



System i
Programming
Workstation Support APIs

Version 6 Release 1





System i
Programming
Workstation Support APIs

Version 6 Release 1

Note

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

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

Workstation Support APIs	1
APIs	1
Query Keyboard Buffering (QWSQRYWS)API	1
Authorities and Locks	2
Required Parameter Group	2
Optional Parameter	2
Error Messages	2
Retrieve Output Information (QWSRTVOI) API.	2
Authorities and Locks	3
Required Parameter Group	3
OINF0100 Format.	4
OINF0200 Format.	4
Field Descriptions	5
Error Messages	6
Set Keyboard Buffering (QWSSETWS) API	6

Authorities and Locks	7
Required Parameter Group	7
Optional Parameter	7
Error Messages	7
Suspend or Restore Display File (QWSSPRST) API.	8
Authorities and Locks	8
Required Parameter Group	8
Error Messages	9

Appendix. Notices	11
Programming interface information	12
Trademarks	13
Terms and conditions	14

Workstation Support APIs

The workstation support APIs allow you to control the type-ahead characteristics of a workstation and to retrieve information about the last output operation to the requester device for the specified interactive job.

Type-ahead, also called **keyboard buffering**, lets the user type data faster than it can be sent to the system. **Attention key buffering** determines how to process the action of pressing an attention key. If attention key buffering is on, the attention key is treated as any other key. If attention key buffering is not on, pressing the attention key results in sending the information to the system even when other workstation input is inhibited.

You can enter the parameters for the QWSQRYWS and QWSSETWS APIs in mixed case. The APIs convert them to uppercase. For all other APIs, you must enter all parameters in uppercase.

The keyboard buffering data stream is supported by the following controllers:

- ASCII Workstation Input/Output Processor
- Twinaxial Workstation Input/Output Processor
- 5394 Remote Control Unit
- 5494 Control Unit

IBM® Personal Computer Systems attached by System i™ Access or workstation emulation (WSE) do not support the type-ahead data stream. Keyboard buffering for these devices is controlled through the emulation programs.

The workstation support APIs are:

- “Query Keyboard Buffering (QWSQRYWS)API” (QWSQRYWS) determines the current type-ahead and attention key buffering settings.
- “Retrieve Output Information (QWSRTVOI) API” on page 2 (QWSRTVOI) gives the caller information on the last attempted output operation to the requester device for the specified job.
- “Set Keyboard Buffering (QWSSETWS) API” on page 6 (QWSSETWS) controls the use of the type-ahead and attention key buffering functions.
- “Suspend or Restore Display File (QWSSPRST) API” on page 8 (QWSSPRST) gives the caller the ability to either suspend the active display file on the requester device or restore a suspended file to the requester device.

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

APIs

These are the APIs for this category.

Query Keyboard Buffering (QWSQRYWS)API

Required Parameter Group:

1	Keyboard buffering	Output	Char(1)
---	--------------------	--------	---------

Optional Parameter Group:

2 Device name Input Char(10)

Default Public Authority: *USE
Threadsafe: No

The Query Keyboard Buffering (QWSQRYWS) API sends the data stream command to query the current value for keyboard buffering for a specified display.

Authorities and Locks

Device Authority
*USE

Display File QDWSTYPA Authority
*USE

Required Parameter Group

Keyboard buffering
OUTPUT; CHAR(1)

The current setting for keyboard buffering for the device. Valid values are:

- 0 The type-ahead and attention key buffering functions are off.
- 1 The type-ahead and attention key buffering functions are on.
- 2 The type-ahead function is on, and the attention key buffering function is off.

Optional Parameter

Device name
INPUT; CHAR(10)

The display to query for the keyboard buffering value. You can specify a particular device or use this special value:

*REQUESTER The job's requesting program device is queried for the keyboard buffering value. This is the default.

Error Messages

Message ID	Error Message Text
CPF3C90 E	Literal value cannot be changed.
CPF94FC E	Type-ahead data stream not supported by controller.
CPF94FD E	Type-ahead option parameter value not valid.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V2R1

Top | "Workstation Support APIs," on page 1 | APIs by category

Retrieve Output Information (QWSRTVOI) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)

3	Format name	Input	Char(8)
4	Qualified job name	Input	Char(26)
5	Internal job identifier	Input	Char(16)
6	Error code	I/O	Char(*)

Default Public Authority: *USE
 Threadsafe: No

The Retrieve Output Information (QWSRTVOI) API gives the caller information on either the last active record format involved in an output operation to the requester device for the specified job or all the active record formats currently displayed on the requester device for the specified job.

Authorities and Locks

Job Authority

*JOBCTL if the job for which information is retrieved has a different user profile from that of the job that calls the QWSRTVOI API.

Device Authority

*READ authority to the display device that the interactive job is using as the *REQUESTER device.

File Authority

*READ authority to the display file that is active currently.

Required Parameter Group

Receiver variable

OUTPUT; CHAR(*)

The variable that is to receive the information requested. You can specify the size of this area to be smaller than the format requested if you specify the length of receiver variable parameter correctly. As a result, the API returns only the data that the area can hold.

