WebSphere® Process Server for z/OS

Version 6.0.1

Tuning

Note

Before using this information, be sure to read the general information in "Notices" on page 11.

23 June 2006

This edition applies to version 6, release 0, modification 1 of WebSphere Process Server for z/OS (product number 5655-N53) and to all subsequent releases and modifications until otherwise indicated in new editions.

To send us your comments about this document, email doc-comments@us.ibm.com. We look forward to hearing from you.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Tuning business processes

Use this task to improve the performance of business processes.

After successfully running business processes, you can perform this task to improve performance.

1. Define how to measure the baseline performance, and which measurements you want to optimize.

   For example, for some business applications, it is preferable to reduce the response time for end-users under peak-load conditions. For other applications, the rate that the system can process transactions might be more important than the actual duration of each transaction.
2. Make baseline measurements.

   Make the baseline measurements under conditions of load, time-of-day, and day-of-week that are appropriate for tuning your application. Normally, the most important baseline measurements are the throughput and response times. Throughput values are measured after a specific bottleneck capacity is reached, for example 100% CPU load, disk I/O at maximum, or network I/O at 100%. Reliable response time values are best measured for a single process instance during low server utilization.
3. Tune the processes.

   Depending on whether your application uses long-running processes or microflows, perform one of the following steps:
   - To tune long-running processes, perform the steps that are described in "Tuning long-running processes" on page 2. These processes are characterized by needing external stimuli or human interaction. Their performance therefore depends on the performance of the Business Process Choreographer database and the messaging service.
   - To tune microflows, perform the steps that are described in "Tuning microflows" on page 9. These processes tend to run for only a short time. They use the database only for audit logging, if enabled, and to retrieve the template information. They do not use messaging support for storing persistent data. These processes involve no human interaction.
4. Tune the application.

   Many different options are available to achieve the same functionality in an application, and some of them are more efficient than others. Identify and review any performance-critical code. Maximize asynchronicity, and ensure that actions are not unnecessarily serialized. Consider the overhead of serializing and deserializing data that is passed along a chain of activities. Consider shortening timeouts that do not result in error conditions. Identify opportunities to cache the results of database queries.
5. Review the current configuration for performance bottlenecks that can be eliminated.

   Possibilities to consider include:
   - Installing more processors, more memory, and faster disks.
   - Storing the database logs on different physical disks from the data, and distributing the data on several disks.
   - Using DB2®, rather than Cloudscape™, for optimal performance.

6. Repeat the benchmark measurements under similar load conditions to those of the baseline measurements.

   Keep a permanent record of the application performance measurements to objectively measure any future changes in performance.

The business processes are configured to run measurably faster.

Tuning long-running processes

Use this task to improve the performance of long-running business processes.

Long-running processes can include user-interaction, asynchronous invocations, multiple receives, picks, and event handlers, for example; they use database and messaging subsystems for storing persistent states. The following topics describe how to improve the performance of long-running processes.

Specifying initial database settings

Use this task to specify initial DB2 database settings. Note that this information is provided only as an example.

Attention: The following information relates to the Business Process Choreographer database. For information about tuning a WebSphere® default messaging database, see Tuning and problem solving for messaging engine data stores in the WebSphere Application Server Network Deployment information center.

To achieve good database operation, specify the initial database settings. You will fine-tune the settings later, in "Fine-tuning the database" on page 8.

1. Separate the log files from the data files.

   Putting the database log file on a disk drive that is separate from the data tends to improve performance, provided that sufficient disk drives are available. If few disk drives are available, distributing the table spaces, as described in the previous section, is usually more beneficial than putting the database log on a separate drive.

   For example, if you use DB2 on a Windows® system, you can change the location of the log files for the database named `BPEDB` to the `F:\db2logs` directory, by entering the following command:

   ```
   db2 UPDATE DB CFG FOR BPEDB USING NEWLOGPATH F:\db2logs
   ```

2. Create table spaces.

   After you create the database, explicitly create table spaces. Example scripts to create table spaces are provided by Business Process Choreographer in the `ProcessChoreographer` subdirectory of your WebSphere Application Server installation. Customize these scripts to accommodate the needs of a particular scenario. Your goal, when creating the table spaces, is to distribute input and output operations over as many disk drives as possible that are available to DB2. By default, these scripts create the following table spaces:

   AUDITLOG

   Contains the audit trail tables for processes and tasks. Depending on the degree of auditing that is used, access to tables in this table space can be significant. If auditing is turned off, tables in this table space are not accessed.

