

Edit Function APIs (V5R2)

Table of Contents

- [Edit Function APIs](#)
 - [Convert Edit Code](#) (QECCVTEC)
 - [Convert Edit Word](#) (QECCVTEW)
 - [Edit](#) (QECEDT)

Edit Function APIs

The Edit Function APIs create and use edit masks.

An **edit mask** is a byte string that tells the edit machine instruction or the Edit (QECEDT) API how to format a numeric value into a readable character string. An edit mask can format a numeric value so that languages that cannot use machine instructions directly can take advantage of this function. The edit mask is defined by the Edit Code (EDTCDE) and Edit Word (EDTWRD) keywords in DDS.

An **edit code** is a standard description of how a number should be formatted. There are many standard edit codes defined by the system. Users can define several edit codes the way they want with the use of the Create Edit Description (CRTEDTD) command.

An **edit word** is a user-defined description of how a number should be formatted. An edit word usually is used when one of the standard edit codes or user-defined edit codes is not sufficient for a particular situation.

For more information on these terms, see:

- [DDS Reference: Concepts](#)
- [DDS Reference: Physical and Logical Files](#)
- [DDS Reference: Display Files](#)
- [DDS Reference: Printer Files](#)

The Edit Function APIs are:

- [Convert Edit Code](#) (QECCVTEC) translates an edit code specification into an edit mask, which is a byte string used to format a numeric value into a readable character string.
- [Convert Edit Word](#) (QECCVTEW) translates an edit word specification into an edit mask. This is useful when one of the standard or user-defined edit codes does not provide the editing required.
- [Edit](#) (QECEDT) uses an edit mask to transform a numeric from its internal format to a character form suitable for displaying.

Convert Edit Code (QECCVTEC) API

Required Parameter Group:

1	Edit mask	Output	Char(256)
2	Edit mask length	Output	Binary(4)
3	Receiver variable length	Output	Binary(4)
4	Zero balance fill character	Output	Char(1)
5	Edit code	Input	Char(1)
6	Fill or Floating currency indication	Input	Char(1)
7	Source variable precision	Input	Binary(4)
8	Source variable decimal positions	Input	Binary(4)
9	Error code	I/O	Char(*)

Default Public Authority: *USE

Threadsafe: No

The Convert Edit Code (QECCVTEC) API translates an edit code specification into an edit mask, which is a byte string used to format a numeric value into a readable character string.

Authorities and Locks

User-Defined Edit Code Authority

*USE

User-Defined Edit Code Library Authority

*EXECUTE

Required Parameter Group

Edit mask

OUTPUT; CHAR(256)

Returns the edit mask generated by this call. The actual length of the edit mask is returned in the edit mask length parameter. The area beyond the actual length of the edit mask is filled with hexadecimal zeros.

The value returned to this parameter should be passed to the Edit (QECEDT) API or the edit machine instruction.

Edit mask length

OUTPUT; BINARY(4)

The actual length of the edit mask.

The value returned in this parameter should be passed to the QECEDT API or used to substring the value returned in the edit mask in the edit machine instruction.

Receiver variable length

OUTPUT; BINARY(4)

Returns the length of the output that is produced by the returned edit mask when it is used.

The value returned in this parameter should be passed to the QECEDT API or used to substring the receiver variable in the edit machine instruction.

Zero balance fill character

OUTPUT; CHAR(1)

Indicates how to perform the edit so that zero balance suppression is done correctly for those edit codes that have zero balance suppression.

The value returned in this parameter should be passed to the QECEDT API or used to determine whether zero suppression requires special handling before issuing the edit machine instruction.

Edit code

INPUT; CHAR(1)

The edit code that is to be translated into an edit mask. The valid values are:

A-D
J-Q
W
Y-Z
1-9

For more information on edit codes, see the discussion of the EDTCDE keyword in the [DDS Reference: Display Files](#).

Fill or floating currency indication

INPUT; CHAR(1)

Indicates how the output should be padded on the left. This parameter should be specified as follows:

" " **Blank fill:** All suppressed zeros are replaced with blanks.

"*" **Asterisk fill:** All suppressed zeros are replaced with asterisks.

Character **Blank fill:** The specified character is used as a floating currency symbol and placed to the left of the first nonsuppressed digit. Characters are X'41' to X'FE'.

Note: You can optionally specify asterisk fill or floating currency symbol with edit codes 1 through 4, A through D, and J through Q.

