leftmost user-defined UDX bit is set.
TR-31 wrap control (one, optional). Key-usage field 2, high-order byte.	
WR-TR31	Specifies that this key-encrypting key can wrap or unwrap a TR-31 key block. Defined for future use.
Raw key wrap control (one, optional). Key-usage field 2, low-order byte.	
KEK-RAW	Specifies that this key-encrypting key can export a RAW key. Defined for future use.
Key-usage wrap algorithm control (any combination, optional). Key-usage field 3, high-order byte. Note: Keywords WR-DES, WR-AES, and WR-HMAC are defaults unless one or more keywords are specified.	

Table 10. Rule array keywords for AES EXPORTER keys (continued)

Keyword	Meaning
WR-DES	Specifies that this key can be used to wrap DES keys.
WR-AES	Specifies that this key can be used to wrap AES keys.
WR-HMAC	Specifies that this key can be used to wrap HMAC keys.
WR-RSA	Specifies that this key can be used to wrap RSA keys.
WR-ECC	Specifies that this key can be used to wrap ECC keys.
Key-usage wrap class control (any combination, optional). Key-usage field 4, high-order byte. Note: Keywords WR-DATA, WR-KEK, WR-PIN, WRDERIVE, and WR-CARD in the list below are defaults unless one or more keywords in the list are specified.	
WR-DATA	Specifies that this key can be used to wrap DATA class keys.
WR-KEK	Specifies that this key can be used to wrap KEK class keys.
WR-PIN	Specifies that this key can be used to wrap PIN class keys.
WRDERIVE	Specifies that this key can be used to wrap DERIVATION class keys.
WR-CARD	Specifies that this key can be used to wrap CARD class keys.
WR-CVAR	Specifies that this key can be used to wrap CVAR class keys.
Symmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEX-SYM	Prohibits the export of the key with a symmetric key.
XPRT-SYM	Permits the export of the key with a symmetric key. This is the default.
Unauthenticated asymmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEXUASY	Prohibits the export of the key with an unauthenticated asymmetric key.
XPRTUASY	Permits the export of the key with an unauthenticated asymmetric key. This is the default.
Authenticated asymmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEXAASY	Prohibits the export of the key with an authenticated asymmetric key.
XPRTAASY	Permits the export of the key with an authenticated asymmetric key. This is the default.
RAW-key export control (one, optional). Key-management field 1, high-order byte.	
NOEX-RAW	Prohibits the export of the key in RAW format. This is the default.
XPRT-RAW	Permits the export of the key in RAW format.
DES-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-DES	Prohibits the export of the key using DES key.
XPRT-DES	Permits the export of the key using DES key. This is the default.
AES-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-AES	Prohibits the export of the key using AES key.
XPRT-AES	Permits the export of the key using AES key. This is the default.
RSA-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-RSA	Prohibits the export of the key using RSA key.
XPRT-RSA	Permits the export of the key using RSA key. This is the default.

Figure 5 on page 30 shows all the valid keyword combinations and their defaults for AES key type IMPORTER. For a description of these keywords, see Table 11 on page 31.

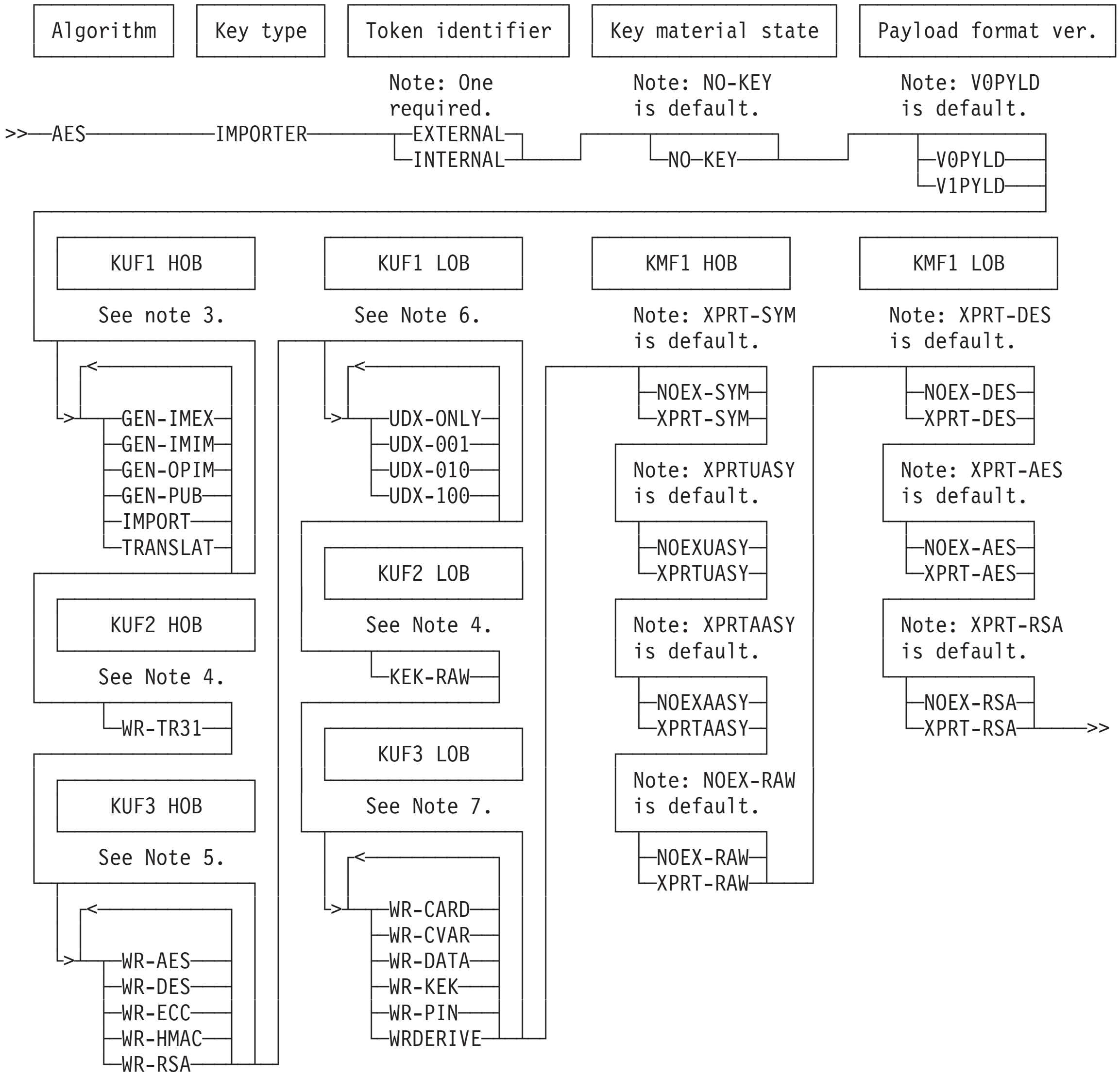


Figure 5. Key Token Build2 keyword combinations for AES IMPORTER keys

Notes:

1. Keyword V0PYLD is the default for compatibility reasons. V1PYLD is recommended because it provides improved security.
2. Each key-usage field (KUF) and key-management field (KMF) of a version X'05' variable-length symmetric key-token consists of a high-order byte (HOB) and a low-order byte (LOB).
3. All keywords in the group are defaults unless one or more keywords in the group are specified.
4. This keyword is defined for future use and its meaning is currently undefined. To avoid this restriction in the future when the meaning is defined, specify this keyword.
5. Keywords WR-AES, WR-DES, and WR-HMAC are defaults unless one or more keywords in the group are specified.
6. Choose any number of keywords in this group. No keywords in the group are defaults.
7. Keywords WR-CARD, WR-DATA, WR-KEK, WR-PIN, and WRDERIVE in the group are defaults unless one or more keywords in the group are specified.

8. NOEX-RAW and XPRT-RAW are defined for future use and their meanings are currently undefined. To avoid this export restriction in the future when the meaning is defined, specify XPRT-RAW.

Table 11. Rule array keywords for AES IMPORTER keys

Keyword	**Meaning**
Key-token header section	
Token identifier (one required).	
EXTERNAL	Build a key token that is not to be used locally.
INTERNAL	Build a key token that is to be used locally.
Wrapping-information section	
Key status (one, optional).	
NO-KEY	Build a key token that does not contain a key value. This is the default.
Payload format version (one, optional). Identifies format of the payload.	
V0PYLD	Build a key token with a version 0 payload format. This format has a variable length and the key length can be inferred from the size of the payload. This is the default.
V1PYLD	Build the key token with a version 1 payload format. This format has a fixed length and the key length cannot be inferred by the size of the payload. An obscured key length is considered more secure.
Associated data section	
Algorithm type (one required).	
AES	Key can be used for AES algorithm.
Key type (one required).	
IMPORTER	Key can be used for wrap a key to be imported.
Importer control (any combination, optional). Key-usage field 1, high-order byte. Note: All keywords in the list below are defaults unless one or more keywords in the list are specified.	
IMPORT	Specifies that this key can be used for import.
TRANSLAT	Specifies that this key can be used for translate.
GEN-OPIM	Specifies that this key can be used for generate OPIM.
GEN-IMEX	Specifies that this key can be used for generate IMEX.
GEN-IMIM	Specifies that this key can be used for generate IMIM.
GEN-PUB	Specifies that this key can be used for generate PUB.
User-defined extension (UDX) control (one or more, optional). Key-usage field 1, low-order byte. No keywords in the group are defaults.	
UDX-ONLY	Key can only be used in UDXs.
UDX-001	Specifies that the rightmost user-defined UDX bit is set.
UDX-010	Specifies that the middle user-defined UDX bit is set.
UDX-100	Specifies that the leftmost user-defined UDX bit is set.
TR-31 wrap control (one, optional). Key-usage field 2, high-order byte.	
WR-TR31	Specifies that this key-encrypting key can wrap or unwrap a TR-31 key block. Defined for future use.
Raw key wrap control (one, optional). Key-usage field 2, low-order byte.	
KEK-RAW	Specifies that this key-encrypting key can export a RAW key. Defined for future use.
Key-usage wrap algorithm control (any combination, optional). Key-usage field 3, high-order byte. Note: Keywords WR-DES, WR-AES, and WR-HMAC are defaults unless one or more keywords are specified.	

Table 11. Rule array keywords for AES IMPORTER keys (continued)

Keyword	Meaning
WR-DES	Specifies that this key can be used to wrap DES keys.
WR-AES	Specifies that this key can be used to wrap AES keys.
WR-HMAC	Specifies that this key can be used to wrap HMAC keys.
WR-RSA	Specifies that this key can be used to wrap RSA keys.
WR-ECC	Specifies that this key can be used to wrap ECC keys.
Key-usage wrap class control (any combination, optional). Key-usage field 4, high-order byte. Note: Keywords WR-DATA, WR-KEK, WR-PIN, WRDERIVE, and WR-CARD in the list below are defaults unless one or more keywords in the list are specified.	
WR-DATA	Specifies that this key can be used to wrap DATA class keys.
WR-KEK	Specifies that this key can be used to wrap KEK class keys.
WR-PIN	Specifies that this key can be used to wrap PIN class keys.
WRDERIVE	Specifies that this key can be used to wrap DERIVATION class keys.
WR-CARD	Specifies that this key can be used to wrap CARD class keys.
WR-CVAR	Specifies that this key can be used to wrap CVAR class keys.
Symmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEX-SYM	Prohibits the export of the key with a symmetric key.
XPRT-SYM	Permits the export of the key with a symmetric key. This is the default.
Unauthenticated asymmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEXUASY	Prohibits the export of the key with an unauthenticated asymmetric key.
XPRTUASY	Permits the export of the key with an unauthenticated asymmetric key. This is the default.
Authenticated asymmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEXAASY	Prohibits the export of the key with an authenticated asymmetric key.
XPRTAASY	Permits the export of the key with an authenticated asymmetric key. This is the default.
RAW-key export control (one, optional). Key-management field 1, high-order byte.	
NOEX-RAW	Prohibits the export of the key in RAW format. This is the default.
XPRT-RAW	Permits the export of the key in RAW format.
DES-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-DES	Prohibits the export of the key using DES key.
XPRT-DES	Permits the export of the key using DES key. This is the default.
AES-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-AES	Prohibits the export of the key using AES key.
XPRT-AES	Permits the export of the key using AES key. This is the default.
RSA-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-RSA	Prohibits the export of the key using RSA key.
XPRT-RSA	Permits the export of the key using RSA key. This is the default.

Figure 6 on page 33 shows all the valid keyword combinations and their defaults for AES key type DKYGENKY. For a description of these keywords, see Table 12 on page 34.