   COMP

   Contains the compensation tables for business processes from Business

Process Choreographer Version 5. Depending on the percentage of compensable processes and activities, the tables in this table space might require high disk bandwidth. If compensation is not used within business processes, the tables in this table space are not used.

INSTANCE
Holds the process instance and task tables. It is always used intensively, regardless of the kind of long-running process that is run. Where possible, spread this table space over several disk drives.

SCHEDTS
Contains the tables that are used by the WebSphere scheduling component. Access to tables in the scheduler table space is usually low, because of the caching mechanisms used in the scheduler.

STAFFQRY
Contains the tables that are used to temporarily store staff query results that are obtained from staff registries like Lightweight Directory Access Protocol (LDAP). When business processes contain many person activities, tables in this table space are frequently accessed.

TEMPLATE
Contains the tables that store the template information for processes and tasks. The tables are populated during the deployment of an application. At run time the access rate is low. The data is not updated, and only new data is inserted during deployment.

WORKITEM
Holds the tables that are required for work item processing. Work items are used for human task interaction. Depending on the number of human tasks in the business processes, access to the tables in this table space can vary from a low access rate to significantly high access rate. The access rate is not zero, even when no explicit human tasks are used, because work items are also generated to support administration of long-running processes.

To create a database for high performance, perform the following actions:

a. Create the database.
b. Create the table spaces on the disks if the database supports table spaces.

   For example, the following script is based on the `createTablespaceDb2.ddl` file that is located in the `ProcessChoreographer` subdirectory of your WebSphere Application Server installation. In the following example, it creates the database, table spaces, and tables that use three disk drives on a Windows system:

```
- Scriptfile to create tablespaces for DB2 8.1 and 8.2
- to run the script call
-      db2 connect to BPEDB
-      db2 -tf createTablespaceDb2.ddl
CREATE TABLESPACE AUDITLOG
  MANAGED BY SYSTEM
  USING( 'F:/BPEDB_TS/AUDITLOG' );

CREATE TABLESPACE COMP
  MANAGED BY SYSTEM
  USING( 'D:/BPEDB_TS/COMP' );

CREATE TABLESPACE INSTANCE
  MANAGED BY SYSTEM
  USING( 'E:/BPEDB_TS/INSTANCE' );

CREATE TABLESPACE STAFFQRY
```

```
    MANAGED BY SYSTEM
    USING( 'D:/BPEDB_TS/STAFFQRY' );

  CREATE TABLESPACE TEMPLATE
    MANAGED BY SYSTEM
    USING( 'D:/BPEDB_TS/TEMPLATE' );

  CREATE TABLESPACE WORKITEM
    MANAGED BY SYSTEM
    USING( 'D:/BPEDB_TS/WORKITEM' );

  -- start import scheduler DDL: createTablespaceDB2.ddl
  CREATE TABLESPACE SCHEDTS MANAGED BY SYSTEM USING( 'D:/BPEDB_TS/SCHEDTS' );
  -- end import scheduler DDL: createTablespaceDB2.ddl
```

c. Create the Business Process Choreographer tables.

   Use the script for your database that is provided in the `ProcessChoreographer` directory. For example, for DB2, use the `createSchemaDb2.ddl` file.

3. Tune the database.

   Use a capacity planning tool for your initial database settings.

   If you are using DB2, start the DB2 configuration advisor from the DB2 Control Center, by selecting **DB2 configuration advisor** from the pop-up menu of the Business Process Choreographer database. Do the following actions:

   a. Allocate memory to DB2.

      For **Server**, allocate to DB2 only as much memory as is physically available to it without swapping.

   b. Specify the type of workload.

      For **Workload**, select **Mixed** (queries and transactions).

   c. For **Transactions**, specify the length of the transactions and the estimated number of transactions to be processed each minute.

      Select **More than 10**, to indicate that long transactions are used.