Source variable precision

INPUT; BINARY(4)

The precision of the numeric variable that is edited with the edit mask. Precision is the displayed length of a field, not including the decimal point. The valid ranges depend on the value specified for the edit code.

Edit Code **Range**

<i>W</i>	5-8
<i>Y</i>	3-8
<i>All others</i>	1-31

The precision of the numeric variable depends on its class:

Variable Class	Precision
<i>Packed</i>	The precision for which the variable was declared. For example, PACKED(8,4) has precision 8.
<i>Zoned</i>	The precision for which the variable was declared. For example, ZONED(8,4) has precision 8.
<i>Binary(2)</i>	5
<i>Binary(4)</i>	10

Notes:

1. Some high-level languages limit the maximum precision of packed and zoned numeric variables.
2. Because the precision of the source variable is so important in creating the edit mask, an edit mask can only be used to edit variables of the exact precision.

Source variable decimal positions

INPUT; BINARY(4)

The number of digits that the source variable precision parameter has placed after the decimal point in the edited output. The value must be less than or equal to source variable precision, but greater than 0. The normal value depends on the class of the source variable precision parameter:

Variable Class	Decimal Position
<i>Packed</i>	The number of decimal positions for which the variable was declared. For example, PACKED (8,4) has 4 decimal positions.
<i>Zoned</i>	The number of decimal positions for which the variable was declared. For example, ZONED (8,4) has 4 decimal positions.
<i>Binary(2)</i>	0
<i>Binary(4)</i>	0

Error code

I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see [Error Code Parameter](#).

Error Messages

Message ID	Error Message Text
-------------------	---------------------------

CPF2620 E	Field longer than integer or fraction mask.
CPF2639 E	Edit mask too large.
CPF27B2 E	Edit code not valid.
CPF27B3 E	Fill/floating currency indication not valid.
CPF27B4 E	Source variable precision not valid.
CPF27B5 E	Source decimal position not valid.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API Introduced: V2R2

[Top](#) | [Edit Function APIs](#) | [APIs by category](#)

Convert Edit Word (QECCVTEW) API

Required Parameter Group:

1	Edit mask	Output	Char(256)
2	Edit mask length	Output	Binary(4)
3	Receiver variable length	Output	Binary(4)
4	Edit word	Input	Char(*)
5	Edit word length	Input	Binary(4)
6	Error code	I/O	Char(*)

Optional Parameter Group:

7	Source length	Input	Binary(4)
8	Currency symbol	Input	Char(1)

Default Public Authority: *USE

Threadsafe: Yes

The Convert Edit Word (QECCVTEW) API translates an edit word specification into an edit mask. This is useful when one of the standard or user-defined edit codes does not provide the editing required.

Authorities and Locks

None.

Required Parameter Group

Edit mask

OUTPUT; CHAR(256)

Returns the edit mask generated by this call. The actual length of the edit mask is returned in the edit mask length parameter. The area beyond the actual length of the edit mask is filled with hexadecimal zeros.

The value returned to this parameter should be passed to the Edit (QECEDT) API or the edit machine instruction.

Edit mask length

OUTPUT; BINARY(4)

Returns the actual length of the edit mask parameter.

The value returned in this parameter should be passed to the QECEDT API or used to substring the value returned in the edit mask in the edit machine instruction.

Receiver variable length

OUTPUT; BINARY(4)

The actual length of the output that is produced by the returned edit mask when it is used.

The value returned in this parameter should be passed to the QECEDT API or used to substring the value returned in the receiver variable in the edit machine instruction.

Edit word

INPUT; CHAR(*)

The edit word is translated into an edit mask. If the optional currency symbol parameter is not used, the character in the system value QCURSYM is treated as a currency symbol if it appears in the edit word.

Edit word length

INPUT; BINARY(4)

The actual length of the edit word. The value passed must be from 1 through 256.

Error code

I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see [Error Code Parameter](#).

Optional Parameter Group

Source length

INPUT; BINARY(4)

The length of the source data that is to be edited. The length used is the same as the value that would be used for the source variable precision parameter of the QECEDT API.

An edit word may begin with a zero suppression character. If it does, two different edit word combinations can be provided:

- The zero suppression character and the remaining number of replacement characters equals the source length. This provides suppression of the leading zero, only.
- The zero suppression character plus the replacement characters is one more than the source length. This results in no zero suppression.

