

Platform Process Manager
Version 9 Release 1 Modification 2

Release Notes



Platform Process Manager
Version 9 Release 1 Modification 2

Release Notes



Note

Before using this information and the product it supports, read the information in "Notices" on page 19.

First edition

This edition applies to version 9, release 1, modification 2 of IBM Platform Process Manager (product number 5725G82) and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright IBM Corporation 1992, 2013.

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

Contents

Chapter 1. What's new 1

Disable local jobs	1
Generate output and error files for local jobs	1
Define a job submission timeout value.	1
Define exit codes to retry job submission when LSF bsub exits	1
Improved performance of JFD startup and enhanced log file	1
User name displayed in the status bar	1
New flow definition examples installed with Flow Editor	1
Add double quotation marks to the job command instead of single quotation marks	2
Display dependency conditions and variable evaluator expressions	2
Globally define output and error file generation for work items	2
Define a default working directory for flows.	2

Chapter 2. Addendum to existing documentation 5

ppmsetvar cannot be used with local jobs	5
Valid timezone IDs	5
Mainframe Support, File Transfer field	5
JS_LOCAL_EXECUTION_THREADS obsolete	5
User variables supported in more fields in the Job Definition	5

Output and error file generation for work items in a flow	6
Default location of output and error files	6
Override order for output and error file generation settings	7
Configuring output and error file generation for work items in a flow	7
js.conf Reference	8
JS_BSUB_RETRY_EXIT_VALUES.	8
JS_DEFAULT_FLOW_WORKING_DIR.	8
JS_ENABLE_DOUBLE_QUOTE.	10
JS_JOB_SUBMISSION_RETRY	11
JS_JOB_SUBMISSION_TIMEOUT	11
JS_LOCAL_EXECUTION_THREADS	11
JS_LOCAL_JOBS_LIMIT	12

Chapter 3. Process Manager Files 13

Chapter 4. Installation Options 15

Chapter 5. Known Issues 17

Notices 19

Trademarks	21
Privacy policy considerations	21

Chapter 1. What's new

Disable local jobs

A local job is a job that immediately runs on the Process Manager server without going through LSF®. A local job is usually a short and small job. For security reasons, you might want to disable local jobs altogether. You can now disable local jobs by setting the parameter `JS_LOCAL_JOBS_LIMIT=0` in `js.conf`.

Generate output and error files for local jobs

In the Job Definition, you can now specify to generate output and error files for local jobs and define their naming pattern. The behavior is equivalent to the LSF `bsub -o` and `-e` options.

Define a job submission timeout value

There is a new parameter `JS_JOB_SUBMISSION_TIMEOUT` in `js.conf` to define how long Process Manager waits before terminating a job script when job submission takes a long time. The default is 300 seconds.

Define exit codes to retry job submission when LSF bsub exits

Use the new parameter `JS_BSUB_RETRY_EXIT_VALUES` in `js.conf` to specify different exit codes to retry job submission when the LSF `bsub` command exits. The parameter `JS_JOB_SUBMISSION_RETRY` is now deprecated and replaced with `JS_BSUB_RETRY_EXIT_VALUES`.

Improved performance of JFD startup and enhanced log file

Performance of JFD startup has been improved, and more details about JFD startup progress are logged in `jfd.log` at the `LOG_INFO` level.

User name displayed in the status bar

The user name that you are logged in as is now displayed in the status bar in Flow Editor, Flow Manager, and Calendar Editor.

For example, if you are logged in as `lsfadmin` to server `server2`, you would see the following in the status bar at the bottom of the Flow Editor window, Flow Manager Window, and Calendar Editor window:

```
Server: server2:1966 User: lsfadmin
```

New flow definition examples installed with Flow Editor

To help you build your own flows, new examples are now installed with Flow Editor and you can access the examples through **Help > Flow Examples**:

- `Sample.xml`: Demonstrates a simple flow with three jobs and the dependency **Completes successfully**.