      Then, in the **Transactions per minute** field, select the estimated number of transactions processed each minute. To determine this number, assume that each activity in the process has one transaction. The number of transactions performed in one minute is then as follows:

      *number of transactions performed each minute = number of processes completed each minute * number of activities in each process*

   d. Tune the database for faster transaction performance and slower recovery.

      For **Priority**, select **Faster transaction performance**.

   e. If possible, tune the database populated with the typical amount of data in production. For **Populated**, select **Yes**. Otherwise, select **No**.

   f. Tune the parallel connections setting.

      For **Connections**, specify the maximum number of parallel connections that can be made to the application server. Guidelines for determining this value are as follows:

      - The number of database connections required is determined by the number of Java™ DataBase Connectivity (JDBC) connections to the WebSphere Application Server. The JDBC connections are provided by the JDBC connection pool, which is in the WebSphere Application Server. For *p* JDBC connections, *p* * 1.1 database connections are required. How to estimate a realistic value for *p* is described in "Tuning the application server" on page 6.

- If Business Process Choreographer and the database are installed on the same physical server, Business Process Choreographer needs no remote database connections. However, because remote connections might be required for remote database management, specify a low value, rather than zero.
- If Business Process Choreographer and DB2 are installed on separate servers, set the number of remote applications in accordance with the rule previously described for local connections.

g. Lock the rows you want to read.

   For **Isolation**, select **Read stability**. This isolation level is required for Business Process Choreographer.

The configuration advisor displays suggested changes. You can either apply the changes now, or save them to a file to apply later.

Your long-running processes are running as fast as possible under the current environment and loading conditions.

Planning queue manager settings

Use this task to plan queue manager settings.

To achieve the best performance for long-running processes, tune the message queuing system for maximum performance of persistent messages.

If you use default messaging, follow the instructions given in Service integration in the WebSphere Application Server Network Deployment information center, to set up and tune the data stores for the messaging engines.

If you use the IBM® WebSphere MQ messaging product, rather than the default messaging services, complete the following steps.

1. Tune MQ parameter settings.

   Tune the following MQ parameter settings:
   - Log file pages
   - Log buffer page
   - Log primary files
   - Log secondary files
   - Log default path
   - Maximum channels
   - Channel application bind type

   The default locations for both the persistent queue data and the MQ logs is the MQ installation directory. Put the data storage for the persistent queues and the WebSphere MQ logs on different disk drives. By changing the path to the log file to refer to another disk drive, you can change the location for the MQ logs. Make these changes before you create the queue managers for Business Process Choreographer.
2. Tune WebSphere MQ service properties for Business Process Choreographer.

   These values must be set before you create the queue managers that are used by Business Process Choreographer. Set each parameter to its maximum value, as shown in the following table:

Table 1. Tuning WebSphere MQ service properties for Business Process Choreographer

Parameter	Value	Comment
Log file pages	16384	On Windows systems, not all versions of WebSphere MQ support setting the number of log file pages to 16384 by using the MQ administration tools. In this case, change the value of the Windows registry key: `HKEY_LOCAL_MACHINE\SOFTWARE\IBM\MQSeries\CurrentVersion\Configuration\LogDefaults` to: `16384`
Log primary files	10	
Log secondary files	53	
Log buffer pages	512	

3. Tune the queue manager properties.

   Specify the queue manager properties for the maximum number of channels and the type of binding for the channel application, as shown in the following table:

Table 2. Tuning queue manager properties

Queue manager properties	Value
Maximum channels	Use the default
Channel application bind type	FASTPATH

Your queue manager is operating optimally.

Tuning the application server

Use this task to tune the application server.

Before you start this task, you must have specified the initial settings for the database, as described in “Specifying initial database settings” on page 2.

To ensure that the business process container can preform optimally, you need to adjust the server settings.

