

IBM® DB2® for Linux®, UNIX®, and Windows®

*Integrating DB2® Workload  
Management with Operating System  
Workload Management*

*Supplement to Best Practices: Implementing DB2®  
Workload Management*

**IBM**



---

## Authors

**Francis Wong**

*DB2<sup>®</sup> Developer*

*Process Model Architecture*

*IBM Toronto Laboratory*

**Paul Bird**

*Senior Technical Staff Member*

*Optim & DB2 Development*

*IBM Toronto Laboratory*

**Rimas P. Kalesnykas**

*DB2 Information Developer*

*IBM Toronto Laboratory*



---

# Contents

<b>Introduction</b> . . . . .	<b>1</b>	<b>Integrating Linux WLM and DB2 workload management</b> . . . . .	<b>17</b>
<b>Planning</b> . . . . .	<b>3</b>	Prerequisites . . . . .	17
<b>Integrating AIX WLM and DB2 workload management.</b> . . . . .	<b>9</b>	Implementation . . . . .	17
Prerequisites . . . . .	9	Monitoring CPU consumption . . . . .	22
Implementation . . . . .	9	Frequently asked questions . . . . .	22
Monitoring CPU consumption . . . . .	15	Linux WLM further reading . . . . .	22
Frequently asked questions . . . . .	15	<b>Index</b> . . . . .	<b>25</b>
AIX WLM further reading . . . . .	16	<b>Notices</b> . . . . .	<b>27</b>
AIX WLM commands summary . . . . .	16		



---

## Introduction

DB2 for Linux, UNIX, and Windows Version 9.7 provides the option of tightly integrating DB2 workload management capabilities with the workload management capabilities provided by the AIX® or Linux operating systems. Specifically, you have the option of integrating DB2 service classes with the equivalent concept in AIX Workload Manager (AIX WLM) and Linux workload management (Linux WLM) in order to take advantage of the native control and monitoring capabilities that these products provide.

The provision of an external *tag*, or outbound correlator, as part of a DB2 service class definition enables the DB2 database manager to interact with the workload management interfaces provided as part of the operating system (OS) and to automatically associate each DB2 agent, which is an operating system thread, with a specific operating system workload management (OS WLM) entity. After this task is done, then the rules and limits defined for that OS WLM entity are then imposed on the DB2 agent by the OS.

As a general rule, OS WLM products provide two types of controls: *hard* and *soft*. *Hard* controls specify a value or allocation of a specific resource which is invariant regardless of the degree of utilization of that resource. *Soft* controls specify a value or allocation of a specific resource which is enforced only when the resource is highly used, otherwise they are not enforced.

AIX WLM supports the definition of a service superclass or service subclass in a two level, parent-child relationship similar to that provided by DB2 service superclasses and subclasses. AIX WLM provides for the control of CPU resource at the AIX WLM service class level in various forms, including *soft* and *hard* settings.

Linux WLM supports the definition of control groups, or cgroups, through which the administrator can control resource consumption. Cgroups can be defined in an unlimited parent-child hierarchical structure to each other which exceeds the two level structure provided by DB2 service superclasses and subclasses. Linux WLM provides for the control of CPU resource at the cgroup level through a *soft* setting.

The typical use of OS WLM integration seen to date in customer implementations is to supplement the native controls within DB2 for Linux, UNIX, and Windows, such as concurrency thresholds, to better protect high priority work from the resource demands of lower priority work. Protecting high priority work is done by *capping* the CPU resource available to the low priority work. The vast majority of these implementations used AIX WLM *hard* controls, with a smaller subset using *soft* controls when the system is highly utilized.





---

## Planning

This section outlines the different areas that need to be considered as part of any OS WLM implementation which includes deciding on what OS WLM counterparts to the DB2 service classes will exist and what controls will be placed upon them. There are also some organizational tasks that are best thought about beforehand in order to simplify the overall process of implementing OS WLM integration.

For the sake of simplicity, we assume that you have completed the design and implementation of your DB2 workload management configuration, at least to the extent of having defined the required DB2 service classes.

### **Designing the mapping scheme between DB2 workload management and OS WLM**

The focus of this section is to define the association, or mapping, between DB2 service classes and an AIX WLM service class or Linux WLM cgroup. This mapping is done by providing an outbound correlator value, representing the external OS WLM entity, to each DB2 service class.

### **Determining the OS WLM configuration**

Before we can do that mapping work, we must first consider the actual configuration to be used when creating the AIX WLM service classes or Linux cgroups that will be associated and used by the DB2 service classes. The layout of the OS WLM entities is a separate and independent activity from that of defining the DB2 workload management configuration and its DB2 service class definitions. There is a degree of flexibility available when defining the OS WLM configuration since any differences can be hidden from the DB2 database manager through the abstraction of mapping.

Although there are a number of options available because of this abstraction, it is still best to keep things simple and as clear as possible. The following best practices guidance is provided for this exercise:

- Always keep a 1:1 mapping between a DB2 service class and an OS WLM entity even if they are not of the same hierarchical level. For example, a DB2 service superclass mapping to an AIX service subclass.
  - This mapping maintains clean, concise control in the OS WLM layer because any controls will be acting upon a very specific and well-defined subset of executing work within the DB2 database manager: the work in that DB2 service class.

For the rest of this section, we assume the following example scenario: We have an inventory database, as depicted in Figure 1 on page 4, with the following DB2 service classes defined:

- Two superclasses: Research and Operations
- Two subclasses under the Operations superclass: Sales and Accounting
- The work in the Sales service subclass is critical to the business and has explicit service level agreements defined on response times.

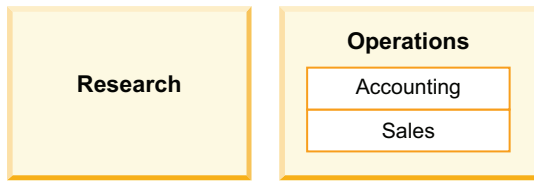


Figure 1. Example inventory database

The three most common OS WLM configurations seen in the field are presented here along with the rationale for why one implementation might be preferred over any other.

