System i

Files and file systems
Spooled files

Version 5 Release 4

System i

Files and file systems
Spooled files

Version 5 Release 4

Note

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

Third Edition (February 2006)

This edition applies to version 5, release 4, modification 0 of i5/OS (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.

Contents

Spooled files 1
Printable PDF 1
Output spooling 2
Output queues and spooled files. 3
Default server output queues 3
Spooling writers 4
Summary of spooling writer commands 4
Input spooling 4
Summary of job input commands 6
Using an inline data file 7
Considerations for opening inline data files 8

Appendix. Notices 9
Programming Interface Information 10
Trademarks 11
Terms and conditions 11

Spooled files

Spooling is a system function that saves data in a spooled file for later processing or printing. Spooled files work in a way similar to tape files or other device files. Spooled files can help you manage your data targeted for externally attached devices, such as a printer.

A spooled file holds output data until it can be printed. The spooled file collects data from a device until a program or device is able to process the data. A program uses a spooled file as if it were reading from or writing to an actual device.

Spooling functions can help system users to manage input and output operations more efficiently. The system supports output spooling and input spooling. Output spooling can be used for printer devices. Input spooling applies to database file input.

At the end of a job, the job log can be written to spooled file QPJOBLOG to be printed. After the job log is written to the spooled file, the job log is deleted.

Related concepts

Basic printing: Spooled file

Printable PDF

Use this to view and print a PDF of this information.

To view or download the PDF version of this document, select Spooled files (about 180 KB).

You can view or download these related topics:

- Control language provides a description of the control language (CL) and its commands. Each command is defined including its syntax diagram, parameters, default values, and keywords.
- Basic Printing provides information about how to understand and control printing: printing elements and concepts, printer file support, print spooling support, printer connectivity, advanced function printing, and printing with personal computers

Saving PDF files

To save a PDF on your workstation for viewing or printing:

1. Right-click the PDF in your browser (right-click the link above).
2. Click the option that saves the PDF locally.
3. Navigate to the directory in which you want to save the PDF.
4. Click **Save**.

Downloading Adobe Reader

You need Adobe Reader installed on your system to view or print these PDFs. You can download a free copy from the Adobe Web site (www.adobe.com/products/acrobat/readstep.html)  .

Output spooling

Output spooling can be used for both printer and diskette devices. Output spooling sends job output to disk storage instead of sending it directly to a printer or diskette output device. Output spooling allows the job that produces the output to continue processing without consideration for the speed or availability of output devices.

Additionally, output spooling allows the server to produce output on multiple output devices, such as printer and diskette devices, in an efficient manner. It does this by sending the output of a job destined for a printer to disk storage. This process breaks a potential job limitation imposed by the availability or speed of the output devices.

The main elements of output spooling are:

- **Device description:** A description of the printer device.
- **Spooled file:** A file containing spooled output records that are to be processed on an output device.
- **Output queue:** An ordered list of spooled files.
- **Writer:** A program that sends files from an output queue to a device.
- **Application program:** A high-level language program that creates a spooled file using a device file with the spooling attribute specified as SPOOL(*YES).
- **Device file:** A description of the format of the output, and a list of attributes that describe how the server should process the spooled file.

Output spooling functions are performed by the server without requiring any special operations by the program that produces the output. When a device file is opened by a program, the operating system determines whether the output is to be spooled. When a printer file specifying spooling is opened, the spooled file containing the output of the program is placed on the appropriate output queue in the server.

A spooled file can be made available for printing when the printer file is opened, when the printer file is closed, or at the end of the job. A printer writer is started in the spooling subsystem to send the records to the printer. The spooled file is selected from an output queue.

Spooling device descriptions

Device descriptions must be created for each printer and diskette device in order to define that device to the server. Printer device descriptions are created using the Create Device Description for Printer (CRTDEVPRT) command; diskette device descriptions are created using the Create Device Description for Diskette (CRTDEVDKT) command.

File redirection of spooled files

File redirection occurs when a spooled file is sent to an output device other than the one for which it was originally intended. File redirection can involve devices that process different media (such as printer output sent to a diskette device) or devices that process the same type of media but are of different devices types (such as 5219 Printer output sent to a 4224 Printer).