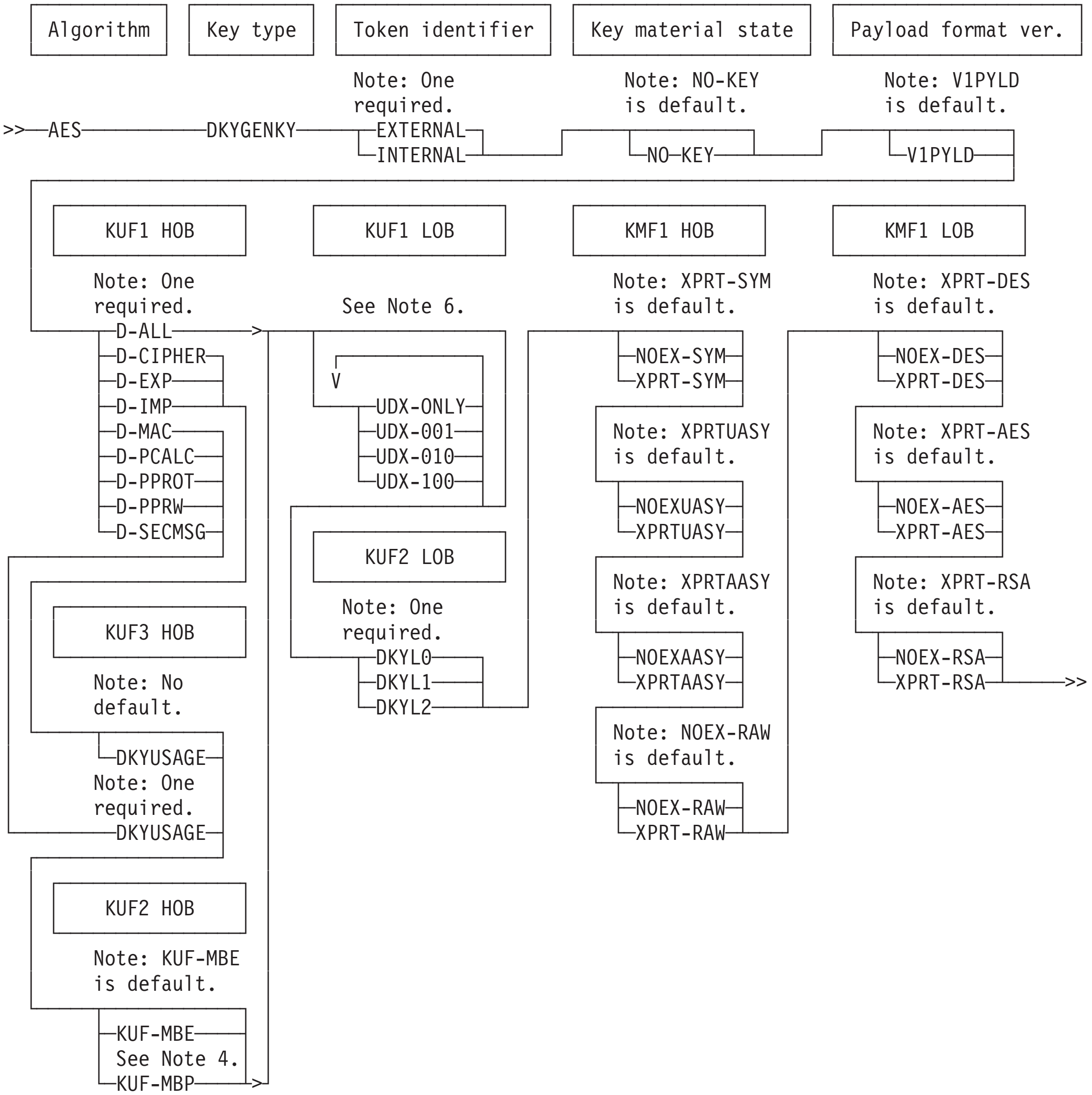


Figure 6. Key Token Build2 keyword combinations for AES DKYGENKY keys

Notes:

1. Keyword V1PYLD is the default. V1PYLD is the only payload format version allowed for this key type.
2. Each key-usage field (KUF) and key-management field (KMF) of a version X'05' variable-length symmetric key-token consists of two bytes: a high-order byte (HOB) and a low-order byte (LOB).
3. NOEX-RAW and XPRT-RAW are defined for future use and their meanings are currently undefined. To avoid this export restriction in the future when the meaning is defined, specify XPRT-RAW.
4. DKYUSAGE specifies that the *service_data* variable contains the keywords that define the key usage attributes related to the type of key to diversify. Based on the *service_data* keywords, CSNBKTB2 appends the key usage attributes of the type of key to diversify to the key usage fields of the DKYGENKY key. The related key usage fields control which key usage attributes are permissible for the final generated diversified key. DKYUSAGE is not valid with D-ALL because the type of key to diversify is unspecified. DKYUSAGE is optional

with D-CIPHER, D-EXP, and D-IMP because key types CIPHER, EXPORTER, and IMPORTER have default key usage attributes. For these key types, if DKYUSAGE is not specified, CSNBKTB2 assigns default key usage attributes to the related KUF fields. DKYUSAGE is required for the remaining values of type of key to diversify because those key types do not have default key usage attributes.

5. KUF-MBP is not valid if DKADMIN1, DKADMIN2, DKPINOP, or DKPINOPP is specified in the *service_data* parameter (that is, the type of key to diversify is DK enabled).
6. Choose any number of keywords in this group. No keywords in this group are defaults.

Table 12. Rule array keywords for AES DKYGENKY keys

Keyword	Meaning
Key-token header section	
Token identifier (one required).	
EXTERNAL	Build a key token that is not to be used locally.
INTERNAL	Build a key token that is to be used locally.
Wrapping-information section	
Key status (one, optional).	
NO-KEY	Build a key token that does not contain a key value. This is the default.
Payload format version (one, optional). Identifies format of the payload.	
V1PYLD	Build the key token with a version 1 payload format. This format has a fixed length and the key length cannot be inferred by the size of the payload. An obscured key length is considered more secure.
Associated data section	
Algorithm type (one required).	
AES	Key can be used for AES algorithm.
Key type (one required).	
DKYGENKY	Key can be used for generating a diversified key.
Type of key to diversify (one, required). Key-usage field 1, high-order byte. DKYUSAGE is required for D-MAC, D-PCALC, D-PPROT, D-PPRW, and D-SECMSG.	
D-ALL	Specifies that this can derive any AES key type listed in this section.
D-CIPHER	Specifies that this key can derive an AES CIPHER key.
D-EXP	Specifies that this key can derive an AES EXPORTER key.
D-IMP	Specifies that this key can derive an AES IMPORTER key.
D-MAC	Specifies that this key can derive an AES MAC key.
D-PCALC	Specifies that this key can derive an AES PINCALC key.
D-PPROT	Specifies that this key can derive an AES PINPROT key.
D-PPRW	Specifies that this key can derive an AES PINPRW key.
D-SECMSG	Specifies that this key can derive an AES SECMSG key.
User-defined extension (UDX) control (one or more, optional). Key-usage field 1, low-order byte. No keywords in the group are defaults.	
UDX-ONLY	Key can only be used in UDXs.
UDX-001	Specifies that the rightmost user-defined UDX bit is set.
UDX-010	Specifies that the middle user-defined UDX bit is set.

Table 12. Rule array keywords for AES DKYGENKY keys (continued)

Keyword	Meaning
UDX-100	Specifies that the leftmost user-defined UDX bit is set.
Key-usage field level of control (one, optional). Key-usage field 2, high-order byte. Note: Not valid when D-ALL key derivation control is specified.	
KUF-MBE	Specifies that the key usage fields of the key to be generated must be equal to the related generated key usage fields of the DKYGENKY generating key. This is the default.
KUF-MBP	Specifies that the key usage fields of the key to be generated must be permitted based on the related generated key usage fields of the DKYGENKY generating key. The key to be diversified is not permitted to have a higher level of usage than the related key usage fields permit. The key to be diversified is only permitted to have key usage that is less than or equal to the related key usage fields. The UDX-ONLY bit of the related key usage fields must always be equal in both the generating key and the generated key. Note: This value is not valid if the key is to be used to derive keys for the DK PIN methods.
Key-derivation sequence level (one, required). Key-usage field 2, low-order byte.	
DKYL0	Specifies that this key-generating key can be used to derive the key specified by the key derivation and derived key usage controls.
DKYL1	Use this diversifying key to generate a level 0 diversified key.
DKYL2	Use this diversifying key to generate a level 1 diversified key.
Related generated key usage fields (not allowed for D-ALL, one, optional, for D-CIPHER, D-EXP, and D-IMP, otherwise one required). Key-usage field 3, high-order byte.	
DKYUSAGE	Specifies that the *service_data* parameter identifies key usage information for a DKYGENKY. This information pertains to the allowable key usage of the key to be derived.
Symmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEX-SYM	Prohibits the export of the key with a symmetric key.
XPRT-SYM	Permits the export of the key with a symmetric key. This is the default.
Unauthenticated asymmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEXUASY	Prohibits the export of the key with an unauthenticated asymmetric key.
XPRTUASY	Permits the export of the key with an unauthenticated asymmetric key. This is the default.
Authenticated asymmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEXAASY	Prohibits the export of the key with an authenticated asymmetric key.
XPRTAASY	Permits the export of the key with an authenticated asymmetric key. This is the default.
RAW-key export control (one, optional). Key-management field 1, high-order byte.	
NOEX-RAW	Prohibits the export of the key in RAW format. This is the default.
XPRT-RAW	Permits the export of the key in RAW format.
DES-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-DES	Prohibits the export of the key using DES key.
XPRT-DES	Permits the export of the key using DES key. This is the default.
AES-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-AES	Prohibits the export of the key using AES key.
XPRT-AES	Permits the export of the key using AES key. This is the default.
RSA-key export control (one, optional). Key-management field 1, low-order byte.	

Table 12. Rule array keywords for AES DKYGENKY keys (continued)

Keyword	Meaning
NOEX-RSA	Prohibits the export of the key using RSA key.
XPRT-RSA	Permits the export of the key using RSA key. This is the default.

Table 13. Meaning of service_data parameter when DKYUSAGE specified

Type of key to diversify	DKYUSAGE keyword	Description of verb_data variable when DKYUSAGE specified
D-ALL	Not allowed	Not applicable.
D-CIPHER	Optional	If keyword DKYUSAGE is specified, the *service_data* parameter must contain key usage fields keywords related to an AES CIPHER key. If not specified, the related key usage fields will be that of a default AES CIPHER key.
D-EXP	Optional	If keyword DKYUSAGE is specified, the *service_data* parameter must contain key usage fields keywords related to an AES EXPORTER key. If not specified, the related key usage fields will be that of a default AES EXPORTER key.
D-IMP	Optional	If keyword DKYUSAGE is specified, the *service_data* parameter must contain key usage fields keywords related to an AES IMPORTER key. If not specified, the related key usage fields will be that of a default AES IMPORTER key.
D-MAC	Required	The *service_data* parameter must contain key usage fields keywords related to an AES MAC key.
D-PCALC	Required	The *service_data* parameter must contain key usage fields keywords related to an AES PINCALC key.
D-PPROT	Required	The *service_data* parameter must contain key usage fields keywords related to an AES PINPROT key.
D-PPRW	Required	The *service_data* parameter must contain key usage fields keywords related to an AES PINPRW key.
D-SECMSG	Required	The *service_data* parameter must contain key usage fields keywords related to an AES SECMSG key.

Building a DKYGENKY key

The way that the DKYGENKY tokens are built is different from the way they were previously built. The token layout itself has been updated. The DKYGENKY key is used to derive other key types.

In order to control the key usage of the key to be derived, key usage field information for the derived key is included in the DKYGENKY token. Consider these scenarios based on the type of key to derive:

- DKYGENKY has a type of key to derive of D-ALL.

  This type of key is allowed to derive any of the allowed key types. No key usage field information is included in this key. Usage is determined by the skeleton token identified by the generated_key_identifier parameter of the CSNBDKG2 callable service. A special access control point must be enabled in the active role to use this option.
- DKYGENKY has a type of key to be derived that has default key usage (D-CIPHER, D-EXP, D-IMP).

  Several key types have default key usage defined, while other key types do not. For those key types which have default key usage defined (D-CIPHER, D-EXP, and D-IMP), the only requirement is to specify the type of key to derive. The default key usage fields is included in the DKYGENKY key, beginning with key usage field 3.
- DKYGENKY has a type of key to be derived that requires non-default key usage.

  If non-default key usage of a key to be derived is required or desired, specify rule array keyword DKYUSAGE. With this keyword, the verb_data parameter is used to identify all of the key-usage field keywords for the key to be diversified. Do not specify any token identifier, type of algorithm, key type, or key management field keywords. Set the verb_data_length value to the number of bytes in the verb_data variable. This length must be a multiple of 8.

  When rule array keyword DKYUSAGE is specified, choose between whether the key usage field attributes in the DKYGENKY starting at key-usage field 3 have the strictest control (KUF-MBE or 'must be equal', which is the default) or allow flexibility in the key usage attributes of the key to be generated (KUF-MBP or 'must be permitted').

  Choosing KUF-MBE ('key usage fields must be equal') provides a one-to-one mapping of usage fields between the generating key and the generated key. The key usage fields related to the key to be diversified in the DKYGENKY key must match exactly with the key usage fields of any skeleton key provided as input to the CSNBDKG2 callable service.

  Choosing KUF-MBP ('key usage fields must be permitted') provides that the key to be diversified is allowed to have any key usage attribute that is enabled in the DKYGENKY. For example, if a DKYGENKY with D-EXP usage has default EXPORTER key usage fields, KUF-MBP allows the diversified EXPORTER key to have only the EXPORT bit on in key-usage field 1. This is permitted because the diversified key actually is more restrictive than the usage allowed by the DKYGENKY key. Conversely, if a DKYGENKY with D-EXP usage has only the EXPORT bit on in key-usage field 3 (which maps to key usage field 1 of the diversified EXPORTER key), it would not be permitted for the skeleton key used as input to the CSNBDKG2 callable service to have the XLATE bit on in key-usage field 1.

  Notes:

  – For rule array keyword KUF-MBP, one exception exists where the value of the UDX-ONLY bit in key usage field 3 of a DKYGENKY key must always match the value of the UDX-ONLY bit in key usage field 1 of the diversified key.

  – Under access control point control, there is one case where a many-to-one mapping is permitted and verb data is not used. This case is when you specify D-ALL which says any allowable key type can be derived.

Figure 7 on page 38 shows all the valid keyword combinations and their defaults for AES key type PINCALC. For a description of these keywords, see Table 14 on page 38 page 38.