- `jobs_and_dependencies.xml`: Demonstrates a flow with three separate branches that run according to whether the first job in the flow completed successfully or ended with a specific exit code or exit codes.
- `event_and_alarm.xml`: Demonstrates how to use time of day and arrival of a file in a specific directory to trigger a subflow. Also uses an alarm to send an email to the administrator.
- `dynamic_flow.xml`, `target_flow.xml`: Demonstrates how to use a subflow by reference, how to pass input variables from a main flow to a subflow, and how to pass a variable from a subflow to the parent flow and use it in a subsequent job in the main flow.
- `flowarray_eval.xml`: Demonstrates how to use the variable evaluator to decide which branch of a flow to run, how to run flow arrays in parallel or sequentially, and how to use a variable to set the array size. This flow contains two subflows to be run as arrays, and a condition evaluator that decides whether to run the arrays in parallel or sequentially.

Add double quotation marks to the job command instead of single quotation marks

By default, Process Manager encloses all job commands with single quotation marks. This can lead to incorrect behavior in some cases. For example, if you have a job command such as `ls -l | awk {print '$2}'`.

Use the new parameter `JS_ENABLE_DOUBLE_QUOTE` in `js.conf` to configure whether job commands are to be quoted with single quotation marks or double quotation marks. When set to `false`(default), all job commands are quoted with single quotation marks. When set to `true`, if the job command does not contain single quotation marks, the job command is quoted with single quotation marks; if the job command contains single quotation marks, it is quoted with double quotation marks.

Display dependency conditions and variable evaluator expressions

You can now display/hide dependency conditions and variable evaluator expressions in the graphical flow in Flow Editor, and in Flow Manager. Select **View > Show Dependency Conditions**. You can also access the menu from the pop-up menu by right-clicking in the flow canvas.

Globally define output and error file generation for work items

You can now set output and error file generation for the flow definition in the Flow Attributes. The settings are inherited by all jobs, local jobs, job arrays, and static subflows, static flow arrays, dynamic subflows, and dynamic flow arrays in the flow. The settings do not apply to job scripts, job array scripts, template jobs, or manual jobs.

The behavior is equivalent to the LSF `bsub -o` and `-e` options.

Define a default working directory for flows

You can now define a default working directory with the parameter `JS_DEFAULT_FLOW_WORKING_DIR` in `js.conf`. The default working directory is used when no working directory is defined in the flow definition or passed to Process Manager with the variable `JS_FLOW_WORKING_DIR`.

Work items inherit the flow settings but users can override the default working directory in individual work items.

Chapter 2. Addendum to existing documentation

ppmsetvar cannot be used with local jobs

The command **ppmsetvar** cannot be used to set variables for local jobs. To set variables for local jobs, use variable files. You can use **ppmsetvar** only to set variables for LSF jobs, job scripts, job arrays and job script arrays.

Valid timezone IDs

You can find a list of valid timezone IDs in `JS_HOME/JS_VERSION/resources/timezones.properties`.

Mainframe Support, File Transfer field

Updated description for File Transfer field in the Execution Environment tab:

File Transfer (Optional).

Specify file transfers between Platform Process Manager Server and LSF execution hosts.

The format of the expression is "*local_file operator [remote_file]*".

The operator(<, >, <>, ><, <<) specifies whether the file is copied to the remote host, or whether the file is copied back from the remote host. The operator must be surrounded by white space.

JS_LOCAL_EXECUTION_THREADS obsolete

The parameter `JS_LOCAL_EXECUTION_THREADS` for local jobs is obsolete since Process Manager version 8.0.2 for Linux/UNIX and Windows.

User variables supported in more fields in the Job Definition

You can now use user variables in more fields in the Job Definition for jobs and job arrays.

The following are the additional fields in which you can now use user variables:

Processing tab

- Number of Processors for Parallel Jobs
 - Minimum
 - Maximum
- User Group
 - Associate job with user group
- Before Execution
 - Run command

Limits tab

- Host Limits

- Maximum CPU time
- Maximum run time
- Job Limits
 - Maximum file size
 - Maximum core file size
 - Maximum memory size
 - Maximum data size
 - Maximum stack size

Output and error file generation for work items in a flow

By default, output and error files are not generated for flows or individual work items.

To troubleshoot flows, however, it is useful to always generate output and error files for work items in the flow.

You can set output and error file generation in the Flow Attributes. The behavior to create output and error files is the same as using the LSF `bsub` command options `-o` and `-e`.