1. Estimate the application server resources that you need for each business process container.
   a. One data source, to read and write business process state information to a database: jdbc/BPEDB
   b. Calculate the maximum concurrency of transactions, *t*, for the process navigation by adding the following:
      - The maximum number of clients concurrently connected through the Business Process Choreographer API
      - The number of concurrent endpoints defined in the JMS activation specification BPEInternalActivationSpec
      - The number of concurrent endpoints defined in the JMS activation specification HTMInternalActivationSpec

Note: To view the activation specifications for the process server, in the administrative console, click **Resources** → **JMS Providers** → **Default messaging** → **JMS activation specification**.

c. Calculate the number of parallel JDBC connections required to the process server database, $p = 1.1 * t$

Note: The value of p must not be greater than the number of connections allowed by the database.

d. Calculate the number of parallel JDBC connections required to the messaging database, $m = 2 * t + x$, where x is the number of additional JMS sessions to allow for overloads situations where additional messages are generated and must be served

2. Tune the JDBC provider settings for the process server database (BPEDB).
   a. Set **Max Connections** to the value p. The value of p must not be greater than the number of connections allowed by the database.
   b. Set the SQL **Statement cache size** to 500.
3. Tune the JDBC provider settings for the messaging database.

   Set **Max Connections** to the value m.
4. Tune the heap size.

   Here are some guidelines for the size of the server heap:
   - 256 MB is too low, and results in poor performance.
   - 512 MB is adequate as an initial heap size for many systems.
   - 1024 MB is a reasonable upper limit.
5. Tune any services that are used by your business processes. Make sure that your supporting services are tuned to cope with the degree of concurrency and load demands that Business Process Choreographer makes on the service.

The application server is tuned for improved performance.

Fine-tuning WebSphere MQ Messaging Provider

Use this task to improve the rate at which WebSphere MQ processes messages if you use IBM WebSphere MQ Messaging Provider.

If, after you complete the tuning actions, less than 80% of your CPU capacity is used, or if the CPU usage graph shows bursts of activity, experiment with the custom property `non.asf.receive.timeout`. To change this property, use the administrative console.

CPU usage is improved by smoothing peaks, or bursts, of activity.

Create the custom property `non.asf.receive.timeout`, on the message listener service of the application server, with a value of 2000.

This setting switches the message listener service to non-ASF mode.

The rate at which WebSphere

MQ processes messages is improved.

Fine-tuning the database

Use this task to fine-tune the database.

The business process container and business processes must be running.

A common problem is that the database runs out of lock list space, resulting in lock escalation, which severely impacts performance. Depending on the structure of the business processes run, you might therefore need to customize the settings of certain performance-related parameters in your database management system.

Note: If you are not using DB2, refer to your database management system documentation for information about monitoring the performance of the database, identifying and eliminating bottlenecks, and fine-tuning its performance. The rest of this topic offers advice for DB2 users.

1. Tune the lock list space, to help ensure optimum performance.

   Check the `db2diag.log` file for your DB2 instance. Look for entries like the following example:

   ```
   2005-07-24-15.53.42.078000 Instance:DB2 Node:000
   PID:2352(db2syscs.exe) TID:4360 Appid:*LOCAL.DB2.027785142343
   data management sqldEscalateLocks Probe:4 Database:BPEDB

   ADM5503E The escalation of "10" locks on table "DB2ADMIN.ACTIVITY_INSTANCE_B_T"
   to lock intent "X" has failed. The SQLCODE is "-911".
   ```

   This type of message indicates that the parallelism for business process applications has improved to the point where the number of available locks is now too small. Increase the LOCKLIST value to approximately 10 * *p*, where *p* is your estimate for the maximum number of parallel JDBC connections that are required at any time. For example, if you sized your Business Process Choreographer database, BPEDB, with a value of *p*=50, enter the following command:

   ```
   db2 UPDATE DB CFG FOR BPEDB USING LOCKLIST 500
   ```

2. If you used the DB2 configuration advisor, your database throughput is already good. You can, however, further improve the performance, in the following ways:
   - Follow the best practices for database tuning that are described in the DB2 online documentation, books, and articles.
   - Use DB2 monitors, and examine the `db2diag.log` file for more information about bottlenecks within the database.
   - Regularly run runstats on your database.
   - Tune the following DB2 parameters:

     LOCKLIST
     : See the description in step 1.

     AVG_APPLS
     : It is better to set this parameter too high rather than too low. For example, if there are a maximum of 20 connected applications, set AVG_APPLS to 50.

     LOGBUFSZ
     : Increasing the size of the buffer for the DB2 log decreases how often a full log buffer must be written to disk.

     LOG_FILSIZ
     : Increasing the size of the log files reduces how often they are switched.

Your long-running processes are running as fast as possible under the current environment and loading conditions.