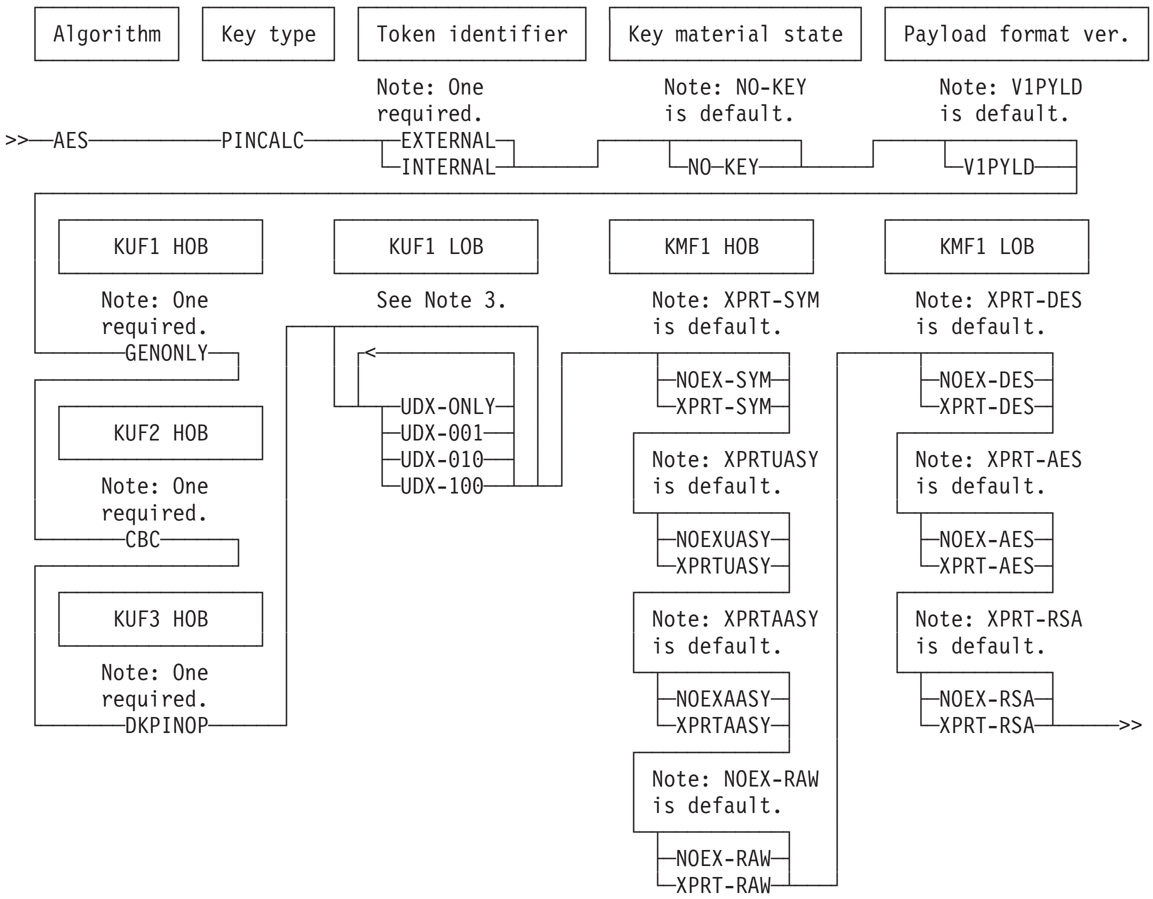


Figure 7. Key Token Build2 keyword combinations for AES PINCALC keys

Notes:

1. Each key-usage field (KUF) and key-management field (KMF) of a version X'05' variable-length symmetric key-token consists of two bytes: a high-order byte (HOB) and a low-order byte (LOB).
2. NOEX-RAW and XPRT-RAW are defined for future use and their meanings are currently undefined. To avoid this export restriction in the future when the meaning is defined, specify XPRT-RAW.
3. Choose any number of keywords in this group. No keywords in the group are defaults.

Table 14. Rule array keywords for AES PINCALC keys

Keyword	Meaning
Key-token header section	
Token identifier (one required).	
EXTERNAL	Build a key token that is not to be used locally.
INTERNAL	Build a key token that is to be used locally.
Wrapping-information section	
Key status (one, optional).	
NO-KEY	Build a key token that does not contain a key value. This is the default.

Table 14. Rule array keywords for AES PINCALC keys (continued)

Keyword	Meaning
Payload format version (one, optional). Identifies format of the payload.	
V1PYLD	Build the key token with a version 1 payload format. This format has a fixed length and the key length cannot be inferred by the size of the payload. An obscured key length is considered more secure.
Associated data section	
Algorithm type (one required).	
AES	Key can be used for AES algorithm.
Key type (one required).	
PINCALC	Key can be used for generating PINs.
Generate control (one required). Key-usage field 1, high-order byte.	
GENONLY	Specifies that this key can only be used to generate a PIN. It cannot be used to verify a PIN.
User-defined extension (UDX) control (one or more, optional). Key-usage field 1, low-order byte. No keywords in the group are defaults.	
UDX-ONLY	Key can only be used in UDXs.
UDX-001	Specifies that the rightmost user-defined UDX bit is set.
UDX-010	Specifies that the middle user-defined UDX bit is set.
UDX-100	Specifies that the leftmost user-defined UDX bit is set.
Encryption mode (one, required). Key-usage field 2, high-order byte.	
CBC	Specifies that this key can be used for cipher block chaining.
Common control (one, optional). Key-usage field 3, high-order byte. Use of a common control keyword causes key-usage field 3, low-order byte (field format identifier at token offset 050) to be set to X'01' (DK enabled).	
DKPINOP	Specifies that this key may be used as a general-purpose key. It may not be used as a special-purpose key.
Symmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEX-SYM	Prohibits the export of the key with a symmetric key.
XPRT-SYM	Permits the export of the key with a symmetric key. This is the default.
Unauthenticated asymmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEXUASY	Prohibits the export of the key with an unauthenticated asymmetric key.
XPRTUASY	Permits the export of the key with an unauthenticated asymmetric key. This is the default.
Authenticated asymmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEXAASY	Prohibits the export of the key with an authenticated asymmetric key.
XPRTAASY	Permits the export of the key with an authenticated asymmetric key. This is the default.
RAW-key export control (one, optional). Key-management field 1, high-order byte.	
NOEX-RAW	Prohibits the export of the key in RAW format. This is the default.
XPRT-RAW	Permits the export of the key in RAW format.
DES-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-DES	Prohibits the export of the key using DES key.
XPRT-DES	Permits the export of the key using DES key. This is the default.
AES-key export control (one, optional). Key-management field 1, low-order byte.	

Table 14. Rule array keywords for AES PINCALC keys (continued)

Keyword	Meaning
NOEX-AES	Prohibits the export of the key using AES key.
XPRT-AES	Permits the export of the key using AES key. This is the default.
RSA-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-RSA	Prohibits the export of the key using RSA key.
XPRT-RSA	Permits the export of the key using RSA key. This is the default.

Figure 8 shows all the valid keyword combinations and their defaults for AES key type PINPROT. For a description of these keywords, see Table 15 on page 41.

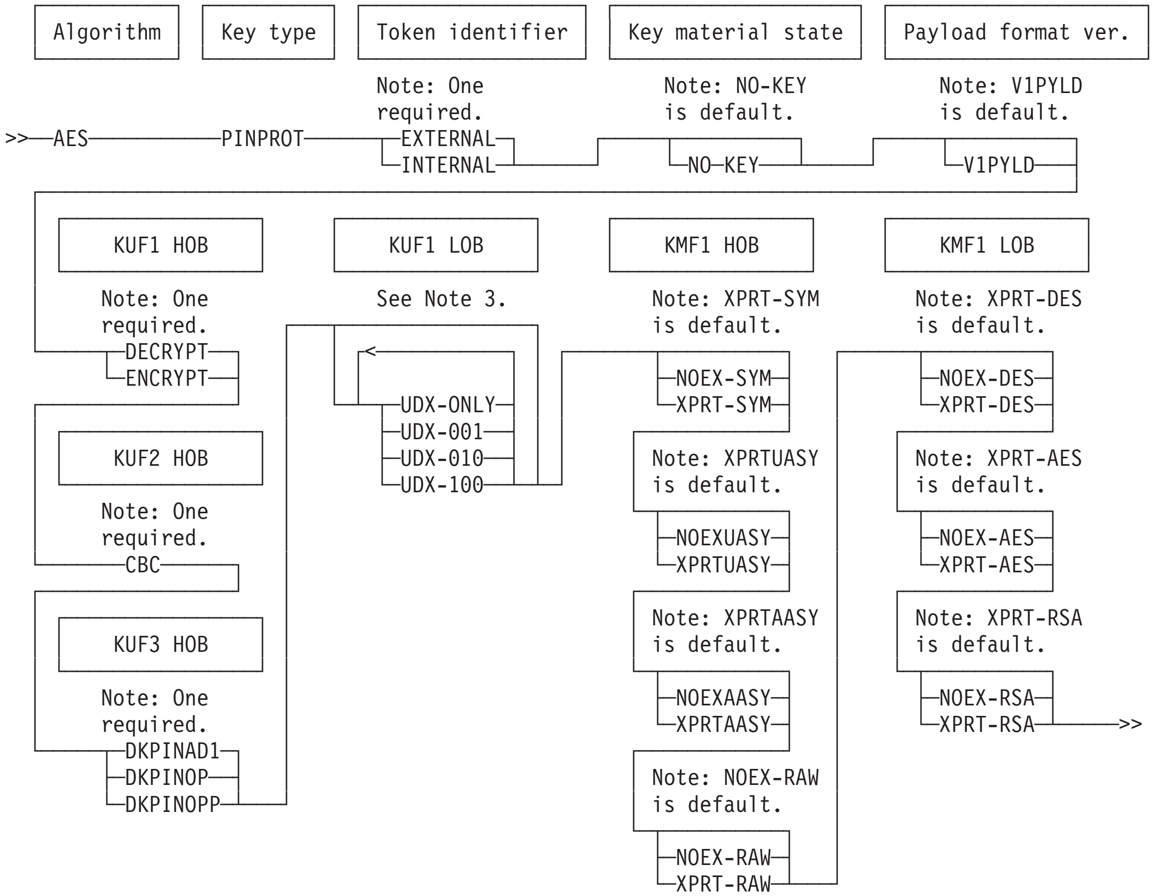


Figure 8. Key Token Build2 keyword combinations for AES PINPROT keys

Notes:

1. Each key-usage field (KUF) and key-management field (KMF) of a version X'05' variable-length symmetric key-token consists of two bytes: a high-order byte (HOB) and a low-order byte (LOB).
2. NOEX-RAW and XPRT-RAW are defined for future use and their meanings are currently undefined. To avoid this export restriction in the future when the meaning is defined, specify XPRT-RAW.
3. Choose any number of keywords in this group. No keywords in the group are defaults.

Table 15. Rule array keywords for AES PINPROT keys

Keyword	Meaning
Key-token header section	
Token identifier (one required).	
EXTERNAL	Build a key token that is not to be used locally.
INTERNAL	Build a key token that is to be used locally.
Wrapping-information section	
Key status (one, optional).	
NO-KEY	Build a key token that does not contain a key value. This is the default.
Payload format version (one, optional). Identifies format of the payload.	
V1PYLD	Build the key token with a version 1 payload format. This format has a fixed length and the key length cannot be inferred by the size of the payload. An obscured key length is considered more secure.
Associated data section	
Algorithm type (one required).	
AES	Key can be used for AES algorithm.
Key type (one required).	
PINPROT	Key can be used for encrypting and decrypting PIN blocks.
Encryption operation (one, required). Key-usage field 1, high-order byte.	
DECRYPT	Specifies that this key can be used to decipher data. The key can not be used to encipher data.
ENCRYPT	Specifies that this key can be used to encipher data. The key can not be used to decipher data.
User-defined extension (UDX) control (one or more, optional). Key-usage field 1, low-order byte. No keywords in the group are defaults.	
UDX-ONLY	Key can only be used in UDXs.
UDX-001	Specifies that the rightmost user-defined UDX bit is set.
UDX-010	Specifies that the middle user-defined UDX bit is set.
UDX-100	Specifies that the leftmost user-defined UDX bit is set.
Encryption mode (one, required). Key-usage field 2, high-order byte.	
CBC	Specifies that this key can be used for cipher block chaining.
Common control (one, optional). Key-usage field 3, high-order byte. Use of a common control keyword causes key-usage field 3, low-order byte (field format identifier at token offset 050) to be set to X'01' (DK enabled).	
DKPINOP	Specifies that this key may be used as a general-purpose key. It may not be used as a special-purpose key.
DKPINOPP	Specifies that this key is to be used to encrypt a PBF-1 format pin block for the specific purpose of creating a PIN mailer.
DKPINAD1	Specifies that this key may be used to create or verify a pin block to allow changing the account number associate with a PIN.
Symmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEX-SYM	Prohibits the export of the key with a symmetric key.
XPRT-SYM	Permits the export of the key with a symmetric key. This is the default.
Unauthenticated asymmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEXUASY	Prohibits the export of the key with an unauthenticated asymmetric key.

Table 15. Rule array keywords for AES PINPROT keys (continued)

Keyword	Meaning
XPRTUASY	Permits the export of the key with an unauthenticated asymmetric key. This is the default.
Authenticated asymmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEXAASY	Prohibits the export of the key with an authenticated asymmetric key.
XPRTAASY	Permits the export of the key with an authenticated asymmetric key. This is the default.
RAW-key export control (one, optional). Key-management field 1, high-order byte.	
NOEX-RAW	Prohibits the export of the key in RAW format. This is the default.
XPRT-RAW	Permits the export of the key in RAW format.
DES-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-DES	Prohibits the export of the key using DES key.
XPRT-DES	Permits the export of the key using DES key. This is the default.
AES-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-AES	Prohibits the export of the key using AES key.
XPRT-AES	Permits the export of the key using AES key. This is the default.
RSA-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-RSA	Prohibits the export of the key using RSA key.
XPRT-RSA	Permits the export of the key using RSA key. This is the default.

Figure 9 on page 43 shows all the valid keyword combinations and their defaults for AES key type PINPRW. For a description of these keywords, see Table 16 on page 43.