Output and error file settings that are defined in the Flow Attributes are inherited by the following work items in the flow:

- Jobs
- Local jobs
- Job arrays
- Static subflows
- Static flow arrays
- Dynamic subflows
- Dynamic flow arrays

Output and error file settings do not apply to job scripts, job array scripts, template jobs, or manual jobs.

Users can override output and error file settings that were defined in the flow in individual work items.

Default location of output and error files

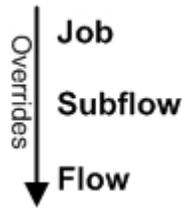
By default, output and error files are generated in the working directory of the work item.

If the working directory is not specified, the location that is used for the working directory is the execution user's home directory:

- Linux: `$HOME`
- Windows: `%HOMEDRIVE%%HOMEPATH%`

You can define a default working directory for flows with the parameter `JS_DEFAULT_FLOW_WORKING_DIR` in `js.conf`. Users can override the default working directory in individual work items.

Override order for output and error file generation settings



The override order for Process Manager to determine output and error file generation, location, and naming is as follows (in order of highest precedence):

1. Use output and error file settings defined at the job level, in the Job Definition.
2. Use output and error file settings defined in the static subflow's Flow Attributes.
3. Use output and error file settings defined in the dynamic subflow's Flow Attributes:
 - If **Use parent flow's settings** is selected, use settings in the parent flow's Flow Attributes.
 - If **Use inserted flow's settings** is selected, use settings in the inserted target flow's Flow Attributes.
 - If **Override parent flow's settings and inserted flow's settings** is selected, use settings in the Dynamic subflow Flow Attributes.
4. Use output and error file settings defined in the flow's Flow Attributes.

Configuring output and error file generation for work items in a flow

Procedure

1. Define a default working directory for the flow by setting `JS_DEFAULT_FLOW_WORKING_DIR` in `js.conf`.
This is useful to centralize output and error file generation when no working directory is defined for the flow in the Flow Definition or with the parameter `JS_FLOW_WORKING_DIR`.
Refer to “`JS_DEFAULT_FLOW_WORKING_DIR`” on page 8 for details on requirements for the default working directory.
2. Restart the Process Manager server to make changes take effect.
3. In Flow Editor, from the menu select **Action > Add Flow Attribute**, and define output and error file settings.

Field	Description
Create output files for jobs and job arrays	Select Yes to configure output and error file generation and naming pattern.
Create error files for jobs and job arrays	Important: If you select to only generate output files, no error files are generated but the content of the error file is appended to the output file(same behavior as the LSF <code>bsub -o</code> option).
Directory	Specify a directory name or path relative to the flow's working directory. Process Manager creates specified subdirectories in the working directory if the directories do not exist.

Field	Description
File Name	<p>Specify the naming pattern for files.</p> <p>Built-in variables you can use:</p> <ul style="list-style-type: none"> • %u for user name • %t for time stamp • %J for job ID • %I for job array element <p>Default file naming pattern for output files:</p> <ul style="list-style-type: none"> • Job: output.#{JS_FLOW_FULL_NAME}.%J • Local job: output.#{JS_FLOW_FULL_NAME} • Job array element: output.#{JS_FLOW_FULL_NAME}.%J[%I] <p>Default file naming pattern for error files:</p> <ul style="list-style-type: none"> • Job: error.#{JS_FLOW_FULL_NAME}.%J • Local job: error.#{JS_FLOW_FULL_NAME} • Job array element: error.#{JS_FLOW_FULL_NAME}.%J[%I]

4. Click **OK** to save your changes.

js.conf Reference

The following are new or changed parameters in `js.conf` for version 9.1.2.

JS_BSUB_RETRY_EXIT_VALUES

Syntax

```
JS_BSUB_RETRY_EXIT_VALUES=exit_code[, exit_code...]
```

Description

Specifies **bsub** exit codes to retry job submission. Separate multiple exit codes with a comma (,).

When job submission fails and the LSF **bsub** command exits with any of the specified exit codes, Process Manager retries to submit the job again. The number of retries is specified with the parameter `JS_START_RETRY` in `js.conf`.

Default

Undefined. There is no retry when job submission fails.