Depending on the new output device for the spooled file, the file can be processed just as it would have been on the originally specified device. However, differences in devices often cause the output to be formatted differently. In these cases, the server sends an inquiry message to the writer's message queue to inform you of the situation and allow you to specify whether you want printing to continue.

Output queues and spooled files

Batch and interactive job processing can result in spooled output records that are to be processed on an output device, such as a printer or diskette drive. These output records are stored in spooled files until they can be processed. There can be many spooled files for a single job.

When a spooled file is created, the file is placed on an output queue. Each output queue contains an ordered list of spooled files. A job can have spooled files on one or more output queues. All spooled files on a particular output queue should have a common set of output attributes, such as device, form type, and lines per inch. Using common attributes on an output queue reduces the amount of intervention required and increases the device throughput.

The following lists some of the parameters on the Create Output Queue (CRTOUTQ) command and what they specify:

- MAXPAGES: Specifies the maximum spooled file size in pages that is allowed to be printed between a starting and ending time of day.
- AUTOSTRWTR: Specifies the number of writers that are started automatically to this output queue.
- DSPDTA: Whether users without any special authority but who do have *USE authority to the output queue can display, copy, or send the contents of spooled files other than their own. By specifying *OWNER for DSPDTA, only the owner of the file or a user with *SPLCTL special authority can display, copy, or send a file.
- JOBSEP: How many, if any, job separator pages are to be printed between the output of each job when the output is printed.
- DTAQ: The data queue associated with this output queue. If specified, an entry is sent to the data queue whenever a spooled file goes to Ready Status on the queue.
- OPRCTL: Whether a user having job control authority can control the output queue (for example, if the user can hold the output queue).
- SEQ: Controls the order in which spooled files will be sorted on the output queue.
- AUTCHK: Specifies what type of authority to the output queue will enable a user to control the spooled files on the output queue (for example, enable the user to hold the spooled files on the output queue).
- AUT: Public authority. Specifies what control users have over the output queue itself.
- TEXT: Text description. Up to 50 characters of text that describes the output queue.

Default server output queues

The server is shipped with the defaults on commands to use the default output queue for the server printer as the default output queue for all spooled output. The server printer is defined by the QPRTDEV server value.

When a spooled file is created by opening a device file and the output queue specified for the file cannot be found, the server will attempt to place the spooled file on output queue QPRINT in library QGPL. If for any reason the spooled file cannot be placed on output queue QPRINT, an error message will be sent and the output will not be spooled.

The following output queues are supplied with the server:

- **QDKT**: Default diskette output queue
- **QPRINT**: Default printer output queue
- **QPRINTS**: Printer output queue for special forms
- **QPRINT2**: Printer output queue for 2-part paper

Spooling writers

A writer is an i5/OS® program that takes spooled files from an output queue and produces them on an output device. The spooled files that have been placed on a particular output queue will remain stored in the server until a writer is started to the output queue.

The writer takes spooled files one at a time from the output queue, based on their priority. The writer processes a spooled file only if its entry on the output queue indicates that it has a ready (RDY) status. You can display the status of a particular spooled file using the Work with Output Queue (WRKOUTQ) command.

If the spooled file has a ready status, the writer takes the entry from the output queue and prints the specified job or file separators or both, followed by the output data in the file. If the spooled file does not have a ready status, the writer leaves the entry on the output queue and goes on to the next entry. In most cases the writer will continue to process spooled files (preceded by job and file separators) until all files with a ready status have been taken from the output queue.

The AUTOEND parameter on the start writer commands determines whether the writer continues to wait for new spooled files to become available to be written, end after processing one file, or end after all spooled files with ready status have been taken from the output queue.

Summary of spooling writer commands

There are many commands that you can use to control spooling writers.

