



@server

iSeries

Edit Function APIs

Version 5 Release 3





@server

iSeries

Edit Function APIs

Version 5 Release 3

Note

Before using this information and the product it supports, be sure to read the information in "Notices," on page 11.

Sixth Edition (August 2005)

This edition applies to version 5, release 3, modification 0 of Operating System/400 (product number 5722-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, 2005. All rights reserved.

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

Contents

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

Edit (QECEDT) API	7
Authorities and Locks	7
Required Parameter Group	7
Error Messages	9

Appendix. Notices 11

Trademarks	12
Terms and conditions for downloading and printing publications	13
Code disclaimer information.	14

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) API” on page 2 (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 5 (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 7 (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 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
W	5-8
Y	3-8
All others	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

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.

Message ID	Error Message Text
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 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,”](#) 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 information 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.

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.

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

Trademarks

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

Advanced 36
Advanced Function Printing
Advanced Peer-to-Peer Networking
AFP
AIX
AS/400
COBOL/400
CUA
DB2
DB2 Universal Database
Distributed Relational Database Architecture
Domino
DPI

DRDA
eServer
GDDM
IBM
Integrated Language Environment
Intelligent Printer Data Stream
IPDS
iSeries
Lotus Notes
MVS
Netfinity
Net.Data
NetView
Notes
OfficeVision
Operating System/2
Operating System/400
OS/2
OS/400
PartnerWorld
PowerPC
PrintManager
Print Services Facility
RISC System/6000
RPG/400
RS/6000
SAA
SecureWay
System/36
System/370
System/38
System/390
VisualAge
WebSphere
xSeries

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, and service names may be trademarks or service marks of others.

Terms and conditions for downloading and printing publications

Permissions for the use of the information you have selected for download are granted subject to the following terms and conditions and your indication of acceptance thereof.

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

Commercial Use: You may reproduce, distribute and display this information solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of this information, or reproduce, distribute or display this information 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 information or any 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 information 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 THIS INFORMATION. THE INFORMATION IS 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.

All material copyrighted by IBM Corporation.

By downloading or printing information from this site, you have indicated your agreement with these terms and conditions.

Code disclaimer information

This document contains programming examples.

SUBJECT TO ANY STATUTORY WARRANTIES WHICH CANNOT BE EXCLUDED, IBM^(R), ITS PROGRAM DEVELOPERS AND SUPPLIERS MAKE NO WARRANTIES OR CONDITIONS EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, REGARDING THE PROGRAM OR TECHNICAL SUPPORT, IF ANY.

UNDER NO CIRCUMSTANCES IS IBM, ITS PROGRAM DEVELOPERS OR SUPPLIERS LIABLE FOR ANY OF THE FOLLOWING, EVEN IF INFORMED OF THEIR POSSIBILITY:

1. LOSS OF, OR DAMAGE TO, DATA;
2. SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES, OR FOR ANY ECONOMIC CONSEQUENTIAL DAMAGES; OR
3. LOST PROFITS, BUSINESS, REVENUE, GOODWILL, OR ANTICIPATED SAVINGS.

SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.



Printed in USA