The simplest implementation for OS WLM is an identical configuration to that implemented within DB2 with the DB2 service classes, such as shown in Figure 2. This configuration is best suited to a dedicated database server environment where each DB2 service class is to be controlled independently of all others, or where the parent-child relationship between DB2 service classes is to be preserved in their OS WLM counterparts.

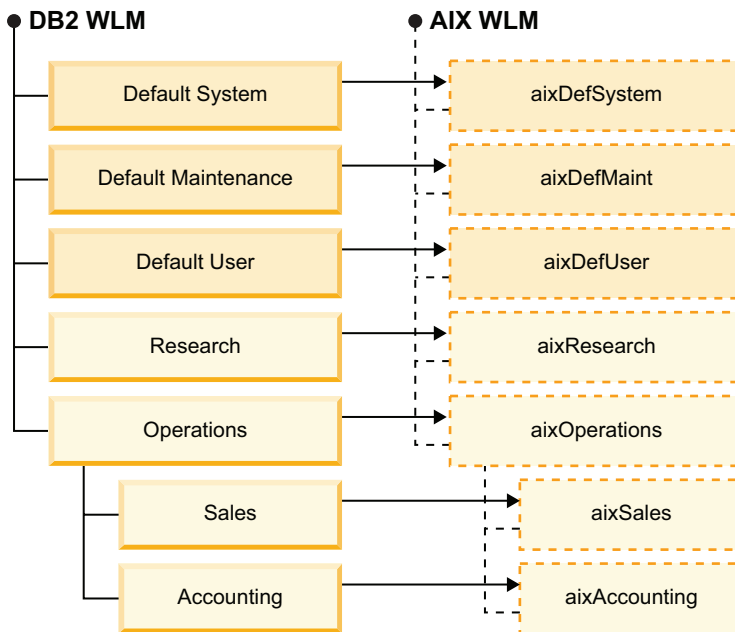


Figure 2. 1:1 service class mapping scheme

Another alternative is to flatten out the OS WLM configuration so that it does not represent any of the hierarchy that is shown by the DB2 service classes. Each DB2 service class, regardless of whether it is a superclass or subclass, is represented by a single OS WLM entity. The hierarchical DB2 service class representation is *flattened* into a series of identical OS WLM entities that are all at the same level. This concept is shown in Figure 3 on page 5.

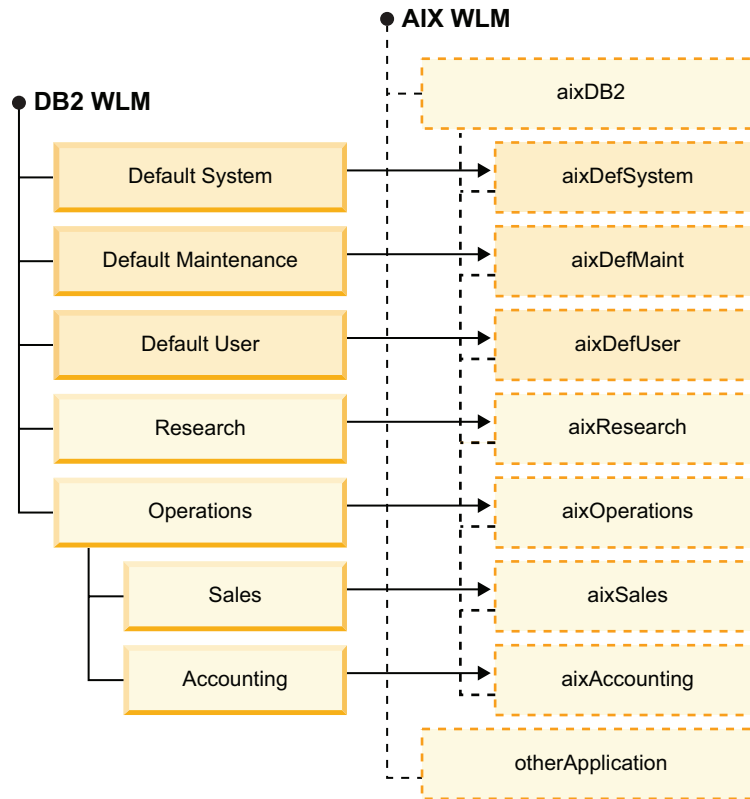


Figure 3. Flat service class mapping scheme

This flat service class mapping configuration is sometimes seen when more than one DB2 database using workload management is present on the same machine. In that environment, each DB2 database is represented as a whole by an OS WLM parent entity (in Figure 3, represented by the aixDB2 AIX WLM service superclass), while the individual DB2 service classes (of all types) are represented under that parent by OS WLM child entities (in Figure 3, all of the DB2 service superclasses and subclasses are mapped to individual AIX WLM service subclasses under the aixDB2 AIX WLM service superclass). In this way, the two independent databases can both be managed.

Finally, an OS WLM configuration can be designed in such a way that it brings work together under the same set of OS WLM controls even though there is no relationship between the DB2 service classes themselves. Figure 4 on page 6 shows such an implementation where the high priority SALES service class is mapped to an AIX WLM service superclass of its own while all of the other lower priority DB2 service classes are mapped to AIX WLM service subclasses under a generic AIX WLM service superclass.