Tuning microflows

Use this task to improve the performance of microflows.

Microflows run in memory, without any user-interaction or persistent messaging support. Database access is required only if audit logging or Common Event Infrastructure (CEI) are enabled for the microflow. The entire processing of a microflow occurs in a single thread, and normally, in a single transaction. The performance of microflows mainly depends on the services called. However, it the memory available for the server is too small, the performance of microflows will be reduced.

Tune the Java Virtual Machine (JVM) heap size.

By increasing the Java heap size, you can improve the throughput of microflows, because a larger heap size reduces the number of garbage collection cycles that are required. Keep the value low enough to avoid heap swapping to disk. For guidelines on the size of the server heap, see the relevant step in "Tuning the application server" on page 6.

Your microflows are running as fast as possible under the current environment and loading conditions.

Tuning business processes that contain human tasks

There are various ways to improve the performance of business processes that contain human tasks.

The following topics describe how to tune business processes that contain human tasks.

Reduce concurrent access to work items

When two or more people try to claim the same work item, only one person will succeed. The other person is denied access.

Only one person can claim a work item. If several people attempt to work with the same work item at the same time, the probability of collision increases. Collisions cause delays, because of lock waits on the database or rollbacks. Some ways to avoid or reduce the incidence of collision are as follows:

- If concurrent access is high, limit the number of users who can access a particular work item.
- Avoid unnecessary work item queries from clients, by using intelligent claim mechanisms. For example, you might take one of the following steps:
  - Try to claim another item from the list if the first claim is unsuccessful.
  - Always claim a random work item.
  - Reduce the number of people in each group.
  - Limit the size of the work item list, either by using a custom property in the where clause of the query or by setting a low threshold.
  - Minimize or avoid dynamic staff queries.

– Use a client caching mechanism for work item queries, to avoid running several queries at the same time.

Reduce query response time

Reduce the time that the database takes to respond to queries.

When you use a custom client, make sure that the queries set a threshold. From a usability viewpoint, retrieving hundreds or thousands of items is typically undesirable, because the larger the number of database operations, the longer the task takes to complete, and because a person can manage only a small number of results at a time. By specifying a threshold, you minimize database load and network traffic, and help to ensure that the client can present the data quickly.

A better way to handle a query that returns a large number of items might be to rewrite the query, to return a smaller result set of items. You can do this by querying work items for only a certain process instance or work items with only a certain date.

You can also reduce the query result by using filter criteria.

Avoid scanning whole tables

When you use the query application programming interfaces (APIs), to list the objects in the database, you can specify filters that narrow the results you want to retrieve. In these filters, you can specify the values and ranges of object attributes.

When database queries are processed, the filter information is translated into WHERE clauses in a Structured Query Language (SQL) statement. These WHERE clauses map the object attributes to column names in the affected database tables.

If your query specifies a filter that does not translate to an indexed table column, the SQL statement will probably cause the table to be scanned. This scanning impacts performance negatively and increases the risk of deadlocks. Although this performance impact can be tolerated if it happens only a few times a day, it could adversely affect efficiency if it took place several times a minute.

In such circumstances, a custom index can dramatically reduce the impact. In a real customer situation, a custom index helped to reduce the API response time from 25 seconds to 300 milliseconds. Instead of reading 724 000 rows of the database table, only six rows had to be read.

Depending on the filter criteria that you specify, some columns might not be included in an index. If this is the case, and if a table scan is used, resulting in slow query performance, check the access path of the statement, using DB2 Explain, for example. If necessary, define a new index.

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
577 Airport Blvd., Suite 800
Burlingame, CA 94010
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 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 is for planning purposes only. The information herein is subject to change before the products described become available.

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.

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

Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries, or both: IBM, IBM (logo), AIX, CICS, Cloudscape, DB2, DB2 Connect, DB2 Universal Database, developerWorks, Domino, IMS, Informix, iSeries, Lotus, MQSeries, MVS, OS/390, Passport Advantage, pSeries, Rational, Redbooks, Tivoli, WebSphere, z/OS, zSeries

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

Microsoft and Windows 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.

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

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

This product includes software developed by the Eclipse Project (http://www.eclipse.org/).

IBM Websphere Process Server for z/OS version 6.0.1

Printed in USA