If the edit word provided is the second form, use this parameter. Otherwise, an incorrect edit mask is generated because QECCVTEW assumes the first form of editing is intended.

Currency symbol

INPUT; CHAR(1)

The currency symbol to be placed in the edited output. If this value appears in the edit word, it is used as the currency symbol.

If the value passed for currency symbol is X'00' (HEX zero), the API uses the system value QCURSYM as the currency symbol.

Error Messages

Message ID	Error Message Text
CPF2620 E	Field longer than integer or fraction mask.
CPF2639 E	Edit mask too large.
CPF265E E	Number of parameters specified not valid.
CPF27B6 E	Edit word length not valid.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.

API Introduced: V2R2

[Top](#) | [Edit Function APIs](#) | [APIs by category](#)

Edit (QECEDT) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Receiver variable length	Input	Binary(4)
3	Source variable	Input	*
4	Source variable class	Input	Char(10)
5	Source variable precision	Input	Binary(4)
6	Edit mask	Input	Char(*)
7	Edit mask length	Input	Binary(4)
8	Zero balance fill character	Input	Char(1)
9	Error code	I/O	Char(*)

Default Public Authority: *USE

Threadsafe: Yes

The Edit (QECEDT) API uses an edit mask to transform a numeric from its internal format to a character form suitable for displaying.

Authorities and Locks

None.

Required Parameter Group

Receiver variable

OUTPUT; CHAR(*)

Receives the edited output. The length of this area must be passed in the receiver variable length parameter.

Receiver variable length

INPUT; BINARY(4)

The length of the referenced area by the receiver variable parameter. This value must be greater than 0.

This value was returned in the receiver variable length parameter on the previous call to the Convert Edit Code (QECCVTEC) API or Convert Edit Word (QECCVTEW) API; otherwise, CPF27AF is returned.

Source variable

INPUT; *

The numeric value that is converted. The type is defined by the source variable class parameter and the length is specified in the source variable precision parameter.

Source variable class

INPUT; CHAR(10)

The type of numeric variable passed in the source variable parameter.

The types are:

*BINARY
*PACKED
*ZONED

Source variable precision

INPUT; BINARY(4)

The precision of the numeric variable specified in the source variable parameter.

The value passed must be from 1 through 31.

Variable

Class	Precision
--------------	------------------

<i>Packed</i>	The precision for which the variable was declared. For example, PACKED(8,4) has precision 8.
---------------	--

<i>Zoned</i>	The precision for which the variable was declared. For example, ZONED(8,4) has precision 8.
--------------	---

<i>Binary(2)</i>	5
------------------	---

<i>Binary(4)</i>	10
------------------	----

Note: Some high-level languages limit the maximum precision of packed and zoned numeric variables.

Edit mask

INPUT; CHAR(*)

The edit mask used for this edit operation. This is the value returned in the edit mask parameter on the call to the QECCVTEC API or QECCVTEW API.

Edit mask length

INPUT; BINARY(4)

The length of the edit mask. The value passed must be from 1 through 256. This is the value returned in the edit mask length parameter on the call to the QECCVTEC API or QECCVTEW API.

Zero balance fill character

INPUT; CHAR(1)

Indicates how to perform the edit operation so that zero balance suppression is done correctly for those edit codes that have zero balance suppression.

If the QECCVTEC API is used to create the edit mask, this should be the value returned in the zero balance fill character parameter; otherwise, unpredictable results may occur.

If the QECCVTEW API is used to create the edit mask, X'00' should be specified for this parameter;

otherwise, unpredictable results may occur.

Error code

I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see [Error Code Parameter](#).

Error Messages

Message ID	Error Message Text
-------------------	---------------------------

CPF27AB E	Source variable class not valid.
-----------	----------------------------------

CPF27AF E	Edit mask not valid.
-----------	----------------------

CPF27B4 E	Source variable precision not valid.
-----------	--------------------------------------

CPF27B7 E	Receiver variable length not valid.
-----------	-------------------------------------

CPF27B8 E	Edit mask length not valid.
-----------	-----------------------------

CPF3C90 E	Literal value cannot be changed.
-----------	----------------------------------

CPF3CF1 E	Error code parameter not valid.
-----------	---------------------------------

API Introduced: V2R2

[Top](#) | [Edit Function APIs](#) | [APIs by category](#)