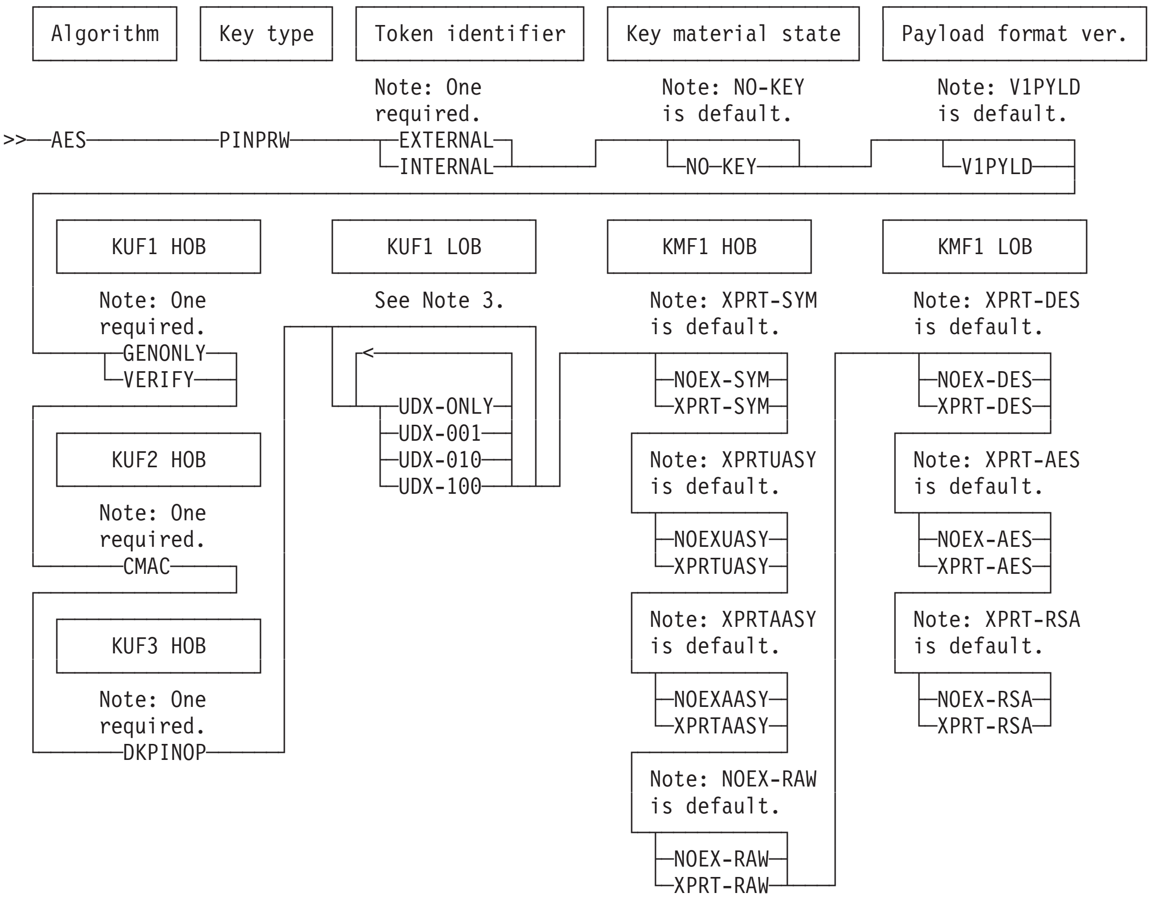


Figure 9. Key Token Build2 keyword combinations for AES PINPRW keys

Notes:

1. Each key-usage field (KUF) and key-management field (KMF) of a version X'05' variable-length symmetric key-token consists of two bytes: a high-order byte (HOB) and a low-order byte (LOB).
2. NOEX-RAW and XPRT-RAW are defined for future use and their meanings are currently undefined. To avoid this export restriction in the future when the meaning is defined, specify XPRT-RAW.
3. Choose any number of keywords in this group. No keywords in the group are defaults.

Table 16. Rule array keywords for AES PINPRW keys

Keyword	Meaning
Key-token header section	
Token identifier (one required).	
EXTERNAL	Build a key token that is not to be used locally.
INTERNAL	Build a key token that is to be used locally.
Wrapping-information section	
Key status (one, optional).	
NO-KEY	Build a key token that does not contain a key value. This is the default.
Payload format version (one, optional). Identifies format of the payload.	

Table 16. Rule array keywords for AES PINPRW keys (continued)

Keyword	Meaning
V1PYLD	Build the key token with a version 1 payload format. This format has a fixed length and the key length cannot be inferred by the size of the payload. An obscured key length is considered more secure.
Associated data section	
Algorithm type (one required).	
AES	Key can be used for AES algorithm.
Key type (one required).	
PINPRW	Key can be used for generating and verifying PIN reference words.
Generate control (one required). Key-usage field 1, high-order byte.	
GENONLY	Specifies that this key can only be used to generate a PRW. It can not be used to verify a PRW.
VERIFY	Specifies that this key cannot be used to generate a PRW. It can only be used to verify a PRW.
User-defined extension (UDX) control (one or more, optional). Key-usage field 1, low-order byte. No keywords in the group are defaults.	
UDX-ONLY	Key can only be used in UDXs.
UDX-001	Specifies that the rightmost user-defined UDX bit is set.
UDX-010	Specifies that the middle user-defined UDX bit is set.
UDX-100	Specifies that the leftmost user-defined UDX bit is set.
Mode control (one, required). Key-usage field 2, high-order byte.	
CMAC	MAC calculation mode is block cipher-based MAC algorithm.
Common control (one, optional). Key-usage field 3, high-order byte. Use of a common control keyword causes key-usage field 3, low-order byte (field format identifier at token offset 050) to be set to X'01' (DK enabled).	
DKPINOP	Specifies that this key may be used as a general-purpose key. It may not be used as a special-purpose key.
Symmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEX-SYM	Prohibits the export of the key with a symmetric key.
XPRT-SYM	Permits the export of the key with a symmetric key. This is the default.
Unauthenticated asymmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEXUASY	Prohibits the export of the key with an unauthenticated asymmetric key.
XPRTUASY	Permits the export of the key with an unauthenticated asymmetric key. This is the default.
Authenticated asymmetric-key export control (one, optional). Key-management field 1, high-order byte.	
NOEXAASY	Prohibits the export of the key with an authenticated asymmetric key.
XPRTAASY	Permits the export of the key with an authenticated asymmetric key. This is the default.
RAW-key export control (one, optional). Key-management field 1, high-order byte.	
NOEX-RAW	Prohibits the export of the key in RAW format. This is the default.
XPRT-RAW	Permits the export of the key in RAW format.
DES-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-DES	Prohibits the export of the key using DES key.
XPRT-DES	Permits the export of the key using DES key. This is the default.
AES-key export control (one, optional). Key-management field 1, low-order byte.	

Table 16. Rule array keywords for AES PINPRW keys (continued)

Keyword	Meaning
NOEX-AES	Prohibits the export of the key using AES key.
XPRT-AES	Permits the export of the key using AES key. This is the default.
RSA-key export control (one, optional). Key-management field 1, low-order byte.	
NOEX-RSA	Prohibits the export of the key using RSA key.
XPRT-RSA	Permits the export of the key using RSA key. This is the default.

Figure 10 shows all the valid keyword combinations and their defaults for AES key type SECMSG. For a description of these keywords, see Table 17 on page 46.

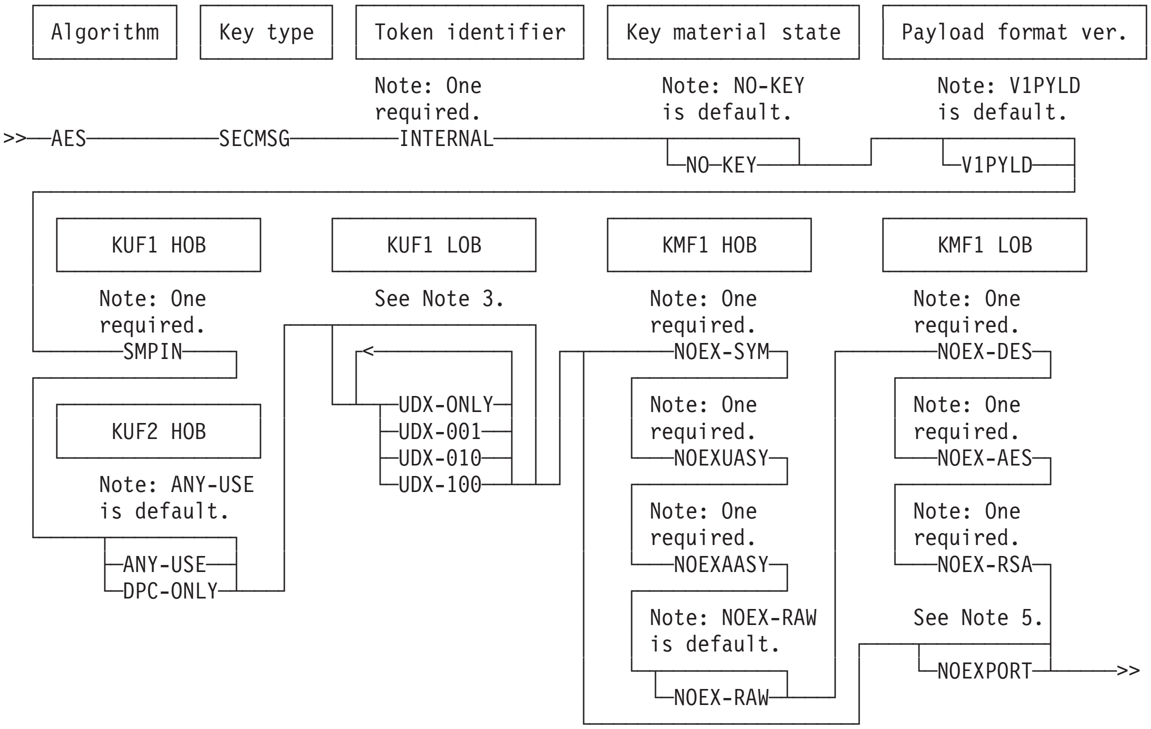


Figure 10. Key Token Build2 keyword combinations for AES SECMSG keys

Notes:

1. An AES SECMSG key is always derived. The derived key is the result of a key derivation function (KDF) applied to a fixed diversified key generating key (DKYGENKY) and derivation data. The final derived key is used as a session key and is typically used to encipher and decipher PIN information between devices. An AES SECMSG key can only be wrapped by an AES master key and cannot be stored in an external key-token.
2. Each key-usage field (KUF) and key-management field (KMF) of a version X'05' variable-length symmetric key-token consists of two bytes: a high-order byte (HOB) and a low-order byte (LOB).
3. Choose any number of keywords in this group. No keywords in the group are defaults.
4. NOEX-RAW is defined for future use and its meaning is currently undefined.

5. There is no default. Specifying NOEXPORT is equivalent to specifying all of the export control keywords (NOEX-SYM, NOEXUASY, NOEXAASY, NOEX-RAW, NOEX-DES, NOEX-AES, and NOEX-RSA). Do not specify any export control keywords together with NOEXPORT.

Table 17. Rule array keywords for AES SECMSG keys

Keyword	Meaning
Key-token header section	
Token identifier (one required).	
INTERNAL	Build a key token that is to be used locally.
Wrapping-information section	
Key status (one, optional).	
NO-KEY	Build a key token that does not contain a key value. This is the default.
Payload format version (one, optional). Identifies format of the payload.	
V1PYLD	Build the key token with a version 1 payload format. This format has a fixed length and the key length cannot be inferred by the size of the payload. An obscured key length is considered more secure.
Associated data section	
Algorithm type (one required).	
AES	Key can be used for AES algorithm.
Key type (one required).	
SECMSG	Key can be used as an EMV secure messaging key for encrypting PINs or for encrypting keys.
Generate control (one required). Secure message encryption enablement (one required). Key-usage field 1, high-order byte.	
SMPIN	Enable the encryption of PINs in an EMV secure message.
User-defined extension (UDX) control (one or more, optional). Key-usage field 1, low-order byte. No keywords in the group are defaults.	
UDX-ONLY	Key can only be used in UDXs.
UDX-001	Specifies that the rightmost user-defined UDX bit is set.
UDX-010	Specifies that the middle user-defined UDX bit is set.
UDX-100	Specifies that the leftmost user-defined UDX bit is set.
Service restriction (one, optional). Key-usage field 2, high-order byte.	
ANY-USE	Any service can use this key. This is the default.
DPC-ONLY	Only CSNBDPC can use this key.
General export control (one, optional). Equivalent to specifying all export control keywords (NOEX-SYM, NOEXUASY, NOEXAASY, NOEX-RAW, NOEX-DES, NOEX-AES, and NOEX-RSA). Not valid with any other export control keyword. There is no default. Key-management field 1, high-order byte and low-order byte.	
NOEXPORT	Prohibits the export of this key in all cases. Equivalent to specifying NOEX-SYM, NOEXUASY, NOEXAASY, NOEX-RAW, NOEX-DES, NOEX-AES, and NOEX-RSA.
Symmetric-key export control (one required if NOEXPORT not specified, otherwise not valid). Key-management field 1, high-order byte.	
NOEX-SYM	Prohibits the export of the key with a symmetric key.
Unauthenticated asymmetric-key export control (one required if NOEXPORT not specified, otherwise not valid). Key-management field 1, high-order byte.	
NOEXUASY	Prohibits the export of the key with an unauthenticated asymmetric key.

Table 17. Rule array keywords for AES SECMSG keys (continued)

Keyword	Meaning
Authenticated asymmetric-key export control (one required if NOEXPORT not specified, otherwise not valid). Key-management field 1, high-order byte.	
NOEXAASY	Prohibits the export of the key with an authenticated asymmetric key.
RAW-key export control (one required if NOEXPORT not specified, otherwise not valid). Key-management field 1, high-order byte.	
NOEX-RAW	Prohibits the export of the key in RAW format. This is the default.
DES-key export control (one required if NOEXPORT not specified, otherwise not valid). Key-management field 1, low-order byte.	
NOEX-DES	Prohibits the export of the key using DES key.
AES-key export control (one required if NOEXPORT not specified, otherwise not valid). Key-management field 1, low-order byte	
NOEX-AES	Prohibits the export of the key using AES key.
RSA-key export control (one required if NOEXPORT not specified, otherwise not valid). Key-management field 1, low-order byte.	
NOEX-RSA	Prohibits the export of the key using RSA key.

Required hardware

No cryptographic hardware is required by this callable service.

