



System i  
Programming  
Edit Function APIs

*Version 6 Release 1*







System i  
Programming  
Edit Function APIs

*Version 6 Release 1*

**Note**

Before using this information and the product it supports, read the information in "Notices," on page 9.

This edition applies to version 6, release 1, modification 0 of IBM i5/OS (product number 5761-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CISC models.

© Copyright International Business Machines Corporation 1998, 2008. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Edit Function APIs . . . . .</b>	<b>1</b>
APIs . . . . .	1
Convert Edit Code (QECCVTEC) API . . . . .	1
Authorities and Locks . . . . .	2
Required Parameter Group . . . . .	2
Error Messages . . . . .	4
Convert Edit Word (QECCVTEW) API. . . . .	4
Authorities and Locks . . . . .	4
Required Parameter Group . . . . .	4
Optional Parameter Group. . . . .	5
Error Messages . . . . .	6

Edit (QECEDT) API . . . . .	6
Authorities and Locks . . . . .	6
Required Parameter Group . . . . .	6
Error Messages . . . . .	8

<b>Appendix. Notices . . . . .</b>	<b>9</b>
Programming interface information . . . . .	10
Trademarks . . . . .	11
Terms and conditions . . . . .	12



---

## 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 about these terms, see the Data description specifications (DDS) topic collections.

The edit function APIs are:

- “Convert Edit Code (QECCVTEC) API” (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) API” on page 4 (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) API” on page 6 (QECEDT) uses an edit mask to transform a numeric from its internal format to a character form suitable for displaying.

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

---

## APIs

These are the APIs for this category.

---

### 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 for display files topic collection.

### 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:

" "	<b>Blank fill:</b> All suppressed zeros are replaced with blanks.
"*"	<b>Asterisk fill:</b> All suppressed zeros are replaced with asterisks.
<i>Character</i>	<b>Blank fill:</b> 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
W	5-8
Y	3-8
<i>All others</i>	1- 63

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,” on page 1](#) | [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,” on page 1](#) | [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  $\gg 63 \ll$ .

**Variable**

<b>Class</b>	<b>Precision</b>
<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,”](#) on page 1 | [APIs by category](#)

---

## Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106-0032, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
Software Interoperability Coordinator, Department YBWA  
3605 Highway 52 N  
Rochester, MN 55901  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. \_enter the year or years\_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

---

## Programming interface information

This API descriptions publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i5/OS.

---

## Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Advanced 36  
Advanced Function Presentation  
Advanced Peer-to-Peer Networking  
AFP  
AIX  
AnyNet  
AS/400  
BCOCA  
C/400  
COBOL/400  
Common User Access  
CUA  
DB2  
DB2 Universal Database  
Distributed Relational Database Architecture  
Domino  
DPI  
DRDA  
Enterprise Storage Server  
eServer  
FlashCopy  
GDDM  
i5/OS  
IBM  
IBM (logo)  
InfoColor  
Infoprint  
Integrated Language Environment  
Intelligent Printer Data Stream  
IPDS  
Lotus  
Lotus Notes  
MO:DCA  
MVS  
Net.Data  
NetServer  
Notes  
OfficeVision  
Operating System/2  
Operating System/400  
OS/2  
OS/400  
PartnerWorld  
POWER5+  
PowerPC  
Print Services Facility  
PrintManager  
PROFS  
RISC System/6000  
RPG/400  
RS/6000

SAA  
SecureWay  
SOM  
System i  
System i5  
System Object Model  
System/36  
System/38  
System/390  
TotalStorage  
VisualAge  
WebSphere  
xSeries  
z/OS

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

---

## Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

**Personal Use:** You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

**Commercial Use:** You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER

EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.







Printed in USA