See also

`JS_START_RETRY`

JS_DEFAULT_FLOW_WORKING_DIR

Syntax

```
JS_DEFAULT_FLOW_WORKING_DIR=path
```

Description

Specifies the default working directory that is used by flows when no working directory is defined in the flow definition or passed to Process Manager with the variable `JS_FLOW_WORKING_DIR`.

Process Manager creates the default working directory and any specified subdirectories if the directories do not exist.

To specify subdirectories, you can use the built-in variables:

- `%u` for user name
- `%t` for time stamp
- `#{JS_FLOW_NAME}`
- `#{JS_FLOW_ID}`

Work items inherit the flow working directory unless users override the working directory in individual work items.

The override order for working directories is (in order of highest precedence):

1. The working directory defined at the job level, in the Job Definition.
2. The working directory defined at the subflow level, in the subflow's Flow Attributes.
3. The working directory defined at the flow level, in the Flow Attributes.
4. The working directory specified with the variable `JS_FLOW_WORKING_DIR` when you trigger a flow.
5. The working directory defined with `JS_DEFAULT_FLOW_WORKING_DIR` in `js.conf`.
6. The execution user's home directory:
 - Linux: `$HOME`
 - Windows: `%HOMEDRIVE%%HOMEPATH%`

Requirements for the working directory:

- The directory and parent directories to the working directory must be shared and accessible to the Process Manager server and all LSF execution hosts.
For example, if you specify `JS_DEFAULT_FLOW_WORKING_DIR=/home/lfsadmin/#{JS_FLOW_NAME}_%t`, the directory `/home/lfsadmin` must exist and must be shared.
- The root shared directory must be writable by the user who triggers the flow, and by execution users of individual work items (Run As user).
- If the root directory cannot be shared, the directory:
 - Must exist on all LSF execution hosts
 - Must have the same path on all LSF execution hosts
 - Must be writable by the user who triggers the flow definition and by execution users of work items in the flow and any subflows
- Windows: The working directory must be set to a physical drive on the machine such as `C:\`. If you need to refer to a network drive, create a symbolic link inside your `C:\` drive.

Automatically created directories have the following permissions:

- Owner is the execution user

- Owning group is the execution user's group
- Read and write execute permissions are set for the owner
- Linux: read and write execute permissions are set for all
- Windows: new folders have full permissions set for the execution user

Default

Undefined. When no working directory is specified for the flow in the flow definition or with `JS_FLOW_WORKING_DIR`, the location that is used for the working directory is the execution user's home directory:

- Linux: `$HOME`
- Windows: `%HOMEDRIVE%%HOMEPATH%`

Examples

Create a separate directory for each flow that is triggered from the flow definition in the user's home directory, with a time stamp at the end:

- Linux: `JS_DEFAULT_FLOW_WORKING_DIR=/home/%u/#{JS_FLOW_NAME}_%t`
If the user is `user1`, the flow name `myflow`, and the flow ID `123`, the directory that is created is:
`/home/user1/123:user1:myflow_1366648793`
- Windows: `JS_DEFAULT_FLOW_WORKING_DIR=C:\shared\%u\#{JS_FLOW_NAME}_%t`
If the user is `user1`, the flow name `myflow`, and the flow ID `123`, the directory that is created is:
`c:\shared\123_user1_myflow_1366648793`

JS_ENABLE_DOUBLE_QUOTE

Syntax

`JS_ENABLE_DOUBLE_QUOTE=true | false`

Description

Applies only to Linux/UNIX.

Applies to LSF jobs.

When this parameter is set to `false`, the LSF **bsub** job command is always quoted with single quotation marks.

When this parameter is set to `true`:

- If the LSF **bsub** job command does not contain single quotation marks, the job command is quoted with single quotation marks.
- If the LSF **bsub** job command contains single quotation marks, it is quoted with double quotation marks.

Adding double quotation marks means that the command will be interpreted before it is sent to LSF.

For example, if the command is `ls -l | awk {print '$2}'`, it becomes: `bsub "ls -l | awk {print '$2}'"`.

Default

False. The LSF `bsub` job command is always quoted with single quotation marks.

JS_JOB_SUBMISSION_RETRY

Syntax

`JS_JOB_SUBMISSION_RETRY=true | false`

Description

Deprecated. Use `JS_BSUB_RETRY_EXIT_VALUES` instead.