Verifying Data Integrity and Authenticating Messages

MAC Verify2 (CSNBMVR2, CSNBMVR3, CSNEMVR2, and CSNEMVR3)

Use the MAC Verify2 callable service to verify a keyed hash message authentication code (HMAC) or a ciphered message authentication code (CMAC) for the message text provided as input. A MAC key with key usage that can be used for verify is required to verify the MAC.

The MAC verify key must be in a variable-length HMAC key token for HMAC and an AES MAC token for CMAC.

The callable service names for AMODE(64) are CSNEMVR2 and CSNEMVR3.

Choosing between CSNBMVR2 and CSNBMVR3

CSNBMVR2 and CSNBMVR3 provide identical functions. When choosing which service to use, consider the following:

- CSNBMVR2 requires the application-supplied text to reside in the caller's primary address space.
- CSNBMVR3 allows the application-supplied text to reside either in the caller's primary address space or in a data space. This allows you to process more data with one call. For CSNBMVR3, *text_id_in* is an access list entry token (ALET) parameter of the data space containing the application-supplied text.

Format

```
CALL CSNBMVR2(
          return_code,
          reason_code,
          exit_data_length,
          exit_data,
             rule_array_count,
             rule_array,
             key_identifier_length,
             key_identifier,
             text_length,
             text,
             chaining_vector_length,
             chaining_vector,
             mac_length,
             mac )
CALL CSNBMVR3(
             return_code,
             reason_code,
             exit_data_length,
             exit_data,
             rule_array_count,
             rule_array,
             key_identifier_length,
             key_identifier,
             text_length,
             text,
             chaining_vector_length,
             chaining_vector,
             mac_length,
             mac,
             text_id_in )
```

Parameters

return_code

Direction	Type
Output	Integer

The return code specifies the general result of the callable service.

reason_code

Direction	Type
Output	Integer

The reason code specifies the result of the callable service that is returned to the application program. Each return code has different reason codes that indicate specific processing problems.

exit_data_length

Direction	Type
Input/Output	Integer

The length of the data that is passed to the installation exit. The data is identified in the *exit_data* parameter.

exit_data

Direction	Type
Input/Output	String

The data that is passed to the installation exit.

rule_array_count

Direction	Type
Input	Integer

The number of keywords you supplied in the *rule_array* parameter. The value must be 1, 2, or 3.

rule_array

Direction	Type
Input	String

The *rule_array* contains keywords that provide control information to the callable service. The keywords must be in contiguous storage with each of the keywords left-justified in its own 8-byte location and padded on the right with blanks.

Table 18. Keywords for MAC Verify2 Control Information

Keyword	Meaning
Token algorithm (One required)	
AES	Specifies the use of the AES CMAC algorithm to generate a MAC.
HMAC	Specifies the use of the HMAC algorithm to generate a MAC.
Hash method (One required for HMAC only)	
SHA-1	Specifies the use of the SHA-1 hash method.
SHA-224	Specifies the use of the SHA-224 hash method.
SHA-256	Specifies the use of the SHA-256 hash method.
SHA-384	Specifies the use of the SHA-384 hash method.
SHA-512	Specifies the use of the SHA-512 hash method.
Segmenting Control (One optional)	
FIRST	First call, this is the first segment of data from the application program.
LAST	Last call; this is the last data segment.
MIDDLE	Middle call; this is an intermediate data segment.
ONLY	Only call; segmenting is not employed by the application program. This is the default value.

key_identifier_length

Direction	Type
Input	Integer

key_identifier_length specifies the length in bytes of the *key_identifier* parameter. If the *key_identifier* parameter contains a label, the value must be 64. Otherwise, the value must be between the actual length of the token and 725.

key_identifier

Direction	Type
Input/Output	String

The identifier of the key to verify the MAC. The *key identifier* is an operational token or the key label of an operational token in key storage.

For the HMAC algorithm, the key algorithm must be HMAC and the key usage fields must indicate GENERATE or VERIFY and the hash method selected. For the AES algorithm, the key algorithm must be AES, the key type must be MAC, and the key usage fields must indicate GENERATE or VERIFY and must indicate CMAC.

If the token supplied was encrypted under the old master key, the token is returned encrypted under the current master key.

text_length

Direction	Type
Input	Integer

The length of the text you supplied in the *text* parameter. The maximum length of *text* is 214783647 bytes. For FIRST and MIDDLE calls, the *text_length* must be:

- A multiple of 64 for the SHA-1, SHA-224, and SHA-256 hash methods.
- A multiple of 128 for the SHA-384 and SHA-512 hash methods.
- A multiple of 16 for the AES CMAC method.

text

Direction	Type
Input	String

The application-supplied text for which the MAC is generated.

chaining_vector_length

Direction	Type
Input/Output	Integer

chaining_vector_length specifies the length in bytes of the *chaining_vector* parameter. The value must be 128.

chaining_vector

Direction	Type
Input/Output	String

An 128-byte string that ICSF uses as a system work area. Your application program must not change the data in this string. The chaining vector permits data to be chained from one invocation call to another.

On the first call, initialize this parameter as binary zeros.

mac_length

Direction	Type
Input	Integer

The length of the *mac* parameter in bytes. For HMAC, the maximum value is 64. For AES, the value must be 8 or 16.

mac

Direction	Type
Input	String

The field that contains the MAC value you want to verify.

text_id_in

Direction	Type
Input	Integer

For CSNBMVR3 only, the ALET of the text for which the MAC is to be verified.

Usage notes

SAF may be invoked to verify the caller is authorized to use this callable service, the key label, or internal secure key tokens that are stored in the CKDS.

Access control points

This table lists the access control points in the domain role that control the function for this service.

Table 19. MAC Verify2 Access Control Points

Hash method	Access control point
CMAC	MAC Verify2 - AES CMAC
SHA-1	HMAC Verify - SHA-1
SHA-224	HMAC Verify - SHA-224
SHA-256	HMAC Verify - SHA-256
SHA-384	HMAC Verify - SHA-384
SHA-512	HMAC Verify - SHA-512

Required hardware

This table lists the required cryptographic hardware for each server type and describes restrictions for this callable service.

Table 20. MAC Verify2 required hardware

Server	Required cryptographic hardware	Restrictions
IBM eServer zSeries 990 IBM eServer zSeries 890		This service is not supported.
IBM System z9 EC IBM System z9 BC		This service is not supported.
IBM System z10 EC IBM System z10 BC		This service is not supported.
IBM zEnterprise 196 IBM zEnterprise 114	Crypto Express3 Coprocessor	Requires the March 2014 or later licensed internal code (LIC). For AES, a MAC length of 8 is not supported.

Table 20. MAC Verify2 required hardware (continued)

Server	Required cryptographic hardware	Restrictions
IBM zEnterprise EC12 IBM zEnterprise BC12	Crypto Express3 Coprocessor	Requires the March 2014 or later licensed internal code (LIC).
	Crypto Express4 CCA Coprocessor	For AES, a MAC length of 8 requires the June 2015 or later licensed internal code (LIC).
IBM z13	Crypto Express5 CCA Coprocessor	For AES, a MAC length of 8 requires the July 2015 or later licensed internal code (LIC).

Financial Services for DK PIN Methods

DK PIN Change (CSNBDPC and CSNEDPC)

Use the DK PIN Change callable service to allow a customer to change their PIN to a value of their choosing.

The current and new PINs are entered into the ATM, where they are encrypted into ISO-1 PIN blocks. The PIN and other needed information are used to verify the current PIN. If the PIN does not verify, the process is aborted. If the PIN does verify, the PIN is reformatted into a PBF-O format and the provided information is used to create a new PIN reference value.

Note: Regarding weak PINs, if the new PIN specified appears in the weak PIN table, the PIN change fails with an indication that the selected new PIN was not valid.

The callable service name for AMODE(64) invocation is CSNEDPC.

Format

```
CALL CSNBDPC(
             return_code,
             reason_code,
             exit_data_length,
             exit_data,
             rule_array_count,
             rule_array,
             PAN_data_length,
             PAN_data,
             card_p_data_length,
             card_p_data,
             card_t_data_length,
             card_t_data,
             cur_ISO1_PIN_block_length,
             cur_ISO1_PIN_block,
             new_ISO1_PIN_block_length,
             new_ISO1_PIN_block,
             card_script_data_length,
             card_script_data,
             script_offset,
             script_offset_field_length,
             script_initialization_vector_length,
             script_initialization_vector,
             output_PIN_profile,
             PIN_reference_value_length,
             PIN_reference_value,
             PRW_random_number_length,
             PRW_random_number,
            PRW_key_identifier_length,
            PRW_key_identifier,
            cur_IPIN_encryption_key_identifier_length,
            cur_IPIN_encryption_key_identifier,
            new_IPIN_encryption_key_identifier_length,
            new_IPIN_encryption_key_identifier,
            script_key_identifier_length,
            script_key_identifier,
            script_MAC_key_identifier_length,
            script_MAC_key_identifier,
            new_PRW_key_identifier_length,
            new_PRW_key_identifier,
            OPIN_encryption_key_identifier_length,
            OPIN_encryption_key_identifier,
            OEPB_MAC_key_identifier_length,
            OEPB_MAC_key_identifier,
            script_length,
            script,
            script_MAC_length,
            script_MAC,
            new_PIN_reference_value_length,
            new_PIN_reference_value,
            new_PRW_random_number_length,
            new_PRW_random_number,
            output_encrypted_PIN_block_length,
            output_encrypted_PIN_block,
            PIN_block_MAC_length,
            PIN_block_MAC)
```

Parameters

return_code

Direction	Type
Output	Integer

The return code specifies the general result of the callable service.

reason_code

Direction	Type
Output	Integer

The reason code specifies the result of the callable service that is returned to the application program. Each return code has different reason codes assigned to it that indicate specific processing problems.

exit_data_length

Direction	Type
Input/Output	Integer

The length of the data that is passed to the installation exit. The data is identified in the *exit_data* parameter.

exit_data

Direction	Type
Input/Output	String

The data that is passed to the installation exit.

rule_array_count

Direction	Type
Input	Integer

The number of keywords you supplied in the *rule_array* parameter. The value must be 0, 1, 2, 3, 4, or 5.

rule_array

Direction	Type
Input	Character

Keywords that provide control information to the callable service. The keywords must be in contiguous storage with each of the keywords left-justified in its own 8-byte location and padded on the right with blanks.

Table 21. Rule array keywords for the Card Replace PRW Generate Service

Keyword	Meaning
PIN Block output selection keyword (One, optional)	
NOEPB	Do not return an encrypted PIN block (EPB). This is the default value.
EPB	Return an encrypted PIN block and a MAC to verify the encrypted PIN block.
Script selection algorithm and method keyword (One, optional)	
AES-CBC	Specifies to use CBC mode to AES encrypt the script.
NOSCRIPT	Do not return an encrypted SMPIN message with a MAC. This is the default value.
TDES-CBC	Specifies to use CBC mode to TDES encrypt the script.
TDES-ECB	Specifies to use ECB mode to TDES encrypt the script.
Pin encryption keyword (One, optional) Only valid if AES-CBC, TDES-CBC or TDES-ECB is selected above.	
CLEARPIN	Do not encrypt the PIN prior to inserting in the script block. This is the default value.
SELF-ENC	Copy the PIN-block self-encrypted to the clear PIN block within the clear output message. Use this rule array keyword to specify that the 8-byte PIN block shall be used as a DES key to encrypt the PIN block. The service copies the self-encrypted PIN block to the clear PIN block in the output message.
MAC Ciphering Method (One, optional) Only valid if TDES-CBC or TDES-ECB is selected above.	
CMAC	Specifies to use the cipher-based MAC algorithm block cipher mode of operation for authentication, recommended in NIST SP 800-38B. Required for AES-CBC. Only valid with AES-CBC.
EMVMACD	Specifies the EMV-related message-padding and calculation method. Not valid with AES-CBC.
TDES-MAC	Specifies the ANS X9.9 Option 1 (binary data) procedure and a CBC Triple-DES encryption of the data. Not valid with AES-CBC.

Table 21. Rule array keywords for the Card Replace PRW Generate Service (continued)

Keyword	Meaning
X9.19OPT	Specifies the ANS X9.19 Optional Procedure. A double-length key is required. This is the default value. Not valid with AES-CBC. This is the default value for TDES-CBC and TDES-ECB.
MAC Length and presentation (One, optional) Only valid if AES-CBC, TDES-CBC or TDES-ECB is selected above.	
MACLEN8	Specifies a 8-byte MAC. This is the default value for TDES-CBC and TDES-ECB.
MACLEN16	Specifies a 16-byte MAC. Only valid with CMAC. This is the default for AES-CBC.

PAN_data_length

Direction	Type
Input	Integer

Specifies the length in bytes of the *PAN_data* parameter. The value must be between 10 and 19, inclusive.