- Start Diskette Writer (STRDKTWTR): Starts a spooling writer to a specified diskette device to process spooled files on that device.
- Start Printer Writer (STRPRTWTR): Starts a spooling writer to a specified printer device to process spooled files on that device.
- Start Remote Writer (STRRMTWTR): Starts a spooling writer that sends spooled files from an output queue to a remote server.
- Change Writer (CHGWTR): Allows you to change some writer attributes, such as form type, number of file separator pages, or output queue attributes.
- Hold Writer (HLDWTR): Stops a writer at the end of a record, at the end of a spooled file, or at the end of a page.
- Release Writer (RLSWTR): Releases a previously held writer for additional processing.
- End Writer (ENDWTR): Ends a spooling writer and makes the associated output device available to the server.

Note: You can define some functions to provide additional spooling support. Example source and documentation for the commands, files, and programs for these functions are part of library QUSRTOOL, which is an optionally installed part of i5/OS.

Input spooling

Input spooling takes the information from the input device, prepares the job for scheduling, and places an entry in a job queue. Using input spooling, you can typically shorten job run time, increase the number of jobs that can be run sequentially, and improve device throughput.

The main elements of input spooling are:

- **Job queue**: An ordered list of batch jobs submitted to the server for running and from which batch jobs are selected to run.
- **Reader**: A function that takes jobs from an input device or database file and places them on a job queue.

When a batch job is read from an input source by a reader, the commands in the input stream are stored in the server as requests for the job, the inline data is spooled as inline data files, and an entry for the job is placed on a job queue. The job information remains stored in the server where it was placed by the reader until the job entry is selected from the job queue for processing by a subsystem.

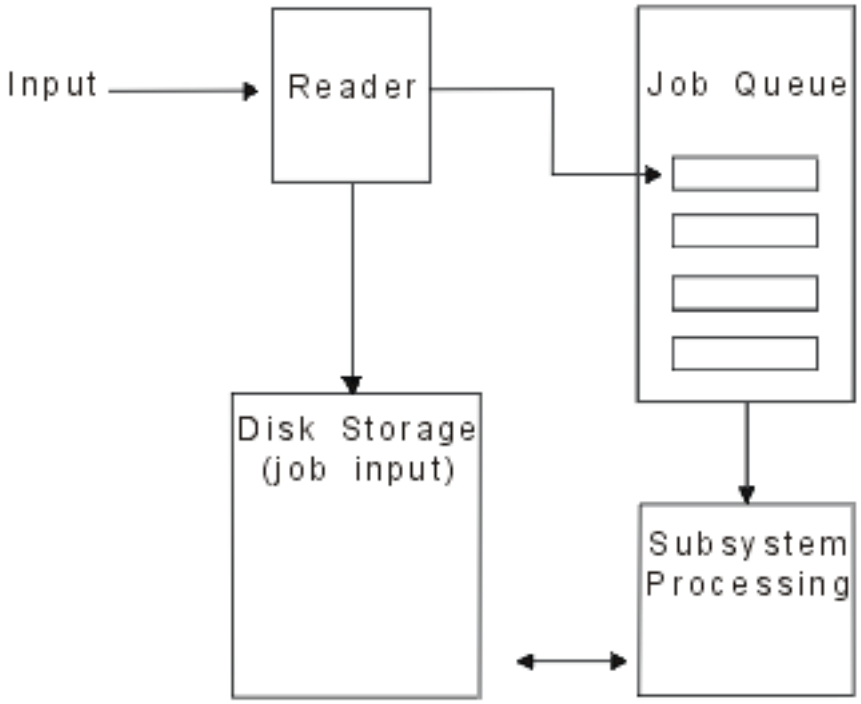

Figure 1. Relationship of Input Spooling Elements

You can use the reader functions to read an input stream from diskette or database files.