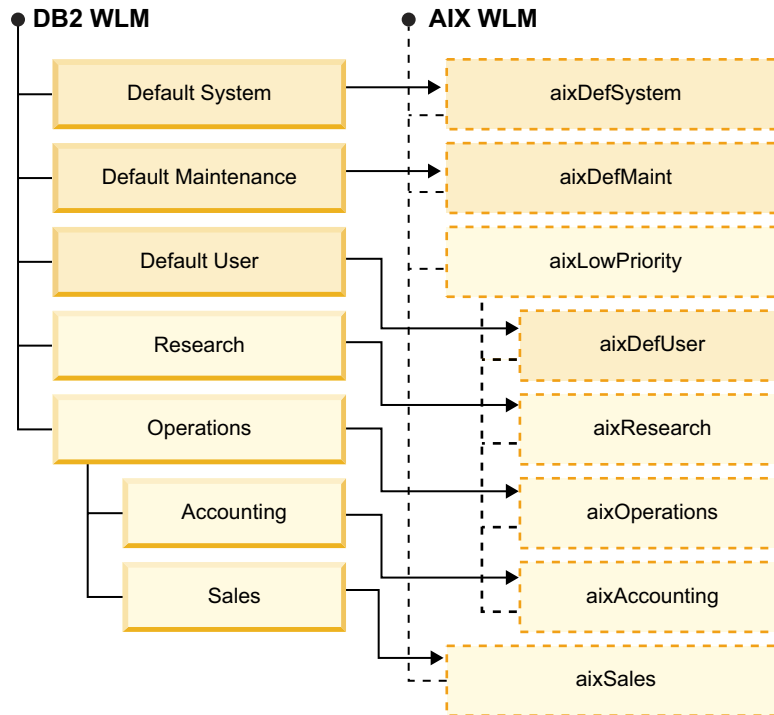


Figure 4. Alternative service class mapping scheme

A common motivation for this type of configuration is the desire to put a common restriction, or limit, on the CPU consumed by low priority work in environments in which there are multiple DB2 service superclasses, each containing at least some portion of the overall low priority work. By mapping all of the low priority work to a common service class, it is possible to put a single common limit in place which affects all of it. For example, a hard CPU limit of 75% could be put on the `aixLowPriority` AIX WLM service superclass in Figure 4 which would preserve 25% of the system, at minimum, for use by work executing in the other AIX WLM service classes.

## Defining outbound correlators

After you have determined what your OS WLM configuration will look like, then you need to define the *tags*, or outbound correlators, to uniquely identify each OS WLM entity so that it can be referenced by the DB2 database manager during the automatic association processing that occurs when DB2 agents are assigned to work in different DB2 service classes. After being defined, these *tags* will be added to the appropriate DB2 service class definitions in order to lay down the desired DB2 workload management to OS WLM relationship.

For Linux WLM, the *tags* are the actual name of the cgroup. For AIX WLM, you must define a unique application tag for each AIX WLM service class (details are provided in a later section).

## Deciding on OS WLM controls

Although the addition of an OS WLM configuration to a DB2 workload management configuration is a useful addition in terms of monitoring, the primary reason that the OS WLM is used with DB2 for Linux, UNIX, and Windows Version 9.7 is to provide an additional, explicit level of control over CPU consumption. Even though each OS WLM provides a different set of capabilities and might

enhance these capabilities over time, the following best practices guidance is provided based on existing field experience:

- Use DB2 workload management concurrency controls (where appropriate) as the initial control on resource consumption and then supplement as needed with OS WLM controls
  - Restricting access to CPU can actually worsen other aspects of system performance because restricted work will run longer, potentially pushing the total concurrency and memory usage higher in the system.
  - Concurrency controls, such as the DB2 `CONCURRENTDBCOORDACTIVITIES` threshold, prevent excessive amounts of work from starting execution which reduces overall resource use.
- Use OS WLM to control only CPU access and not other resources such as I/O or memory
  - Field experience has been solely with CPU access control and DB2 internals do not lend themselves easily to I/O or memory control by OS WLM.
- Always ensure that the DB2 `SYSDEFAULTSYSTEMCLASS` service class for system processing is given equal or greater access to CPU resources than any other service class and never limit its CPU resources (that is, do not *cap* it)
  - Any impact on the work in this important service class will affect all other work being processed within the DB2 database manager because it provides key services such as logging, request processing, buffer pool services, and other services.
- As a rule, do not try to directly accelerate or enhance CPU access for high priority work. Instead, achieve this goal indirectly by restricting the CPU access for lower priority work
  - Experience shows that it is easier to achieve more predictable results by restricting CPU access.
  - By restricting only the lower priority work, high priority work is able to borrow unused resources from the entire system while being guaranteed the amount reserved by the restriction. For example, a system with low priority work that is capped at 75% leaves the high priority work with 25% of the CPU resources, but the high priority work can access more than that (if available) because it is not restricted.
- Use *hard* controls, where available on systems, which are not highly utilized if consistent control is desired
  - *Soft* controls, that are implemented by the OS WLM, are only enforced when the overall system is highly utilized (for example, >80+%) with the result that they are of little value in situations when the system is under-utilized as a whole.
- We do not recommend applying *hard* controls in a Micro-Partitioning® (shared LPAR) environment
  - In situations where the available CPU capacity for the LPAR is not fully consumed, the Hypervisor reduces the CPU available to the LPAR. With the use of *hard* controls, which prevent restricted work from fully using the available CPU capacity, the absence of sufficient unrestricted work to consume these resources can lead to a feedback loop in which the hypervisor continuously reduces the CPU available to the LPAR until it is running at its minimum configured capacity.

## Organizational tasks

In addition to the technical aspects of implementing an OS WLM solution to complement your DB2 workload management configuration, there are several organizational tasks which must also be contemplated ahead of time to ensure that they do not create an obstacle to the actual implementation itself. This section outlines a few specific items to consider early in the process to help ensure a successful deployment.

First and foremost, experience often shows that the biggest obstacle to a successful OS WLM implementation exists in the lack of necessary communication and cooperation between the database administrators and system administrators. System administrators, in many companies, are a separate group and are often the only ones with the required access needed to implement OS WLM on the machines. In order to implement an OS WLM configuration, early system administrator involvement in the overall process is critical for success.

The next aspect that needs to be considered, after the OS WLM configuration is determined, is the deployment of that configuration. In a partitioned database environment, there are multiple physical and logical machines, involved as part of that environment, which need to be modified to reflect the desired OS WLM configuration. Planning for OS WLM deployment must take into account that all of the machines must be updated with the identical definitions in a coordinated manner, the relevant OS WLM engaged on each platform, and the DB2 database manager itself restarted (after the OS WLM has been enabled)<sup>1</sup>. Any future changes or modifications to the OS WLM configuration or controls also need to have their deployment coordinated across all members that encompass the DB2 database server.