Length of receiver variable

INPUT; BINARY(4)

The length of the receiver variable. If this value is larger than the actual size of the receiver variable, the result may not be predictable. The minimum length is 8 bytes.

Format name

INPUT; CHAR(8)

The content and format of the information returned for the job. The following format names can be specified:

<i>OINF0100</i>	Display file and last record format name involved in an output operation.
<i>OINF0200</i>	Display file and a list of active records currently on the display.

For more information, see "OINF0100 Format" on page 4 or "OINF0200 Format" on page 4.

Qualified job name

INPUT; CHAR(26)

The name of the job for which information is to be returned. The qualified job name has three parts:

Job name CHAR(10). A specific job name or one of the following special values:

- * The job in which this program is running. The rest of the qualified job name parameter must be blank.
- *INT The internal job identifier locates the job. The user name and job number must be blank.

User name CHAR(10). A specific user profile name, or blanks when the job name is a special value or *INT.

Job number CHAR(6). A specific job number, or blanks when the job name specified is a special value or *INT.

Internal job identifier

INPUT; CHAR(16)

The internal identifier for the job. The List Job (QUSLJOB) API creates this identifier. If you do not specify *INT for the job name parameter, this parameter must contain blanks. With this parameter, the system can locate the job more quickly than with a job name.

Error code

I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

OINF0100 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(10)	Display file name
18	12	CHAR(10)	Display file library name
28	1C	CHAR(10)	Display file record format name

OINF0200 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Offset to the list of active record formats
12	C	BINARY(4)	Number of active record formats
16	10	BINARY(4)	Size of an active record format entry
20	14	CHAR(10)	Display file name
30	1E	CHAR(10)	Display file library name
These fields repeat in the order listed for the number of active record formats		CHAR(10)	Active record format name
		CHAR(10)	Record format type
		BINARY(4)	Start row
		BINARY(4)	End row
		BINARY(4)	Window start column
		BINARY(4)	Window end column

Field Descriptions

Active record format name. The name of an active record format for the active file on the requester device for the specified interactive job. An active record format is a record format that is currently displayed on the requester device.

The following internal record names may be returned:

<i>*ERRSFL</i>	The name for an internal sub-file created when the ERRSFL keyword is used at the file level in the DDS. The start and end row values both indicate the message line. If the MSGLOC keyword is used in the DDS, the values indicate that line. Otherwise, they indicate the default message line.
<i>*ERRCTL</i>	The name of an internal sub-file control record created for the ERRSFL keyword.

Bytes available. The number of bytes of data available to be returned. All available data is returned if enough space is provided.

Bytes returned. The number of bytes of data returned.

Display file library name. The name of the library that contains the display file.

Display file name. The name of the display file. The user interface manager (UIM) panel groups generate a user-defined data stream through a display file. The name of the display file (such as QDUI80 or QDUI132) is returned, not the panel group. If format OINF0200 is being used, the record format type for a UIM panel is **USRDFN*.

Display file record format name. The name of the record format used from the display file.

End row. Ending row of the record format.

Number of active record formats. The number of currently active record formats for the given file. This number is used to determine the size of the list of active record formats array. The maximum number of active records is 25.

Offset to the list of active record formats. An offset from the beginning of format OINF0200 to the list of active record formats.

Record format type. The type of the record format. Blanks are returned in this field if the record format has no special type. Otherwise, one of the following is returned:

<i>*SFL</i>	Subfile record
<i>*SFLCTL</i>	Subfile control record
<i>*SFLMSGRC</i>	Subfile message record
<i>*MNUBAR</i>	Menu bar record
<i>*PULLDOWN</i>	Pull down record
<i>*USRDFN</i>	The record is a user defined data stream.
<i>*WINDOW</i>	Window definition record.

If a window is active on the display, any previously written records are inactive. The window definition record is active, as are any records written "in" the window. Non-window definition records active in the window are records with the WINDOW keyword with a parameter that names the window definition record format; these are window reference records. Reported for an active window are:

- An entry that names the window definition record and for which the the start row, end row, start column and end column define the positions of the window's borders on the display. This is the only entry which has a format type of "*WINDOW".
- Another entry with the same record name, if the window definition record contains fields. Only the start row and end row are non-zero in this entry. The start row and end row are the actual rows on the display that the record's fields occupy.
- Any additional records that are active in the window, with their actual start row and end row positions.

If a window is active, the sub-file for the ERRSFL keyword (if present in the DDS) is not active.

Size of an active record format entry. The size of one of the repeating entries in the active records format information.

Start row. Starting row of the record format.

Window end column. If the record format type is *WINDOW, this is the ending column of the window record. For all other record format types, this field is zero.

Window start column. If the record format type is *WINDOW, this is the starting column of the window record. For all other record format types, this field is zero.