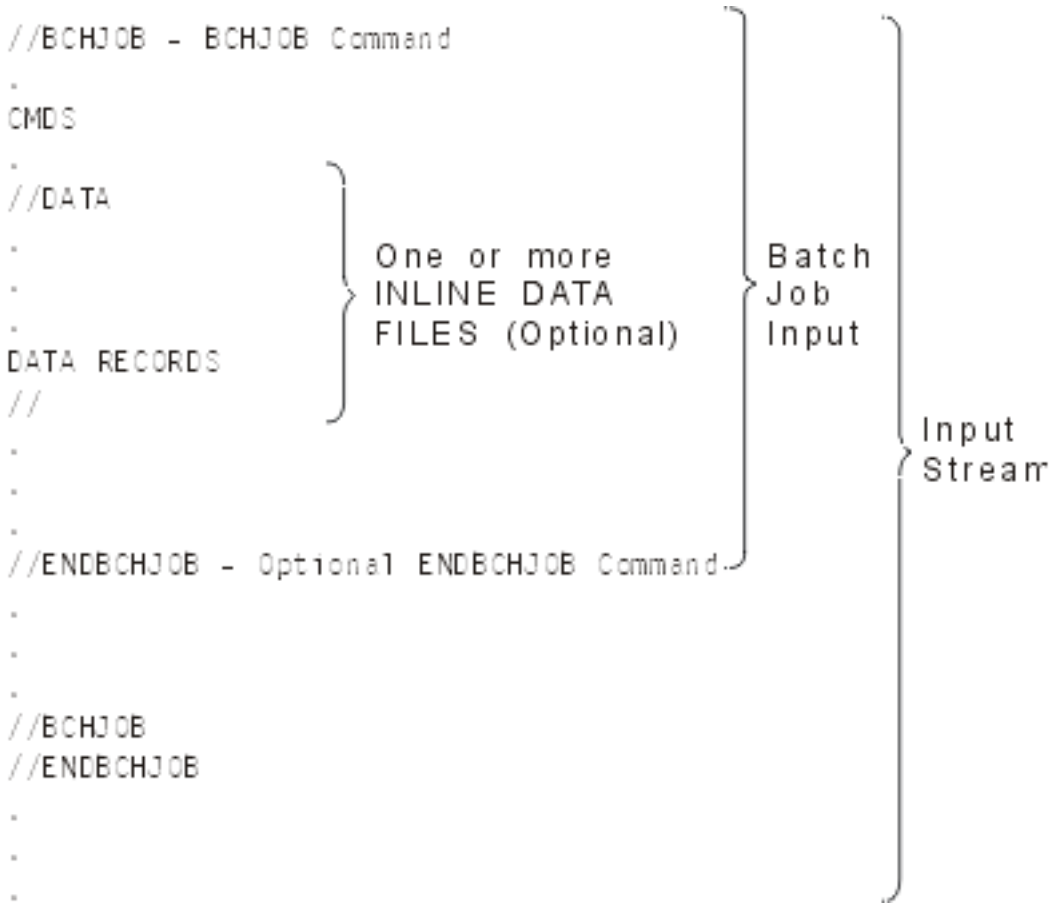

Figure 2. Typical Organization of an Input Stream

The job queue on which the job is placed is specified on the JOBQ parameter on the Batch Job BCHJOB command, on the Start Database Reader STRDBRDR command, or in the job description. If the JOBQ parameter on the BCHJOB command is:

- *RDR: The job queue is selected from the JOBQ parameter on the STRDBRDR command.
- *JOBD: The job queue is selected from the JOBQ parameter in the job description.
- A specific job queue: The specified queue is used.

For jobs with small input streams, you might improve server performance by not using input spooling. The Submit Job (SBMJOB) command reads the input stream and places the job on the job queue in the appropriate subsystem, bypassing the spooling subsystem and reader operations.

If your job requires a large input stream to be read, you should use input spooling (Start Diskette Reader STRDKTRDR or STRDBRDR command) to allow the job to be imported independent of when the job is actually processed.

Summary of job input commands

You can use these commands to submit jobs to the system. The start reader commands may be used for spooling job input; the submit job commands do not use spooling. For detailed descriptions of these commands, see the information about control language in the i5/OS Information Center.

- Batch Job (BCHJOB): Marks the start of a job in a batch input stream and defines the operating characteristics of the job.
- Data (DATA): Marks the start of an inline data file.