Finally, the last consideration is when to deploy the OS WLM configuration. Experience shows that it is rarely possible to rush an OS WLM implementation into an environment during peak season in order to address an emergency situation. Having the OS WLM configuration deployed, but in a passive mode without any CPU controls in place, gives you the flexibility to implement such controls very quickly, if they are needed. The passive mode starts the automatic association between the DB2 database manager and the OS WLM product without exerting any control over CPU access.

With AIX WLM, you can deploy a workload management configuration in a passive mode without any CPU controls specified. However, Linux WLM does not provide a completely passive configuration because you must specify shares for each control group specified. By specifying that each control group in the configuration has the same number of shares as the root group (1024), this specification gives all of the groups equal access to the CPU. Depending on the distribution and timing of the work on the database, this type of configuration behaves slightly differently than a system without any Linux WLM configuration.

If you are using the AIX platform and you have intentions of eventually using AIX WLM, then it is often better to deploy the AIX WLM configuration in a passive mode from the beginning. This approach is often beneficial when you already know the shape of your desired AIX WLM configuration because it makes activation a much faster exercise and allows early validation of your AIX WLM design.

---

1. Note that the last two steps, enabling OS WLM and restarting the DB2 database manager, are required only once per machine when first starting.

---

## Integrating AIX WLM and DB2 workload management

Integrating AIX WLM with DB2 workload management results in the mapping of DB2 threads into AIX WLM service classes through the use of application tags.

### Controlling CPU resources

The following is a list of the several ways to control CPU usage between service classes:

- **Share:** a soft, relative setting between the AIX WLM service classes. When there is CPU available on the system, a AIX WLM service class can use more than its CPU share.
- **Limit:** a maximum percentage of the available CPU of the system. A limit can be hard or soft.
- **Tier:** an ordering with which CPU is allocated to the AIX WLM service classes. An AIX WLM service class in tier 0 gets CPU allocation before an AIX WLM service class in tier 1, and so on.
- For more information about shares, limits, and tiers, refer to the “pSeries® and AIX Information Center”.

---

### Prerequisites

The following is a list of prerequisites for the best practices integration of AIX WLM with DB2 workload management:

- DB2 Version 9.5, and above
- AIX 5.3H (5300-05), or later

**Note:** For AIX 6.100-06 systems, apply the following APAR: IZ91305 WLM CRASHES SYSTEM WHEN CLASS PAGE COUNT BECOMES < 0 APPLIES TO AIX 6100-06 (PMR 74562,499).

- Root authority

---

### Implementation

Before the actual implementation, you need to do enough planning to identify the desired AIX WLM configuration, how the DB2 workload management and AIX WLM configurations will relate to each other, and what CPU controls you will be applying within the AIX WLM configuration. These topics are discussed in the Planning section.

This section highlights the key tasks needed to implement an AIX WLM configuration after completing the planning phase.

#### Determine application tags

As the association between DB2 service classes and AIX WLM service classes is done through the use of application tags, it is best to spend some time identifying the tags to be used to represent each AIX WLM service class before proceeding with the actual implementation.

Table 1 lists the example application tag determined for each AIX WLM service class.

Table 1. Examples of a unique application tag for each AIX WLM service class

AIX WLM service class	Application tag
aixDefSystem	_DefSystem
aixDefMaint	_DefMaint
aixDefUser	_DefUser
aixResearch	_Research
aixOperations	_Operations
aixSales	_Operations.Sales
aixAccounting	_Operations.Accounting

## Create AIX WLM configuration

This task shows how to create a unique AIX WLM configuration for your database based on a provided AIX WLM template.

1. Create an AIX WLM configuration by copying the `/etc/wlm/template` directory to a new directory by running the highlighted command:

```
(root@combop511) /etc/wlm
$ ls
.lock      .running/  current@   standard/  template/

(root@combop511) /etc/wlm
$ cp -r template inventory

(root@combop511) /etc/wlm
$ ls
.lock      .running/  current@   inventory/  standard/  template/
```

Set the new directory as current by running the following command:

```
(root@combop511) /etc/wlm
$ wlmcntrl -d inventory
```

**Note:** You can create multiple configuration directories and switch between them using the preceding command. Configuration directories are also portable across AIX machines.

## Create AIX WLM classes

In this task, we create new AIX WLM service classes within the new AIX WLM configuration created in the previous step.

1. Create AIX WLM service classes. Set the desired CPU setting for each AIX WLM service class. Shares are used in this example.
2. Ensure that the DB2 default system service class gets the highest CPU setting.
3. Ensure that inheritance is OFF.
4. To meet the previous objectives, run the following commands:

```
mkclass -a inheritance=no -c shares=100 aixDefSystem
mkclass -a inheritance=no -c shares=20 aixDefMaint
mkclass -a inheritance=no -c shares=30 aixDefUser
mkclass -a inheritance=no -c shares=30 aixResearch
mkclass -a inheritance=no -c shares=70 aixOperations
mkclass -a inheritance=no -c shares=80 aixOperations.aixSales
mkclass -a inheritance=no -c shares=20 aixOperations.aixAccounting
```



5. Review AIX WLM service class settings. To review the AIX WLM service class settings made by running the preceding commands, run the following highlighted command:

```
(root@combop511) /etc/wlm  
$ lsclass -f -r
```

The following is a sample output resulting from the preceding command:

```
System:  
    memorymin = 1  
  
Default:  
  
Shared:  
  
aixDefMaint:  
    inheritance = "no"  
    CPUshares = 20  
  
aixDefSystem:  
    inheritance = "no"  
    CPUshares = 100  
  
aixDefUser:  
    inheritance = "no"  
    CPUshares = 30  
    .  
    .  
    .  
aixResearch:  
    inheritance = "no"  
    CPUshares = 30  
  
aixOperations:  
    inheritance = "no"  
    CPUshares = 70  
  
aixOperations.Default:  
  
aixOperations.Shared:  
  
aixOperations.aixSales:  
    inheritance = "no"  
    CPUshares = 80  
  
aixOperations.aixAccounting:  
    inheritance = "no"  
    CPUshares = 20
```

## Edit AIX WLM mapping rules files

This task associates the application tags that were defined earlier with the appropriate AIX WLM service classes.