Specifies whether to retry job submission after the job fails.

If `JS_JOB_SUBMISSION_RETRY=true` and `JS_BSUB_RETRY_EXIT_VALUES` is not defined, job submission is retried when the LSF `bsub` exit code is 1, 255, 127, -1, 128.

If `JS_BSUB_RETRY_EXIT_VALUES` is defined, `JS_JOB_SUBMISSION_RETRY` is ignored and considered deprecated. Process Manager attempts to submit the job again when LSF `bsub` exits with the exit codes specified in `JS_BSUB_RETRY_EXIT_VALUES`.

If `JS_BSUB_RETRY_EXIT_VALUES` and `JS_JOB_SUBMISSION_RETRY` are not defined, there is no retry when job submission fails.

Default

False. There is no retry when job submission fails.

JS_JOB_SUBMISSION_TIMEOUT

Syntax

`JS_JOB_SUBMISSION_TIMEOUT=seconds`

Description

Applies to job scripts.

Maximum number of seconds that the job script can take to submit jobs to LSF before the Process Manager daemon (`jfd`) terminates the script.

Specify 0 to set the maximum time to unlimited.

Default

300 seconds

JS_LOCAL_EXECUTION_THREADS

Syntax

`JS_LOCAL_EXECUTION_THREADS=number_of_jobs`

Description

This parameter is deprecated since version 8.0.2.

Specifies the maximum number of local jobs that can be run in parallel independent of flows. There is one thread created per job.

Default

The parameter value is fixed to one, regardless of the value that you configure.

JS_LOCAL_JOBS_LIMIT

Syntax

`JS_LOCAL_JOBS_LIMIT=number_of_jobs`

Description

Specifies the maximum number of local jobs that can be run in parallel on the Process Manager Server.

When this parameter is set to 0, local jobs are disabled:

- If any existing flows contain local jobs, the local jobs are not run and exit with an exit code of 1.
- In Flow Editor, local jobs cannot be inserted in the flow definition, and any flow definitions that contain local jobs cannot be submitted.
- In Flow Manager, flow definitions that contain local jobs cannot be triggered, released, or published.

Default

The larger number between 1, and the number of cores on the Process Manager host - 2. For example, if the Process Manager host has 4 cores, the maximum number of local jobs that can be run in parallel is 2.

Chapter 3. Process Manager Files

Documentation	File Name
<i>System Requirements</i>	sys_requirements.html
<i>Installing Platform Process Manager (UNIX)</i>	pm_install_unix.pdf
<i>Installing Platform Process Manager (Windows)</i>	pm_install_windows.pdf
<i>Administering Platform Process Manager</i>	pm_admin.pdf
<i>Using Platform Process Manager</i>	pm_using.pdf
<i>Guide to Using Templates</i>	pm_using_templates.pdf

Chapter 4. Installation Options

Default installation

For a default installation on either Windows or UNIX, follow the applicable installation instructions in the following guides:

Book	File
<i>Installing Platform Process Manager (Windows)</i>	pm_install_windows.pdf
<i>Installing Platform Process Manager (UNIX)</i>	pm_install_unix.pdf

Control the JFD

- To start JFD:
 - UNIX: **jadmin start**
 - Windows:
 1. From the Windows Control Panel, select **Administrative Tools > Services**.
 2. Right-click **Platform Process Manager** and select **Start**.
- To stop JFD, run **jadmin stop**. Do not manually terminate the jfd process.

Chapter 5. Known Issues

Process Manager 9.1.2.0 has the following known issues:

- On Linux, `jfd` requires the `libstdc++.so.5` library. If the library is not found on your system, install `libstdc++` or the `compat-libstdc++` RPM containing the `libstdc++.so.5` library that is appropriate to your system.
- **bhist** and **bacct** do not work on Windows hosts running in a UNIX cluster. This occurs because Windows hosts cannot read the shared directory on UNIX. To work around this issue, run **bhist** or **bacct** from a UNIX host in the cluster.
- If you installed YouDao (translation software), dialog windows in the Calendar Editor may disappear when you double-click or drag the window, or drag a scroll bar in the window. This problem occurs with every item in the Calendar Editor except **Calendar > New Calendar > Combine calendars**.