Error Messages

Message ID	Error Message Text
CPF2207 E	Not authorized to requester device or display file.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3C51 E	Internal job identifier not valid.
CPF3C52 E	Internal job identifier no longer valid.
CPF3C53 E	Job &3/&2/&1 not found.
CPF3C55 E	Job &3/&2/&1 does not exist.
CPF3C57 E	Not authorized to retrieve job information.
CPF3C58 E	Job name specified is not valid.
CPF3C59 E	Internal identifier is not blanks and job name is not *INT.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF919F E	Job is not interactive.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V3R6

Top | "Workstation Support APIs," on page 1 | APIs by category

Set Keyboard Buffering (QWSSETWS) API

Required Parameter Group:

1	Keyboard buffering	Input	Char(1)
---	--------------------	-------	---------

Optional Parameter Group:

2	Device name	Input	Char(10)
---	-------------	-------	----------

Default Public Authority: *USE
Threadsafe: No

The Set Keyboard Buffering (QWSSETWS) API controls the type-ahead and attention key buffering functions for a display. With the QWSSETWS API, you can:

- Turn both functions off
- Turn both functions on
- Turn on the type-ahead function without buffering the Attention key
- Send the data stream to a specific device

Any changes to the keyboard buffering value made through this program take effect immediately.

The keyboard buffering data stream is supported by the ASCII Workstation Input/Output Processor, Twinaxial Workstation Input/Output Processor, 5394 Remote Control Unit, and 5494 Control Unit. Keyboard buffering for personal computer systems attached by System i™ Access or workstation emulation (WSE) is controlled through the emulation programs; these devices do not support the type-ahead data stream.

Authorities and Locks

Device Authority
*USE

Display File QDWSTYPA Authority
*USE

Required Parameter Group

Keyboard buffering
INPUT; CHAR(1)

The setting for keyboard buffering for a display. Valid values are:

- 0 The type-ahead and attention key buffering functions are off.
- 1 The type-ahead and attention key buffering functions are on.
- 2 The type-ahead function is on, and the attention key buffering function is off.

Optional Parameter

Device name
INPUT; CHAR(10)

The device to set the keyboard buffering value on. You can specify the name of a particular device or use this special value:

*REQUESTER The keyboard buffering value is set on the job's requesting program device. This is the default.

Error Messages

Message ID	Error Message Text
CPF3C90 E	Literal value cannot be changed.
CPF94FC E	Type-ahead data stream not supported by controller.
CPF94FD E	Type-ahead option parameter value not valid.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

Suspend or Restore Display File (QWSSPRST) API

Required Parameter Group:

1	Function requested	Input	Char(1)
2	Handle to suspended file	I/O	Binary(4)
3	Screen image option	Input	Char(1)
4	Error code	I/O	Char(*)

Default Public Authority: *USE

Threadsafe: No

The Suspend or Restore Display File (QWSSPRST) API gives the caller the ability to either suspend the active display file on the requester device or restore a suspended file to the requester device. The suspend function returns a handle to the suspended file. The restore function accepts a handle of a suspended file to be restored.

The QWSSPRST API also allows the caller to have the screen image saved before the file is suspended. When the file is restored, this screen image also can be restored.

Authorities and Locks

None.

Required Parameter Group

Function requested

INPUT; CHAR(1)

The function requested for the start of QWSSPRST. The following functions can be specified:

- 0 Suspend the active file on the requester device and return a corresponding handle to that file.
- 1 Restore a suspended file to the requester device. The handle indicates which file will be restored.

Handle to suspended file

I/O; BINARY(4)

When suspending the active file (function 0), a handle to the suspended file will be returned in this parameter. When restoring a suspended file (function 1), this parameter accepts the handle of the file to be restored.

Screen image option

INPUT; CHAR(1)

When suspending the active file (function 0), this parameter gives the caller the option to save the current screen image. When restoring a suspended file (function 1), this parameter gives the caller the option to restore the saved screen image associated with the suspended file. If there is no saved screen image associated with the suspended file, this parameter is ignored when doing a restore.

If the screen image is not saved and restored when the file is restored, it is assumed that the application that is using the file will redraw the screen image. The following options can be specified:

- 0 If suspending the active file (function 0), do not save the screen image. If restoring an active file (function 1), do not restore the screen image associated with the suspended file.
- 1 If suspending the active file (function 0), save the current screen image. If restoring an active file (function 1), restore the screen image associated with the suspended file.

This parameter provides a function similar to that of the RSTDSP parameter of the CRTDSPF and CHGDSPF commands. Be aware that the command parameter causes the system to save and restore the screen image if RSTDSP(*YES) is used. The user of this API must request save the screen image at suspend time and restore the screen image at restore time to have the same function.

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
CPF24B4 E	Severe error while addressing parameter list.
CPF3CF1 E	Error code parameter not valid.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF3C90 E	Literal value cannot be changed.
CPF4738 E	Parameter value not valid.
CPF5082 E	Suspend request failed.
CPF5083 E	Restore request failed.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V4R5

[Top](#) | [“Workstation Support 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