PAN_data

Direction	Type
Input	Character

The PAN data which the PIN is associated. The full account number, including check digit, should be included. This parameter is character data.

card_p_data_length

Direction	Type
Input	Integer

Specifies the length in bytes of the *card_p_data* parameter. The value must be between 2 and 256, inclusive.

card_p_data

Direction	Type
Input	String

The time-invariant card data (CDp), determined by the card issuer, which is used to differentiate between multiple cards for one account.

card_t_data_length

Direction	Type
Input	Integer

Specifies the length in bytes of the *card_t_data* parameter. The value must be between 2 and 256, inclusive.

card_t_data

Direction	Type
Input	String

The time-sensitive card data, determined by the card issuer, which, together with the account number and the card_p_data, specifies an individual card.

cur_ISO1_PIN_block_length

Direction	Type
Input	Integer

Specifies the length in bytes of the *cur_ISO1_PIN_block* parameter. This value must be 8.

cur_ISO1_PIN_block

Direction	Type
Input	String

The 8-byte encrypted PIN block with the current PIN in ISO-1 format.

new_ISO1_PIN_block_length

Direction	Type
Input	Integer

Specifies the length in bytes of the *new_ISO1_PIN_block* parameter. This value must be 8.

new_ISO1_PIN_block

Direction	Type
Input	String

The new encrypted PIN block with the customer chosen PIN. The PIN block must be in ISO-1 format.

card_script_data_length

Direction	Type
Input	Integer

Specifies the length in bytes of the *card_script_data* parameter. If NOSCRIPT is specified in the rule array, this value must be 0. The length must be a multiple of 8 when TDES-CBC or TDES-ECB is specified in the rule array or a multiple of 16 when AES-CBC is specified. The value must be no greater than 4096.

card_script_data

Direction	Type
Input	String

The clear text string to be updated with the clear PIN block and encrypted.

script_offset

Direction	Type
Input	Integer

The offset to the location for the PIN block in the script. Specify the first byte of the clear text as offset 0. This offset plus the value of *script_offset_field_length* must be less than or equal to the *card_script_data_length*. If NOSCRIPT is specified in the rule array, this parameter is ignored.

script_offset_field_length

Direction	Type
Input	Integer

The length of the field within *card_script_text* parameter at *script_offset* where the new PIN value is to be placed. Length must be 8. The PIN block must fit entirely within the *card_script_text*. If NOSCRIPT is specified in the rule array, this parameter is ignored.

script_initialization_vector_length

Direction	Type
Input	Integer

Specifies the length in bytes of the *script_initialization_vector* parameter. For script selection algorithm and method keyword AES-CBC, the value must be 16. For TDES-CBC, the value must be 8. Otherwise, the value must be 0.

script_initialization_vector

Direction	Type
Input	String

The 8-byte or 16 byte initialization data for encrypting the script. The value of this parameter must be a string of hexadecimal zeroes. If the *script_initialization_vector_length* is 0, this parameter is ignored.

output_PIN_profile

Direction	Type
Input	String

A 24-byte string containing the PIN profile, including the PIN block format for the script. See 'The PIN Profile' for additional information. You can use PIN-block formats ISO-0, ISO-1, ISO-2, and ISO-3 with this service. If NOSCRIPT is specified in the rule array, this parameter is ignored.

PIN_reference_value_length

Direction	Type
Input	Integer

Specifies the length in bytes of the *PIN_reference_value* parameter. This value must be 16.

PIN_reference_value

Direction	Type
Input	String

The 16-byte PIN reference value of the current PIN for comparison to the calculated value.

PRW_random_number_length

Direction	Type
Input	Integer

Specifies the length in bytes of the *PRW_random_number* parameter. The value must be 4.

PRW_random_number

Direction	Type
Input	String

The 4-byte random number associated with the PIN reference value of the current PIN.

PRW_key_identifier_length

Direction	Type
Input	Integer

Specifies the length in bytes of the *PRW_key_identifier* parameter. If the *PRW_key_identifier* contains a label, the length must be 64. Otherwise, the value must be between the actual length of the token and 725.

PRW_key_identifier

Direction	Type
Input/Output	String

The identifier of the key to verify the PRW of the current PIN block. The key identifier is an operational token or the key label of an operational token in key storage. The key algorithm of this key must be AES, the key type must be PINPRW, and the key usage fields must indicate VERIFY, CMAC, and DKPINOP.

If the token supplied was encrypted under the old master key, the token will be returned encrypted under the current master key.

cur_IPIN_encryption_key_identifier_length

Direction	Type
Input	Integer

Specifies the length in bytes of the *cur_IPIN_encryption_key_identifier* parameter. If the *cur_IPIN_encryption_key_identifier* contains a label, the length must be 64. Otherwise, the value must be between the actual length of the token and 725.

cur_IPIN_encryption_key_identifier

Direction	Type
Input/Output	String

The identifier of the key to decrypt the PIN_block containing the current PIN. The key identifier is an operational token or the key label of an operational token in key storage. The key algorithm of this key must be DES and the key type must be IPINENC.

If the token supplied was encrypted under the old master key, the token will be returned encrypted under the current master key.

new_IPIN_encryption_key_identifier_length

Direction	Type
Input	Integer

Specifies the length in bytes of the *new_IPIN_encryption_key_identifier* parameter. If the *new_IPIN_encryption_key_identifier* contains a label, the length must be 64. Otherwise, the value must be between the actual length of the token and 725.

new_IPIN_encryption_key_identifier

Direction	Type
Input/Output	String

The identifier of the key to decrypt the PIN_block containing the new PIN. The key identifier is an operational token or the key label of an operational token in key storage. The key algorithm of this key must be DES and the key type must be IPINENC.

If the token supplied was encrypted under the old master key, the token will be returned encrypted under the current master key.

script_key_identifier_length

Direction	Type
Input	Integer

Specifies the length in bytes of the *script_key_identifier* parameter. If the rule array indicates that no script is to be processed, this value must be 0. If the *script_key_identifier* contains a label, the length must be 64. Otherwise, the value must be between the actual length of the token and 725.

script_key_identifier

Direction	Type
Input/Output Ignored	String

The identifier of the key for encryption of the script. The key identifier is an operational token or the key label of an operational token in key storage. For script selection algorithm and method keyword AES-CBC, the key algorithm of the key must be AES, the key type must be SECMSG, and the key usage fields must indicate SMPIN and allow use by the CSNBDPC service (ANY-USE or

DPC-ONLY). For keywords TDES-CBC or TDES-ECB, the key algorithm of this key must be DES, the key type must be SECMSG with the SMPIN usage bit (CV bit 19) set to B'1'.

If the token supplied was encrypted under the old master key, the token will be returned encrypted under the current master key.

script_MAC_key_identifier_length

Direction	Type
Input	Integer

Specifies the length in bytes of the *script_MAC_key_identifier* parameter. If the rule array indicates that no script is to be processed, this value must be 0. If the *script_MAC_key_identifier* contains a label, the length must be 64. Otherwise, the value must be between the actual length of the token and 725.

script_MAC_key_identifier

Direction	Type
Input/Output	String

The identifier of the key to generate the MAC of the script. The key identifier is an operational token or the key label of an operational token in key storage. For script selection algorithm and method keyword AES-CBC, the key algorithm of the key must be AES, the key type must be MAC, and the key usage fields must indicate GENERATE or GENONLY and CMAC. For keywords TDES-CBC or TDES-ECB, the key algorithm of this key must be DES, the key type must be MAC, and the key must be double-length.

If the token supplied was encrypted under the old master key, the token will be returned encrypted under the current master key.

new_PRW_key_identifier_length

Direction	Type
Input	Integer

Specifies the length in bytes of the *new_PRW_key_identifier* parameter. If the *new_PRW_key_identifier* contains a label, the length must be 64. Otherwise, the value must be between the actual length of the token and 725.

new_PRW_key_identifier

Direction	Type
Input/Output	String

The identifier of the key to verify the new PRW. The key identifier is an operational token or the key label of an operational token in key storage. The key algorithm of this key must be AES, the key type must be PINPRW, and the key usage fields must indicate GENONLY, CMAC, and DKPINOP.

If the token supplied was encrypted under the old master key, the token will be returned encrypted under the current master key.

OPIN_encryption_key_identifier_length

Direction	Type
Input	Integer

Specifies the length in bytes of the *OPIN_encryption_key_identifier* parameter. If the rule array indicates that no encrypted PIN block is to be returned, this value must be 0. If the *OPIN_encryption_key_identifier* contains a label, the length must be 64. Otherwise, the value must be between the actual length of the token and 725.

OPIN_encryption_key_identifier

Direction	Type
Input/Output	String

The identifier of the key to encrypt the new PIN block. The key identifier is an operational token or the key label of an operational token in key storage. If the *OPIN_encryption_key_identifier_length* is 0, this parameter is ignored. The key algorithm of this key must be AES, the key type must be PINPROT, and the key usage fields must indicate ENCRYPT, CBC, and DKPINOP.

If the token supplied was encrypted under the old master key, the token will be returned encrypted under the current master key.

OEPB_MAC_key_identifier_length

Direction	Type
Input	Integer

Specifies the length in bytes of the *OEPB_MAC_key_identifier* parameter. If the rule array indicates that no encrypted PIN block MAC is to be returned, this value must be 0. If the *OEPB_MAC_key_identifier* contains a label, the length must be 64. Otherwise, the value must be between the actual length of the token and 725.

OEPB_MAC_key_identifier

Direction	Type
Input/Output	String

The identifier of the key to generate the MAC of new PIN block. The key identifier is an operational token or the key label of an operational token in key storage. If the *OEPB_MAC_key_identifier_length* is 0, this parameter is ignored. The key algorithm of this key must be AES, the key type must be MAC, and the key usage fields must indicate CMAC, GENONLY, and DKPINOP.

If the token supplied was encrypted under the old master key, the token will be returned encrypted under the current master key.

script_length

Direction	Type
Input/Output	Integer

Specifies the length in bytes of the *script* parameter. If the rule array specifies TDES-CBC or TDES-ECB, this value must be at least as long as the *script* parameter. Otherwise, it must be 0.

script

Direction	Type
Output	String

The encrypted output script. The length of the field must be at least as long as the input script.

script_MAC_length

Direction	Type
Input/Output	Integer

Specifies the length in bytes of the *script_MAC* parameter. If the NOSCRIPT keyword is selected, this value must be 0. Otherwise, this value must be at least as large as indicated by the MAC length keyword specified. On output, the parameter is updated with the actual length of the *script_MAC* parameter.

script_MAC

Direction	Type
Output	String

The 8 byte or 16 byte MAC of the encrypted script. If the *script_MAC_length* is 0, this parameter is ignored.

new_PIN_reference_value_length

Direction	Type
Input/Output	Integer

Specifies the length in bytes of the *new_PIN_reference_value* parameter. The value must be at least 16. On output, it will be set to 16.

new_PIN_reference_value

Direction	Type
Output	String

The 16-byte new PIN reference value of the new PIN block.

new_PRW_random_number_length

Direction	Type
Input/Output	Integer

Specifies the length in bytes of the *new_PRW_random_number* parameter. The value must be at least 4. On output, it will be set to 4.

new_PRW_random_number

Direction	Type
Output	String

The 4-byte random number associated with the new PIN reference value.

output_encrypted_PIN_block_length

Direction	Type
Input/Output	Integer

Specifies the length in bytes of the *output_encrypted_PIN_block* parameter. If the rule array indicates that no encrypted PIN block should be returned, this value must be 0. Otherwise, it should be at least 32. On output it will be set to 32.

output_encrypted_PIN_block

Direction	Type
Output	String

The 32-byte encrypted new PIN block. If the *output_encrypted_PIN_block_length* is 0, this parameter is ignored.

PIN_block_MAC_length

Direction	Type
Input/Output	Integer

Specifies the length in bytes of the *PIN_block_MAC* parameter. If the rule_array indicates that no PIN block MAC should be returned, this value must be 0. Otherwise, it must be at least 8.

PIN_block_MAC

Direction	Type
Output	String

The 8-byte MAC of the new encrypted PIN block. If the *PIN_block_MAC_length* is 0, this parameter is ignored.

Usage notes

SAF may be invoked to verify the caller is authorized to use this callable service, the key label, or internal secure key tokens that are stored in the CKDS.

Access control points

The **DK PIN Change** access control point in the domain role controls the function of this service.

Required hardware

This table lists the required cryptographic hardware for each server type and describes restrictions for this callable service.

DK PIN Change

Table 22. DK PIN Change required hardware

Server	Required cryptographic hardware	Restrictions
IBM eServer zSeries 990 IBM eServer zSeries 890		This service is not supported.
IBM System z9 EC IBM System z9 BC		This service is not supported.
IBM System z10 EC IBM System z10 BC		This service is not supported.
IBM zEnterprise 196 IBM zEnterprise 114	Crypto Express3 Coprocessor	DK AES PIN key support requires the November 2013 or later licensed internal code (LIC). Rule array keywords AES-CBC and CMAC are not supported.
IBM zEnterprise EC12 IBM zEnterprise BC12	Crypto Express3 Coprocessor Crypto Express4 CCA Coprocessor	DK AES PIN key support requires the September 2013 or later licensed internal code (LIC). Rule array keywords AES-CBC and CMAC require the June 2015 or later licensed internal code (LIC).
IBM z13	Crypto Express5 CCA Coprocessor	Rule array keywords AES-CBC, CMAC, and MACLEN16 require the July 2015 or later licensed internal code (LIC).

Reason codes for return code 8 (8)

Table 23. Reason codes for return code 8 (8)

Reason Code Hex (Decimal)	Description
8B5 (2229)	The type of key specified is not valid because a diversified key generating key must be used to derive this symmetric key type. **User action**: Supply a valid key type or token for the service.

Appendix B. Key Token Formats

Variable-length symmetric key token

The following table presents the presents the format for a variable-length symmetric key token. The length of the token depends on the key type and algorithm.

Table 24. Variable-length symmetric key token

Offset (Dec)	Length of Field (Bytes)	Description
		Header
0	1	Token flag **X'00'** for null tokens **X'01'** for internal tokens **X'02'** for external tokens
1	1	Reserved (X'00')

Table 24. Variable-length symmetric key token (continued)

Offset (Dec)	Length of Field (Bytes)	Description
2	2	Length of the token in bytes
4	1	Token version number X'05' (May be X'00' for null tokens)
5	3	Reserved (X'000000')
		Wrapping information
8	1	Key material state. **X'00'** no key present (internal or external) **X'01'** key is clear (internal) **X'02'** key is encrypted under a key-encrypting key (external) **X'03'** key is encrypted under the master key (internal)
9	1	Key verification pattern (KVP) type. **X'00'** No KVP **X'01'** AES master key verification pattern **X'02'** key-encrypting key verification pattern
10	16	Verification pattern of the key used to wrap the payload. Value is left justified.
26	1	Wrapping method - This value indicates the wrapping method used to protect the data in the encrypted section. **X'00'** key is in the clear **X'02'** AESKW **X'03'** PKOAEP2
27	1	Hash algorithm used in wrapping algorithm. • For wrapping method X'00' **X'00'** None. For clear key tokens. • For wrapping method X'02' **X'02'** SHA-256 • For wrapping method X'03' **X'01'** SHA-1 **X'02'** SHA-256 **X'04'** SHA-384 **X'08'** SHA-512
28	1	Payload version **X'00'** Variable-length payload **X'01'** Fixed-length payload All other values are reserved and must not be used.
29	1	Reserved (X'00')
		Associated data section
30	1	Associated data version (X'01')
31	1	Reserved (X'00')
32	2	Length of the associated data in bytes: *adl*

Table 24. Variable-length symmetric key token (continued)

Offset (Dec)	Length of Field (Bytes)	Description
34	1	Length of the key name in bytes: *kl*
35	1	Length of the IBM extended associated data in bytes: *iead*
36	1	Length of the installation-definable associated data in bytes: *uad*
37	1	Reserved (X'00')
38	2	Length of the payload in bits: *pl*
40	1	Reserved (X'00')
41	1	Type of algorithm for which the key can be used **X'01'** DES **X'02'** AES **X'03'** HMAC
42	2	Key type: For algorithm AES: **X'0001'** CIPHER **X'0002'** MAC **X'0003'** EXPORTER **X'0004'** IMPORTER **X'0005'** PINPROT **X'0006'** PINCALC **X'0007'** PINPRW **X'0009'** DKYGENKY **X'000A'** SECMSG For algorithm HMAC: **X'0002'** MAC For algorithm DES: **X'0008'** DESUSECV
44	1	Key-usage field count (*kuf*) - (1 byte) Key-usage field information defines restrictions on the use of the key.