1. Edit the `/etc/wlm/inventory/rules` mapping rules file to associate the identified application tags with the appropriate AIX WLM service class. The mapping rules file is evaluated from top to bottom. After a rule is matched, the thread/process maps to the AIX class associated with that rule. The Default class is for all threads/processes that do not fall into any other class. This Default class is the catch-all class and should be left at the bottom of the mapping rules file. The highlighted lines in the following example show the application tags being associated with the AIX WLM service classes:

```

* IBM_PROLOG_BEGIN_TAG
* This is an automatically generated prolog.
*
* bos530 src/bos/etc/wlm/rules 1.2
*
* Licensed Materials - Property of IBM
*
* (C) COPYRIGHT International Business Machines Corp. 1999,2002
* All Rights Reserved
*
* US Government Users Restricted Rights - Use, duplication or
* disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*
* IBM_PROLOG_END_TAG
* class resvd user group application type tag
System - root - - - - -
aixDefSystem - - - - -_DefSystem
aixDefMaint - - - - -_DefMaint
aixDefUser - - - - -_DefUser
aixResearch - - - - -_Research
aixOperations - - - - -_Operations
aixOperations - - - - -_Operations.Sales
aixOperations - - - - -_Operations.Accounting
Default - - - - -

```

2. Edit the /etc/wlm/inventory/aixOperations/rules AIX WLM service subclass level mapping rules file for the aixOperations superclass to correspond with the following highlighted lines:

```

* class resvd user group application type tag
aixSales - - - - -_Operations.Sales
aixAccounting - - - - -_Operations.Accounting

```

## Start AIX WLM

Now that we have created our AIX WLM configuration, we must enable AIX WLM so that it is available to interact with the DB2 database manager.

- To turn ON AIX WLM, run the following command:

```
wlmcntrl -a
```

After AIX WLM is enabled, you must restart the DB2 database manager in order for it to establish the relationship between AIX WLM and itself. Note that AIX WLM needs to be enabled only once, after which it stays enabled until turned OFF.

AIX WLM and DB2 workload management configuration settings can be altered on-the-fly and take effect in real time.

- To update AIX WLM, run the following command:

```
wlmcntrl -u
```

An update is needed when any AIX WLM setting is altered.

- To turn OFF AIX WLM, run the following command:

```
wlmcntrl -o
```

For more information about the **wlmcntrl** command, run the following command:

```
man wlmcntrl
```

## Alter DB2 default service classes to set outbound correlators

The final step is to inform the DB2 database manager about the application tags to be used in its communications with AIX WLM. This step is done by altering the

DB2 service class definitions to specify the AIX WLM application tag as the outbound correlator for the appropriate DB2 service class.

In our example scenario, the following DDL statements set the outbound correlators:

```
ALTER SERVICE CLASS SYSDEFAULTSYSTEMCLASS OUTBOUND CORRELATOR '_DefSystem'
ALTER SERVICE CLASS SYSDEFAULTMAINTENANCECLASS OUTBOUND CORRELATOR '_DefMaint'
ALTER SERVICE CLASS SYSDEFAULTUSERCLASS OUTBOUND CORRELATOR '_DefUser'
ALTER SERVICE CLASS RESEARCH OUTBOUND CORRELATOR '_Research'
ALTER SERVICE CLASS OPERATIONS OUTBOUND CORRELATOR '_Operations'
ALTER SERVICE CLASS SALES UNDER OPERATIONS OUTBOUND CORRELATOR '_Operations.Sales'
ALTER SERVICE CLASS ACCOUNTING UNDER OPERATIONS OUTBOUND CORRELATOR
'_Operations.Accounting'
```

## Check the DB2 threads mapping

The last implementation task is to validate that the DB2 threads are being mapped to the correct AIX WLM service class.

The mapping of DB2 agents to DB2 service classes can be acquired from the WLM\_GET\_SERVICE\_CLASS\_AGENTS\_V97 table function with the following SQL statement example:

```
SELECT SERVICE_SUPERCLASS_NAME,
       SERVICE_SUBCLASS_NAME,
       AGENT_TID
FROM TABLE(WLM_GET_SERVICE_CLASS_AGENTS_V97(NULL, NULL, NULL, -2))
```

You can use the output from this statement, combined with output from the **db2pd** and the **ps** commands, to validate that the relationships between DB2 service classes and AIX WLM service classes are as you expected.

To validate the relationships, use the output of the **db2pd -edu** command to determine the kernel thread ID for each DB2 agent; the AGENT\_TID column in the result obtained from the preceding SQL statement corresponds to the EDU ID column in the **db2pd** output. With this information, you can now use the output of the **ps** command to determine the mapping of the individual threads to the AIX WLM service classes.

To check the mapping of DB2 threads to AIX WLM service classes, run the following **ps** command, where *process\_id* is the process ID of the db2sysc process for the database you want to validate:

```
ps -m -o THREAD,class -p process_id
```

Be aware that DB2 agents can change service classes as they process different requests and statements.