For example, when in the **Owners > Sys** directory tree, if you double-click **Daily**, the **Daily@Sys** window is displayed in the main window. If you then double-click or drag this window, it disappears. If this window has a scroll bar, the window also disappears if you drag the scroll bar.

This is a problem with the YouDao software. To work around this issue, shut down YouDao.

- When a user variable is used in a job definition's **Name** field, the value of the variable must be set when the flow is triggered. Otherwise, the flow fails to be triggered. For example, in Flow Manager, choose **Trigger > With Variables**, or use the command `jtrigger -v variable_list`.
- Applies to Linux hosts when `JS_TIME_ZONE` is set to `client`, the default. During daylight saving time (in the March-November time period), some time events may trigger an hour early. This occurs when Java™ gets the wrong timezone information from the client machine. Solution: On the (Linux) client, edit the file `/etc/sysconfig/clock` and replace `TIMEZONE=zone` with `ZONE=zone`. For example, `ZONE="US/Eastern"`.
- On Linux Red Hat 7.0, due to a known Java™ problem, scripts that run Process Manager commands and monitor for exit codes never see an exit code 1 when it occurs—they see an exit code of 0.
- A Misschedule exception is not propagated through two consecutive link events in a flow.
- When using the Flow Editor or Flow Manager, and viewing or changing a job or job array definition, the entire job or job array definition dialog cannot be viewed if you are using a screen resolution lower than 1024 x 768. We recommend you set your resolution to 1024 x 768 or higher when using the Process Manager graphical user interface.
- When launching the graphical interfaces (Flow Editor, Calendar Editor and Flow Manager) on some UNIX systems, the user receives font not found warnings. The Java Virtual Machine user interface manager automatically determines which fonts are used on each machine. If the fonts are not available on the machine, the warning is issued. These warning messages do not affect the operation of the client.
- Process Manager does not validate the end date of a calendar expression. Consequently, if you define an end date prior to the start date, you receive no error message, but the calendar cannot be used in a time event to successfully trigger a flow or job.

- In the Flow Editor, on the Job Definition tabs, the **Reset** button only resets data on the current tab—it does not affect the hidden tabs. Values specified in the following fields do not reset at all when you click the **Reset** button:
 - Submit to queue/partition(s)—General tab
 - User name—General tab
 - Host requirements fields—Processing tab
- Some days of the week are not completely shown in the Calendar Editor for some platforms:
 - On Solaris: Mon, Wed, Thu are shown as M..., W..., T...
 - On HP: Mon, Wed are shown as M..., W...
- Some historical information is not logged correctly in the following circumstances:
 - If a job or other activity has a rerun exception handler that is triggered, no information about the exit code of the job is logged.
 - When user variables are used for file transfer names, the resolved name of the file is not logged.
- When using the **jsub -r** command to submit and replace a flow definition, if the **jsub** command fails, the replacement does not occur, but the replaced flow definition is already deleted and cannot be restored.
- In the Flow Manager, it is possible for the user to have two different flows selected in the tree view and the right-hand pane. If the user selects an option in the **Action** menu, such as Kill, it is the flow selected in the tree view that is killed, not the one in the right-hand pane. This may not be the behavior the user expects.
- Process Manager does not support duplicate event logging in LSF. The `lsh.events` file must be in a shared location where the Process Manager server can locate job information.
- If a job using a rerun exception handler has a rerun delay specified, and the Process Manager restarts during the delay period (prior to rerunning the job), the job is never rerun.
- If a user tries to open the job dependency definition on the arrow coming out of the job with the variable in the job name field, the user sees an empty message dialog, prior to seeing the dependency definition. The user can click **OK**, to see the definition. However, the job name field is blank.
- If a job command (command itself, not parameters) contains space and if the job is submitted to a queue with a job starter, the command doesn't work.

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 character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, 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
Intellectual Property Law
Mail Station P300
2455 South Road,
Poughkeepsie, NY 12601-5400
USA

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 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 measurement 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 illustrates 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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample 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_.

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

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)[®] are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>.



Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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

LSF, Platform, and Platform Computing are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation 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.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, See IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.



Printed in USA

GI13-1891-02