Table 24. Variable-length symmetric key token (continued)

Offset (Dec)	Length of Field (Bytes)	Description
45	*kuf* * 2	Key-usage fields (*kuf* * 2 bytes) • For HMAC algorithm keys, refer to Table 406 in *ICSF Application Programmer's Guide*. • For AES algorithm Key-Encrypting keys (Exporter or Importer), refer to Table 414 in *ICSF Application Programmer's Guide*. • For AES algorithm CIPHER keys, refer to Table 26 on page 70. • For AES algorithm MAC keys, refer to 407 in *ICSF Application Programmer's Guide*. • For AES algorithm PINCALC keys, refer to Table 408 in *ICSF Application Programmer's Guide*. • For AES algorithm PINPROT keys, refer to Table 409 in *ICSF Application Programmer's Guide*. • For AES algorithm PINPRW keys, refer to Table 410 in *ICSF Application Programmer's Guide*. • For AES algorithm DKYGENKY keys, refer to Table 25 on page 68. • For AES algorithm SECMSG keys, refer to Table 27 on page 71. • For DESUSECV keys, refer to Table 405 in *ICSF Application Programmer's Guide*.
45 + *kuf* * 2	1	Key-management field count (*kmf*) - (2 byte): • For AES and HMAC keys: 2 (no pedigree information) or 3 (has pedigree information) • For DESUSECV keys: 1 Key-management field information describes how the data is to be managed or helps with management of the key material.
46 + *kuf* * 2	*kuf* * 2	Key-management fields (kmf * 2 bytes): • For AES and HMAC algorithm keys, refer to Table 415 in *ICSF Application Programmer's Guide*. • For DESUSECV keys, refer to Table 416 in *ICSF Application Programmer's Guide*.
46 + *kuf* * 2 + *kmf* * 2	*kl*	Key name
46 + *kuf* * 2 + *kmf* * 2 + *kl*	*iead*	IBM extended associated data
46 + *kuf* * 2 + *kmf* * 2 + *kl* + *iead*	*uad*	Installation-defined associated data
		Clear key or encrypted payload
30 + *adl*	*(pl+7)/8*	**Encrypted AESKW payload (internal keys)**: The encrypted AESKW payload is created from the unencrypted AESKW payload which is made up of the ICV/pad length/hash options and hash length/hash options/hash of the associated data/key material/padding. See unencrypted AESKW payload below. **Encrypted PKOAEP2 payload (external keys)**: The encrypted PKOAEP2 payload is created using the PKCS #1 v1.2 encoding method for a given hash algorithm. The message (M) inside the encoding contains: [2 bytes: bit length of key] || [clear HMAC key]. M is encoded using OAEP and then encrypted with an RSA public key according to the standard. **Clear key payload**: When the key is clear, only the key material will be in the payload padded to the nearest byte with binary zeros.

Table 25. AES algorithm DKYGENKY key associated data

Offset (Dec)	Length of Field (Bytes)	Description
44	1	Key-usage field count (kuf): 2, 4, 5, or 6. Count is based on the type of key to diversify (value of offset 45): **Value at offset 45** **Type of key to diversify / *kuf* count** **X'00'** D-ALL / *kuf* count: 2 **X'01'** D-CIPHER / *kuf* count: 4 **X'02'** D-MAC / *kuf* count: 4 (not DK enabled) or 5 (DK enabled) **X'03'** D-EXP / *kuf* count: 6 **X'04'** D-IMP / *kuf* count: 6 **X'05'** D-PPROT / *kuf* count: 5 **X'06'** D-PCALC / *kuf* count: 5 **X'07'** D-PPRW / *kuf* count: 5 **X'08'** D-SECMSG / *kuf* count: 4 Each key-usage field is 2 bytes in length. The value in this field indicates how many 2-byte key usage fields follow.
45	2	Key-usage field 1 High-order byte: Defines the key type to be generated. **X'00'** Any type listed below (D-ALL) **X'01'** CIPHER (D-CIPHER) **X'02'** MAC (D-MAC) **X'03'** EXPORTER (D-EXP) **X'04'** IMPORTER (D-IMP) **X'05'** PINPROT (D-PPROT) **X'06'** PINCALC (D-PCALC) **X'07'** PINPRW (D-PPRW) **X'08'** SECMSG (D-SECMSG) All other values are reserved and undefined. Low-order byte: **xxxx 1xxx** The key can only be used in UDXs (used in KGN, KIM, KEX). **xxxx 0xxx** The key can be used in both UDXs and CCA. **xxxx xuuu** Reserved for UDXs, where *uuu* are UDX-defined bits. All unused bits are reserved and must be zero.

Table 25. AES algorithm DKYGENKY key associated data (continued)

Offset (Dec)	Length of Field (Bytes)	Description
47	2	Key-usage field 2: Indicates the key usage. High-order byte (key-usage field level of control): **B'1xxx xxxx'** The key usage fields of the key to be generated must be equal (KUF-MBE) to the related generated key usage fields that start with key usage field 3 below. **B'0xxx xxxx'** The key usage fields of the key identifier to be generated must be permitted (KUF-MBP) based on the related generated-key usage fields that start with key usage field 3 below. A key to be diversified is not permitted to have a higher level of usage than the related key usage fields permit. The key to be diversified is only permitted to have key usage that is less than or equal to the related key usage fields. The UDX-ONLY bit of the related key usage fields must always be equal in both the generating key and the generated key. Undefined when the value at offset 45 = X'00' (D-ALL). All other values are reserved and undefined. Low-order byte (key-derivation sequence level): **X'00'** DKYL0. Generate a key based on the key usage byte at offset 45. **X'01'** DKYL1. Generate a level 0 diversified key with key type DKYGENKY. **X'02'** DKYL2. Generate a level 1 diversified key with key type DKYGENKY. All other values are reserved and undefined.
49 (if defined)	2	Key-usage field 3 (related generated key usage fields): These values determine allowable key usage of key to be generated. Meaning depends on value of offset 45: **X'01'** Same as key-usage field 1 of AES CIPHER key. **X'02'** Same as key-usage field 1 of AES MAC key. **X'03'** Same as key-usage field 1 of AES EXPORTER key. **X'04'** Same as key-usage field 1 of AES IMPORTER key. **X'05'** Same as key-usage field 1 of AES PINPROT key. **X'06'** Same as key-usage field 1 of AES PINCALC key. **X'07'** Same as key-usage field 1 of AES PINPRW key. **X'08'** Same as key-usage field 1 of AES SECMSG key.

Table 25. AES algorithm DKYGENKY key associated data (continued)

Offset (Dec)	Length of Field (Bytes)	Description
51 (if defined)	2	Key-usage field 4 (related generated key usage fields): These values determine allowable key usage of key to be generated. Meaning depends on value of offset 45: **X'01'** Same as key-usage field 2 of AES CIPHER key. **X'02'** Same as key-usage field 2 of AES MAC key. **X'03'** Same as key-usage field 2 of AES EXPORTER key. **X'04'** Same as key-usage field 2 of AES IMPORTER key. **X'05'** Same as key-usage field 2 of AES PINPROT key. **X'06'** Same as key-usage field 2 of AES PINCALC key. **X'07'** Same as key-usage field 2 of AES PINPRW key. **X'08'** Same as key-usage field 2 of AES SECMSG key.
53 (if defined)	2	Key-usage field 5 (related generated key usage fields): These values determine allowable key usage of key to be generated. Meaning depends on value of offset 45: **X'02'** Same as key-usage field 3 of AES MAC key. **X'03'** Same as key-usage field 3 of AES EXPORTER key. **X'04'** Same as key-usage field 3 of AES IMPORTER key. **X'05'** Same as key-usage field 3 of AES PINPROT key. **X'06'** Same as key-usage field 3 of AES PINCALC key. **X'07'** Same as key-usage field 3 of AES PINPRW key.
55 (if defined)	2	Key-usage field 6 (related generated key usage fields): These values determine allowable key usage of key to be generated. Meaning depends on value of offset 45: **X'03'** Same as key-usage field 4 of AES EXPORTER key. **X'04'** Same as key-usage field 4 of AES IMPORTER key.

Table 26. AES algorithm CIPHER Key associated data

Offset (Dec)	Length of Field (Bytes)	Description
44	1	Key-usage field count (*kuf*): 2

Table 26. AES algorithm CIPHER Key associated data (continued)

Offset (Dec)	Length of Field (Bytes)	Description
45	2	Key-usage field 1 High-order byte: **1xxx xxxx** Key can be used for encryption. **x1xx xxxx** Key can be used for decryption. **xx1x xxxx** Key can be used for cipher text translate only. All unused bits are reserved and must be zero. Low-order byte: **xxxx 1xxx** The key can only be used in UDXs (used in KGN, KIM, KEX). **xxxx 0xxx** The key can be used in both UDXs and CCA. **xxxx xuuu** Reserved for UDXs, where uuu are UDX-defined bits. All unused bits are reserved and must be zero.
47	2	Key-usage field 2 High-order byte: **X'00'** Key can be used for Cipher Block Chaining (CBC). **X'01'** Key can be used for Electronic Code Book (ECB). **X'02'** Key can be used for Cipher Feedback (CFB). **X'03'** Key can be used for Output Feedback (OFB). **X'04'** Key can be used for Galois/Counter Mode (GCM) **X'05'** Key can be used for XEX-based Tweaked CodeBook Mode with CipherText Stealing (XTS) **X'FF'** Key can be used for any mode of encryption All unused values are reserved and must not be used. Low-order byte: All bits are reserved and must be zero.

Table 27. AES algorithm SECMSG key associated data

Offset (Dec)	Length of Field (Bytes)	Description
44	1	Key-usage field count (*kuf*): 2

Table 27. AES algorithm SECMSG key associated data (continued)

Offset (Dec)	Length of Field (Bytes)	Description
45	2	Key-usage field 1 High-order byte: Secure message encryption enablement: **Value** **Meaning** **X'00'** Enable the encryption of PINs in an EMV secure message (SMPIN). All other values are reserved and undefined. Low-order byte: **xxxx 1xxx** The key can only be used in UDXs (used in KGN, KIM, KEX). **xxxx 0xxx** The key can be used in both UDXs and CCA. **xxxx x*uuu*** Reserved for UDXs, where *uuu* are UDX-defined bits. All unused bits are reserved and must be zero.
47	2	Key-usage field 2: Indicates the key usage. High-order byte: Service restriction: **Value** **Meaning** **X'00'** Any verb can use this key (ANY-USE). **X'01'** Only CSNBDPC can use this key (DPC-ONLY). All other values are reserved and undefined. Low-order byte (reserved). All unused bits are reserved and must be zero

Appendix G. Access Control Points and Callable Services

The following access control points are new.

The following tables list usage information using the following abbreviations:

AE Always enabled, cannot be disabled.

ED Enabled by default.

DD Disabled by default.

SC Usage of this access control point requires special consideration.

Table 28. Access control points – Callable Services

Name	Callable service	Usage
Diversified Key Generate2 - Allow length option with KDFFM-DK	CSNBDKG2 / CSNEDKG2	DD
Diversified Key Generate2 – KDFFM-DK	CSNBDKG2 / CSNEDKG2	ED

Table 28. Access control points – Callable Services (continued)

Name	Callable service	Usage
Diversified Key Generate2 - MK-OPTC	CSNBDKG2 / CSNEDKG2	ED

Chapter 3. Update of z/OS Cryptographic Services ICSF Administrator's Guide, SC14-7506-03, information

This topic contains updates to the document *z/OS Cryptographic Services ICSF Administrator's Guide*, SC14-7506-03, for the DK AES PIN Part 4 support provided by this APAR. Refer to this source document if background information is needed.

Understanding cryptographic keys

Secure messaging keys

These keys are used to encrypt keys and PINs for incorporation into a text block. The text block is then encrypted to preserve the security of the key value. The encrypted text block, normally the value field in a TLV item, can be incorporated into a message sent to an EMV smart card.

Table 29. DES secure messaging keys

DES keys	Callable services
Secure-messaging class (data operation keys) These keys are used to encrypt keys or PINs. The keys are double-length keys. The key usage flags in the control vector determine which services the key may be used with.	
SECMSG	Diversified Key Generate, Secure Messaging for Keys, Secure Messaging for PINs

Table 30. AES secure messaging keys

AES keys	Callable services
Secure-messaging class (data operation keys) These keys are used to encrypt keys or PINs. The keys can be 128, 192, or 256 bits in length.	
SECMSG	DK PIN Change

Managing Cryptographic Keys Using the Key Generator Utility Program

Syntax of the ADD and UPDATE control statements

KEYUSAGE(key-usage-value1[,...,key-usage-value2])

This keyword defines key usage values for the key being generated. The usage values are used to restrict a key to a specific algorithm or usage.

The associated data for variable length tokens is described in Appendix B. of the Application Programmer's Guide. The DES control vector is described in Appendix C. of the Application Programmer's Guide.

The following values have been defined. The usage values are specific to a key type. The values can only be specified for the key type indicated in the tables below.