The following is a sample output resulting from the preceding command:

USER	PID	PPID	TID	ST	CP	PRI	SC	WCHAN	F	TT	BND	COMMAND	CLASS
danforth	893436	3023202	-	A	0	60	29	*	40401	-	-	db2sysc	Default
-	-	-	745657	S	0	60	1	f1000004e0aaf200	410400	-	-	-	-
-	-	-	905381	S	0	60	1	-	400400	-	-	-	-
-	-	-	1507407	S	0	60	1	-	400400	-	-	-	-
-	-	-	2232555	S	0	60	1	f100011064feee30	410400	-	-	-	-
-	-	-	3764465	S	0	156	1	f1000110605b3018	410400	-	-	-	aixDefUser
-	-	-	3829877	S	0	60	1	f10001303ffc030	410400	-	-	-	-
-	-	-	6180907	S	0	60	1	-	400400	-	-	-	aixDefSystem
-	-	-	7020623	S	0	60	1	-	400400	-	-	-	-
-	-	-	7975081	S	0	120	1	f10001106051e318	410400	-	-	-	aixOperations.aixSales

```

- - - 9904249 S 0 60 1 - 400400 - - - -
- - - 172433 S 0 60 1 f1000004e013e000 410400 - - - -
- - - 495979 S 0 60 1 - 418400 - - - -
- - - 696747 S 0 60 1 f10001304022a2b0 410400 - - - -
- - - 3617239 S 0 60 1 - 400400 - - - aixDefSystem
- - - 5247431 S 0 156 1 f100011011ce2918 410400 - - - aixDefUser
- - - 5829033 S 0 60 1 - 400400 - - - -
- - - 823947 S 0 60 1 - 400400 - - - -
- - - 1364487 S 0 60 1 - 400400 - - - -
- - - 2663143 S 0 60 1 - 400400 - - - -
- - - 3777147 S 0 60 1 f1000004e011be00 410400 - - - aixDefSystem
- - - 4514547 S 0 60 1 - 400400 - - - -
- - - 8323597 S 0 60 1 - 400400 - - - -
- - - 1409969 S 0 60 1 - 400400 - - - -
- - - 1651617 S 0 60 1 - 400400 - - - -
- - - 3204051 S 0 60 1 - 400400 - - - -
- - - 3539935 S 0 60 1 f1000004e094fa00 410400 - - - aixDefSystem
- - - 4883387 S 0 60 1 - 400400 - - - -
- - - 6271899 S 0 60 1 - 400400 - - - -

```

## Corresponding db2pd command output

To obtain the corresponding **db2pd** command output, run the following command:

```
db2pd -edu
```

The following is a sample output resulting from the preceding command:

```

EDU
ID      TID          Kernel TID      EDU Name
=====
6941    6941          7975081         db2agent (SAMPLE)
6684    6684          3764465         db2agent (SAMPLE)
6427    6427          4514547         db2evmli (DB2DETAILDEADLOCK)
6170    6170          905381          db2w1md (SAMPLE)
5913    5913          3204051         db2taskd (SAMPLE)
5656    5656          8323597         db2stmm (SAMPLE)
5399    5399          745657          db2pfchr (SAMPLE)
5142    5142          3539935         db2pfchr (SAMPLE)
4885    4885          3777147         db2pfchr (SAMPLE)
4628    4628          4883387         db2pc1nr (SAMPLE)
4371    4371          823947          db2pc1nr (SAMPLE)
4114    4114          7020623         db2pc1nr (SAMPLE)
3857    3857          5829033         db2pc1nr (SAMPLE)
3600    3600          6271899         db2pc1nr (SAMPLE)
3343    3343          1364487         db2pc1nr (SAMPLE)
3086    3086          9904249         db2pc1nr (SAMPLE)
2829    2829          3617239         db2d1ock (SAMPLE)
2572    2572          1409969         db21fr (SAMPLE)
2315    2315          2663143         db21oggw (SAMPLE)

```

2058	2058	6180907	db2loggr (SAMPLE)
1801	1801	5247431	db2agent (SAMPLE)
1543	1543	1507407	db2resync
1286	1286	172433	db2ipccm
1029	1029	2232555	db2licc
772	772	1651617	db2thc1n
515	515	696747	db2aiiothr
2	2	3829877	db2alarm
258	258	495979	db2sysc

---

## Monitoring CPU consumption

The most convenient tool for monitoring AIX WLM CPU consumption is the **wlmstat** command. The following is a sample output resulting from the **wlmstat** command:

```
/etc/wlm # wlmstat
          CLASS CPU MEM DKIO
Unclassified  0 25  0
  Unmanaged   2 19  0
    Default  13 11  0
      Shared   0  1  0
        System  2  7  0
          aixDefMaint  0  0  0
          aixDefSystem  0  0  0
          aixDefUser   0  0  0
          aixResearch  0  0  0
          aixOperations  0  0  0
aixOperations.Default - - -
aixOperations.Shared - - -
aixOperations.aixSales - - -
aixOperations.aixAccounting - - -
          test    0  0  0
          TOTAL  15 44  0
```

---

## Frequently asked questions

**Why are all DB2 threads not mapped to AIX WLM service classes?**

Agents and system entities are only mapped when they begin work.

**Why is a DB2 thread still mapped to an AIX WLM service class even after it leaves its DB2 service class?**

We do not unmap a DB2 thread from its AIX WLM service class when it leaves its DB2 service class. The thread remaps to a new AIX WLM service class when it begins work in a new DB2 service class. This remapping is done for performance reasons.

**I see "1495-589 \_wlm\_get\_thrinfo: A system call received a parameter that is not valid." errors while running the ps command. What are they?**

These errors are harmless. You will see them when the **ps** command displays canceled or terminated threads (state = Z).

---

## AIX WLM further reading

Additional information about AIX Workload Manager can be found in the references provided here.