- End Batch Job (ENDBCHJOB): Marks the end of a job in a batch input stream.
- End Input (ENDINP): Marks the end of the batch input stream.
- Submit Database Jobs (SBMDBJOB): Reads an input stream from a database file and places the jobs in the input stream on the appropriate job queues.
- Submit Diskette Jobs (SBMDKTJOB): Reads an input stream from diskette and places the jobs in the input stream on the appropriate job queues.
- Start Database Reader (STRDBRDR): Starts a reader to read an input stream from a database file and places the job in the input stream on the appropriate job queue.
- Start Diskette Reader (STRDKTRDR): Starts a reader to read an input stream from diskette and places the job in the input stream on the appropriate job queue.

Using an inline data file

An inline data file is a data file that is included as part of a batch job when the job is read by a reader or a submit jobs command. You use SBMDBJOB or STRDBRDR to queue up a CL batch stream (stream of CL commands to be executed or run). That CL batch stream can include data to be placed into ″temporary″ files (inline files). When the job ends, the inline files are deleted.

An inline data file is delimited in the job by a //DATA command at the start of the file and by an end-of-data delimiter at the end of the file.

The end-of-data delimiter can be a user-defined character string or the default of //. The // must appear in positions 1 and 2. If your data contains a // in positions 1 and 2, you should use a unique set of characters such as: // *** END OF DATA To specify this as a unique end-of-data delimiter, the ENDCHAR parameter on the //DATA command should be coded as:

```
ENDCHAR('// *** END OF DATA')
```

Note: Inline data files can be accessed only during the first routing step of a batch job. If a batch job contains a Transfer Job (TFRJOB), a Reroute Job (RRTJOB), or a Transfer Batch Job (TFRBCHJOB) command, the inline data files cannot be accessed in the new routing step.

An inline data file can be either named or unnamed. For an unnamed inline data file, either QINLINE is specified as the file name in the //DATA command or no name is specified. For a named inline data file, a file name is specified.

A named inline data file has the following characteristics:

- It has a unique name in a job. No other inline data file can have the same name.
- It can be used more than once in a job.
- Each time it is opened, it is positioned to the first record.

To use a named inline data file, you must either specify the file name in the program or use an override command to change the file name specified in the program to the name of the inline data file. The file must be opened for input only.

An unnamed inline data file has the following characteristics:

- Its name is QINLINE. (In a batch job, all unnamed inline data files are given the same name.)
- It can only be used once in a job.
- When more than one unnamed inline data file is included in a job, the files must be in the input stream in the same order as when the files are opened.

To use an unnamed inline data file, do one of the following:

- Specify QINLINE in the program.

- Use an override file command to change the file name specified in the program to QINLINE.

If your high-level language requires unique file names within one program, you can use QINLINE as a file name only once. If you need to use more than one unnamed inline data file, you can use an override file command in the program to specify QINLINE for additional unnamed inline data files.

Note: If you run commands conditionally and process more than one unnamed inline data file, the results cannot be predicted if the wrong unnamed inline data file is used.

Considerations for opening inline data files

You need to consider these elements when you open inline date files.

- Record length specifies the length of the input records. (Record length is optional.) When the record length exceeds the length of the data, a message is sent to your program. The data is padded with blanks. When the record length is less than the data length, the records are truncated.
- When a file is specified in a program, the server searches for the file as a named inline data file before it searches for the file in a library. Therefore, if a named inline data file has the same name as a file that is not an inline data file, the inline data file is always used, even if the file name is qualified by a library name.
- Named inline data files can be shared between programs in the same job by specifying SHARE(*YES) on a create file or override file command. For example, if an override file command specifying a file named INPUT and SHARE(*YES) is in a batch job with an inline data file named INPUT, any programs running in the job that specify the file name INPUT will share the same named inline data file. Unnamed inline data files cannot be shared between programs in the same job.
- When you use inline data files, you should make sure the correct file type is specified on the //DATA command. For example, if the file is to be used as a source file, the file type on the //DATA command must be source.
- Inline data files must be opened for input only.

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.

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 Spooled files 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:

| i5/OS
| IBM
| IBM (logo)
| System i

Other company, product, and 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