Note: Any value with a non-alphanumeric character must be enclosed in quotes when specified with the KEYUSAGE keyword. For example:

```
KEYUSAGE( 'CVVKEY-A' )
```

When a pair of keys is generated, one for the local system and the other for a remote system, both keys will be generated with the same key-usage flags when the KEYUSAGE keyword is used.

Table 31. Usage values for key types

Key type	Key algorithm	Key Usage Values
CIPHER	AES	The following values are optional: C-XLATE, V1PYLD and One or both may be specified: DECRYPT, ENCRYPT. **Note:** The key generated when KEYUSAGE is not specified will have only the DECRYPT and ENCRYPT key-usage. This is the default.
DKYGENKY	DES	One of the following must be specified: DKYL0, DKYL1, DKYL2, DKYL3, DKYL4, DKYL5, DKYL6, DKYL7 and One of the following must be specified: DALL, DDATA, DEXP, DIMP, DMAC, DMKEY, DMPIN, DMV, DPVR
DKYGENKY	AES	One of the following must be specified: D-PPROT, D-PCALC, D-PPRW and One of the following values must be specified: DKYL0, DKYL1, DKYL2 and The following values are required: KUF-MBE, DKYUSAGE
DKYGENKY	AES	One of the following must be specified: D-MAC, D-SECMSG and The following value is required: DKYUSAGE and One of the following values must be specified: KUF-MBE, KUF-MBP and One of the following values must be specified: DKYL0, DKYL1, DKYL2
DKYGENKY	AES	The following values is required: D-CIPHER and One of the following values must be specified: DKYL0, DKYL1, DKYL2 and The following value is optional: DKYUSAGE and One of the following values may be specified when DKYUSAGE is specified: KUF-MBE, KUF-MBP (KUP-MBE is the default)

Table 31. Usage values for key types (continued)

Key type	Key algorithm	Key Usage Values
DKYGENKY	AES	One of the following must be specified: D-ALL, D-EXP, D-IMP and One of the following values must be specified: DKYL0, DKYL1, DKYL2
EXPORTER	AES	The following value is optional: V1PYLD
IMPORTER	AES	The following value is optional: V1PYLD
KEYGENKY	DES	One of the following must be specified: UKPT, CLR8-ENC
MAC	DES	One of the following may be specified: ANY-MAC, CVVKEY-A, CVVKEY-B
MACVER	DES	One of the following may be specified: ANY-MAC, CVVKEY-A, CVVKEY-B
MAC	AES	One of the following must be specified: GENERATE, GENONLY, VERIFY and The following value must be specified: CMAC and One of the following is optional: DKPINOP, DKPINAD1, DKPINAD2 **Notes:** • One of either DKPINOP, DKPINAD1, or DKPINAD2 is required for keys to be used with the DK PIN services. • When DKPINOP, DKPINAD1, or DKPINAD2 is specified, GENERATE is not allowed.
PINCALC	AES	Three values must be specified: GENONLY, DKPINOP, and CBC.
PINPROT	AES	One of the following must be specified: ENCRYPT, DECRYPT and One of the following must be specified: DKPINOPP, DKPINOP, DKPINAD1 and The following value must be specified: CBC
PINPRW	AES	One of the following must be specified: GENONLY, VERIFY and The following values must be specified: DKPINOP, CMAC

Notes:

- **Diversified Key Generating Keys:** The key-derivation sequence level specifies the hierarchical level of the DKYGENKY. If the sequence level is non-zero, the DKYGENKY can only generate another DKYGENKY key with the sequence level decremented by one. If the sequence level is zero, the DKYGENKY can only generate the final diversified key (a non-DKYGENKY key) with the key type specified by the usage bits.
- **PINPROT Keys:** When specifying an AES CIPHER as the OUTTYPE for an AES PINPROT key, the key usage values must be ENCRYPT and DKINOPP. The key usage value for the AES CIIPHER key is DECRYPT.

Table 32. Meaning of usage values

Key Usage Value	Key types	Meaning
ANY-MAC	MAC, MACVER	The MAC usage field (control vector offset 0-3) is set to '0000'b. There is no restriction for this key. This is the default value.
C-XLATE	CIPHER	Restricts the key to be used with the cipher text translate2 service only.
CBC	PINCALC, PINPRW	Use the CBC encryption mode.
CLR8-ENC	KEYGENKY	The CLR8-ENC key usage bit (control vector offset 19) is set to '1'b. The key may only be used with the 'CLR8-ENC' rule array keyword for CSNBDKG.
CMAC	MAC, PINPROT	Use the CMAC algorithm.
CVVKEY-A	MAC, MACVER	The MAC usage field (control vector offset 0-3) is set to '0010'b. When this key is used with CSNBCVG or CSNBCVV, it can only be used as the key A parameter. This is is valid with single- and double-length keys.
CVVKEY-B	MAC, MACVER	The MAC usage field (control vector offset 0-3) is set to '0011'b. When this key is used with CSNBCVG or CSNBCVV, it can only be used as the key B parameter. This is valid with single-length keys.
D-ALL	DKYGENKY	All key types may be derived except DKYGENKY keys.
D-CIPHER	DKYGENKY	CIPHER keys may be derived.
D-EXP	DKYGENKY	EXPORTER keys may be derived.
D-IMP	DKYGENKY	IMPORTER keys may be derived.
D-MAC	DKYGENKY	MAC keys may be derived.
D-PCALC	DKYGENKY	PINCALC keys may be derived.
D-PPROT	DKYGENKY	PINPROT keys may be derived.
D-PPRW	DKYGENKY	PINPRW keys may be derived.
D-SECMSG	DKYGENKY	SECMSG keys may be derived.
DALL	DKYGENKY	All key types may be generated except DKYGENKY and KEYGENKY keys. Usage is restricted by an access control point. See Diversified key generate callable service.
DDATA	DKYGENKY	Generate single- and double-length DATA keys
DECRYPT	PINPROT CIPHER	This key can be used to decrypt DK PIN blocks. This key can be used to decrypt data.
DEXP	DKYGENKY	Generate EXPORTER and OKEYXLAT keys
DIMP	DKYGENKY	Generate IMPORTER and IKEYXLAT keys
DKPINAD1	MAC, PINPROT	This key may be used in the DK PIN protection methods to create or verify a pin block to allow the changing of the account number associated with a PIN.
DKPINAD2	MAC	This key may be used in the DK PIN protection methods to create or verify an account change string to allow the changing of the account number associated with a PIN.

Table 32. Meaning of usage values (continued)

Key Usage Value	Key types	Meaning
DKPINOP	MAC, PINCALC, PINPROT, PINPRW	This key may be used in the DK PIN protection methods as a general-purpose key. It may not be used as a special-purpose key.
DKPINOPP	PINPROT	This key is to be used to encrypt a PBF-1 format pin block for the specific purpose of creating a DK PIN mailer.
DKYL0	DKYGENKY	Specifies that this key-generating key can be used to derive the key specified by the Key derivation and Derived key usage controls (AES) or control vector (DES).
DKYL1	DKYGENKY	Specifies that this key-generating key can be used to derive a DKYGENKY with a subtype of DKYL0.
DKYL2	DKYGENKY	Specifies that this key-generating key can be used to derive a DKYGENKY with a subtype of DKYL1.
DKYL3	DKYGENKY	Specifies that this key-generating key can be used to derive a DKYGENKY with a subtype of DKYL2.
DKYL4	DKYGENKY	Specifies that this key-generating key can be used to derive a DKYGENKY with a subtype of DKYL3.
DKYL5	DKYGENKY	Specifies that this key-generating key can be used to derive a DKYGENKY with a subtype of DKYL4.
DKYL6	DKYGENKY	Specifies that this key-generating key can be used to derive a DKYGENKY with a subtype of DKYL5.
DKYL7	DKYGENKY	Specifies that this key-generating key can be used to derive a DKYGENKY with a subtype of DKYL6.
DKYUSAGE	DKYGENKY	Specifies that the DKYUSAGE keyword identifies key usage information for the key to be derived by the DKYGENKY. This value is required when the key type to be derived is MAC, PINCALC, PINPROT, PINPRW, and SECMSG. Not valid for D-ALL, D-CIPHER, D-IMP and D-EXP.
DMAC	DKYGENKY	Generate single- and double-length MAC keys
DMKEY	DKYGENKY	Generate secure messaging keys for encrypting keys
DMPIN	DKYGENKY	Generate secure messaging keys for encrypting PINs
DMV	DKYGENKY	Generate single- and double-length MACVER keys
DPVR	DKYGENKY	Generate PINVER keys
ENCRYPT	PINPROT CIPHER	This key can be used to encrypt DK PIN blocks. This key can be used to encrypt data.
GENERATE	MAC	This key can generate and verify MACs.
GENONLY	MAC, PINCALC, PINPRW	This key can be used to only generate data (MACs, PINs, or PRWs).
KUF-MBE	DKYGENKY	Specifies that the key usage fields of the key to be generated must be equal to the related generated key usage fields of the DKYGENKY generating key. Not valid for D-ALL, D-CIPHER, D-IMP and D-EXP.

Table 32. Meaning of usage values (continued)

Key Usage Value	Key types	Meaning
KUF-MBP	DKYGENKY	Specifies that the key usage fields of the key to be generated must be permitted based on the related generated key usage fields of the DKYGENKY generating key. The key to be derived is not permitted to have a higher level of usage than the related key usage fields permit. The key to be derived is only permitted to have key usage that is less than or equal to the related key usage fields. Not valid for D-ALL, D-CIPHER, D-IMP and D-EXP.
TRANSLAT	CIPHER	Restricts the key to be used with the cipher text translate2 service only.
UKPT	KEYGENKY	The UKPT key usage bit (control vector offset 18) is set to '1'b. The key may only be used in the CSNBPTR and CSNBPVR services.
VERIFY	MAC, PINPRW	This key can be used to verify data (MACs or PRWs).
V1PYLD	CIPHER, EXPORTER, IMPORTER	The generated key or keys will have version 1 (fixed-length) format of the payload for the variable-length symmetric key token. Applies to AES keys only.

Notes:

- **Diversified Key Generating Key Note:** The subtype field specifies the hierarchical level of the DKYGENKY. If the subtype is non-zero, then the DKYGENKY can only generate another DKYGENKY key with the hierarchy level decremented by one. If the subtype is zero, the DKYGENKY can only generate the final diversified key (a non-DKYGENKY key) with the key type specified by the usage bits.
- **PINPROT Keys:** When specifying an AES CIPHER as the OUTTYPE for an AES PINPROT key, the key usage values must be ENCRYPT and DKINOPP. The key usage value for the AES CIIPHER key is DECRYPT.
- **AES MAC Keys:** When DKPINOP, DKPINAD1, or DKPINAD2 is specified, GENERATE is not allowed.

DKYGENKYUSAGE(key-usage-value1[,...,key-usage-value2])

This keyword defines key usage values to be supplied for the AES DKYGENKY key being generated. This keyword is required when the DKYUSAGE value is specified in the KEYUSAGE keyword.

The following values have been defined. The usage values are specific to the key type to be derived. The values can only be specified for the key type indicated in Table 33 on page 81 and Table 34 on page 81. The values for the specific key types are detailed in this document in the Key Token Build2 callable service description.

Note: Any value with a non-alphanumeric character must be enclosed in quotes when specified with the DKYGENKYUSAGE keyword. For example: DKYGENKYUSAGE('CVVKEY-A').

Table 33. Values by type for DKYGENKYUSAGE

Type of key to be derived	DKYGENKYUSAGE values
CIPHER	The following values are optional: C-XLATE, DECRYPT, ENCRYPT **Note:** The key generated when DKYGENKYUSAGE is not specified will have DECRYPT and ENCRYPT key-usage. This is the default.
MAC	One of the following values is required: GENERATE, GENONLY, VERIFY and The following value is required: CMAC and One of the following values is optional: DKPINAD1, DKPINAD2, DKPINOP **Notes:** • One of DKPINOP, DKPINAD1, or DKPINAD2 is required for keys to be used with the DK PIN services. • When DKPINOP, DKPINAD1, or DKPINAD2 is specified, GENERATE is not allowed.
PINCALC	The following values are required: GENONLY, CBC, DKPINOP.
PINPROT	One of the following values is required: DECRYPT, ENCRYPT and The following value is required: CBC and One of the following values is required: DKPINAD1, DKPINOP, DKPINOPP
PINPRW	One of the following values is required: GENONLY, VERIFY and The following values are required: CMAC, DKPINOP
SECMSG	The following value is required: SMPIN and One of the following values is required: ANY-USE, DPC-ONLY

Table 34. Meaning of usage values

Value	Key types	Description
ANY-USE	SECMSG	The use of the key in a callable service is not restricted.
CBC	PINPROT, PINCALC	The derived key must use the CBC encryption mode.
CMAC	MAC, PINPRW	The derived key must use the CMAC algorithm.
C-XLATE	CIPHER	Restricts the key to be used with the cipher text translate2 service only.
DPC-ONLY	SECMSG	The use of the key is restricted to the DK PIN Change service.
DECRYPT	CIPHER, PINPROT	The derived key may be used to decrypt PIN blocks.

Table 34. Meaning of usage values (continued)

Value	Key types	Description
DKPINAD1	MAC, PINPROT	The derived key may be used to create or verify a pin block to allow changing the account number associate with a PIN for the DK PIN methods.
DKPINAD2	MAC	The derived key may be used to create or verify an account change string to allow changing the account number associated with a PIN for the DK PIN methods.
DKPINOP	MAC, PINCALC, PINPROT, PINPRW	The derived key may be used as a general purpose key for the DK PIN methods.
DKPINOPP	PINPROT	The derived key may be used to encrypt a PIN block for the specific purpose of creating a PIN mailer for the DK PIN methods.
ENCRYPT	CIPHER, PINPROT	The derived key may be used to encrypt PIN blocks.
GENERATE	MAC	The derived key may be used to generate and verify MACs.
GENONLY	MAC, PINCALC	The derived key may be used to generate MACs or PINs.
SMPIN	SECMSG	Enable the encryption of PINs in an EMV secure message.
VERIFY	MAC	The derived key may be used to verify MACs.

Printed in USA