### Workload manager

<http://publib.boulder.ibm.com/infocenter/aix/v6r1/topic/com.ibm.aix.baseadm/doc/baseadmndita/wlm.htm>

### Mapping rules: Workload Manager rules File

<http://pic.dhe.ibm.com/infocenter/aix/v7r1/topic/com.ibm.aix.files/doc/aixfiles/rules.htm>

### Micro-Partitioning

[http://publib.boulder.ibm.com/infocenter/pseries/v5r3/topic/com.ibm.aix.prftungd/doc/prftungd/micro\\_part.htm](http://publib.boulder.ibm.com/infocenter/pseries/v5r3/topic/com.ibm.aix.prftungd/doc/prftungd/micro_part.htm)

## AIX WLM commands summary

The following is a summary of AIX WLM commands:

wlmcntrl

-----

- Starts, stops, updates and queries AIX WLM
- To start AIX WLM in active mode, issue "wlmcntrl -a"
- To update AIX WLM, issue "wlmcntrl -u"
- To stop AIX WLM, issue "wlmcntrl -o"
- To query state of AIX WLM, issue "wlmcntrl -q"

mkclass

-----

- Creates an AIX WLM class

chclass

-----

- Changes attributes for an AIX WLM class

rmclass

-----

- Removes an AIX WLM class
- Usage: "rmclass [-S <superclass>] <class name>"

lsclass

-----

- List current AIX WLM classes configuration
- To see all classes, issue "lsclass -f -r"

ps

--

- To see the AIX WLM service class associated with each thread that is associated with the DB2 server process, issue the following:

```
ps -m -o THREAD,class -p <pid of db2sysc>
```

---

## Integrating Linux WLM and DB2 workload management

Integrating Linux WLM with DB2 workload management results in the mapping of DB2 threads into Linux WLM control groups, also referred to as cgroups.

### Controlling CPU resources

To control CPU usage between cgroups, you can use the following method:

- Shares: a soft, relative setting between cgroups. When there is CPU available on the system, a cgroup can use more than its CPU share.

Tasks that are not mapped to any user defined cgroup run in the root cgroup. The default CPU shares value of this cgroup is 1024.

---

### Prerequisites

The following is a list of prerequisites for the best practices integration of Linux WLM with DB2 workload management:

- DB2 Version 9.5, and above
- SUSE Linux Enterprise Server (SLES) 11, and Red Hat Enterprise Linux (RHEL) 6, or later
- Root authority
- Installed libcgroup package

---

### Implementation

Before the actual implementation, you need to do enough planning to identify the desired Linux WLM configuration, how the DB2 workload management and Linux WLM configurations will relate to each other, and what CPU controls you will be applying within the Linux WLM configuration. These topics are discussed in the Planning section.

This section highlights the key tasks needed to implement a Linux WLM configuration after completing the planning phase.

#### Determine cgroups

As the association between DB2 service classes and Linux WLM service classes is done through the use of cgroups, it is best to spend some time identifying the cgroups to be used to represent each Linux WLM service class before proceeding with the actual implementation.

Table 1 on page 10 lists the example cgroup determined for each DB2 service class.

*Table 2. Examples of a unique Linux WLM cgroup for each DB2 service class*

DB2 service class	Linux WLM service class (cgroup)
Default System	InxDefSystem
Default Maintenance	InxDefMaintenance
Default User	InxDefUser
Research	InxResearch

Table 2. Examples of a unique Linux WLM cgroup for each DB2 service class (continued)

DB2 service class	Linux WLM service class (cgroup)
Operations	InxOperations
Sales	InxSales
Accounting	InxAccounting

## Edit Linux WLM configuration file

This task shows how to create a unique Linux WLM configuration for your database within the standard Linux WLM configuration file.

The Linux WLM `/etc/cgconfig.conf` configuration file contains 2 main sections:

**mount** Specifies which resource subsystems you want to control with your cgroups.

**group** Specifies a cgroup.

For DB2 OS WLM integration, use the **cpu** subsystem to control CPU consumption and the **cpuacct** subsystem to account for CPU consumption.

Modify the Linux WLM `/etc/cgconfig.conf` configuration file to introduce the required new cgroups. Assign the appropriate shares to each cgroup, but ensure that the cgroup for the DB2 default system service class gets the highest CPU shares.

Keep in mind that all shares allocated to any newly defined cgroups are relative to the default number of shares automatically assigned to the root cgroup (1024 shares).

The following example illustrates the modifications that are required for our example scenario:

```
# List the resource subsystems to use for
# this configuration
mount {
    cpu = /cgroup/cpu;
    cpuacct = /cgroup/cpu;
}

# Define cgroup for DB2 Default System class
group InxDefSystem {

    # Permission properties
    perm {

        # Specifies whose threads can run in
        # this cgroup
        task {
            uid = db2inst1;
            gid = db2iadm1;
        }

        # Specifies who can edit this cgroup's
        # properties
        admin {
            uid = db2inst1;
            gid = db2iadm1;
        }
    }
}
```



```

# Relative CPU shares setting compared
# to other cgroups
cpu {
    cpu.shares = 2048;
}
}

# Define cgroup for DB2 Default Maintenance class
group lnxDefMaint {

    perm {
        task {
            uid = db2inst1;
            gid = db2iadm1;
        }
        admin {
            uid = db2inst1;
            gid = db2iadm1;
        }
    }
    cpu {
        cpu.shares = 512;
    }
}

# Define cgroup for DB2 Default User class
group lnxDefUser {

    perm {
        task {
            uid = db2inst1;
            gid = db2iadm1;
        }
        admin {
            uid = db2inst1;
            gid = db2iadm1;
        }
    }
    cpu {
        cpu.shares = 1024;
    }
}

# Define cgroup for Research class
group lnxResearch {

    perm {
        task {
            uid = db2inst1;
            gid = db2iadm1;
        }
        admin {
            uid = db2inst1;
            gid = db2iadm1;
        }
    }
    cpu {
        cpu.shares = 512;
    }
}

# Define cgroup for Operations class
group lnxOperations {

    perm {
        task {

```

```

        uid = db2inst1;
        gid = db2iadm1;
    }
    admin {
        uid = db2inst1;
        gid = db2iadm1;
    }
}
cpu {
    cpu.shares = 2048;
}
}

# Define cgroup for Sales subclass under Operations
group lnxOperations/lnxSales {
    perm {
        task {
            uid = db2inst1;
            gid = db2iadm1;
        }
        admin {
            uid = db2inst1;
            gid = db2iadm1;
        }
    }
}

# Shares based on CPU available to lnxOperations
# parent cgroup, relative to sibling cgroups
cpu {
    cpu.shares = 2048;
}
}

# Define cgroup for Accounting subclass under Operations
group lnxOperations/lnxAccounting {
    perm {
        task {
            uid = db2inst1;
            gid = db2iadm1;
        }
        admin {
            uid = db2inst1;
            gid = db2iadm1;
        }
    }
}

# Shares based on CPU available to lnxOperations
# parent cgroup, relative to sibling cgroups
cpu {
    cpu.shares = 1024;
}
}

```

## Start Linux WLM

After we have created our Linux WLM configuration, we must enable Linux WLM so that it is available to interact with the DB2 database manager.

- To turn ON Linux WLM, run the following command:  

```
service cgconfig start
```

After Linux WLM is enabled, you must restart the DB2 database manager in order for it to establish the relationship between Linux WLM and itself. Note that Linux WLM needs to be enabled only once, after which it stays active until turned OFF.

Although DB2 workload management configuration settings can be altered on-the-fly and take effect in real time, Linux WLM changes require that you update the `/etc/cgconfig.conf` configuration file, then run the following command to restart Linux WLM:

```
service cgconfig restart
```

- To turn OFF Linux WLM, run the following command:

```
service cgconfig stop
```

## Alter DB2 default service classes to set outbound correlators

The final step is to inform the DB2 database manager about the cgroups to be used in its communications with Linux WLM. This step is done by altering the DB2 service class definitions to specify the Linux WLM cgroup as the outbound correlator for the appropriate DB2 service class.

In our example scenario, the following DDL statements set the outbound correlators:

```
ALTER SERVICE CLASS SYSDEFAULTSYSTEMCLASS OUTBOUND CORRELATOR 'lnxDefSystem'  
ALTER SERVICE CLASS SYSDEFAULTMAINTENANCECLASS OUTBOUND CORRELATOR 'lnxDefMaint'  
ALTER SERVICE CLASS SYSDEFAULTUSERCLASS OUTBOUND CORRELATOR 'lnxDefUser'  
ALTER SERVICE CLASS RESEARCH OUTBOUND CORRELATOR 'lnxResearch'  
ALTER SERVICE CLASS OPERATIONS OUTBOUND CORRELATOR 'lnxOperations'  
ALTER SERVICE CLASS SALES UNDER OPERATIONS OUTBOUND CORRELATOR  
'lnxOperations/lnxSalesr'  
ALTER SERVICE CLASS ACCOUNTING UNDER OPERATIONS OUTBOUND CORRELATOR  
'lnxOperations/lnxAccounting'
```

## Check the DB2 tasks mapping

To validate that the DB2 threads are being mapped to the correct Linux WLM service class, you need to compare the DB2 service class in which a DB2 agent is running to the Linux WLM cgroup to which the kernel thread representing the agent is assigned to make sure that the relationship is as intended.

To see which kernel threads are running in each cgroup, you need to look at the tasks file of the cgroup and then correspond the task IDs in this file to the kernel TIDs in the output obtained after running the `db2pd -edu` command. The following is an example of how to look at the tasks file:

```
$ cat /cgroup/cpu/lnxResearch/tasks  
30107  
30108
```

After you have mapped the ID in the tasks file to the kernel thread IDs in the **db2pd** output, you can use the EDU ID in the same output to link the thread to a specific DB2 agent and its DB2 service class by using the `WLM_GET_SERVICE_CLASS_AGENTS_V97` table function, as with the following SQL statement example:

```
SELECT SERVICE_SUPERCLASS_NAME,  
       SERVICE_SUBCLASS_NAME,  
       AGENT_TID  
FROM TABLE(WLM_GET_SERVICE_CLASS_AGENTS_V97(NULL, NULL, NULL, -2))
```

The `AGENT_TID` column in the result obtained from the preceding SQL statement corresponds to the EDU ID column in the **db2pd** output and you can now link the DB2 service class to the Linux WLM cgroup.

Be aware that DB2 agents can change service classes, and cgroups, as they process different requests and statements.

## Corresponding db2pd command output

To obtain the corresponding **db2pd** command output, run the following command:  
db2pd -edu

The following is a sample output resulting from the preceding command:

EDU ID	TID	Kernel TID	EDU Name	USR (s)	SYS (s)
43	47334809397568	30132	db2evml i (DB2DETAILDEADLOCK)	0.000000	0.000000
42	47334813591872	30131	db2pcsd (TEST)	0.000000	0.000000
41	47334817786176	30130	db2fw3 (TEST)	0.000000	0.000000
40	47334821980480	30129	db2fw2 (TEST)	0.000000	0.000000
39	47334826174784	30128	db2fw1 (TEST)	0.000000	0.000000
38	47334830369088	30127	db2fw0 (TEST)	0.000000	0.000000
37	47334834563392	30126	db2lused (TEST)	0.000000	0.000000
36	47334838757696	30125	db2w1md (TEST)	0.000000	0.000000
35	47334842952000	30124	db2taskd (TEST)	0.000000	0.000000
34	47334847146304	30123	db2stmm (TEST)	0.020000	0.000000
<b>18</b>	<b>47334914255168</b>	<b>30107</b>	<b>db2agent (TEST)</b>	<b>1.340000</b>	<b>0.880000</b>
<b>44</b>	<b>47334914876292</b>	<b>30108</b>	<b>db2agent (TEST)</b>	<b>1.020000</b>	<b>0.810000</b>
33	47334851340608	30122	db2pfchr (TEST)	0.000000	0.000000
32	47334855534912	30121	db2pfchr (TEST)	0.000000	0.000000
31	47334859729216	30120	db2pfchr (TEST)	0.000000	0.000000
30	47334863923520	30119	db2pc1nr (TEST)	0.000000	0.000000
29	47334868117824	30118	db2pc1nr (TEST)	0.000000	0.000000
28	47334872312128	30117	db2pc1nr (TEST)	0.000000	0.000000
....					

---

## Monitoring CPU consumption

The CPU accounting subsystem provides CPU consumption information. To see the CPU usage of a cgroup, look at the `cpuacct.stat` and `cpuacct.usage` files in the directory of the cgroup.

---

## Frequently asked questions

### Why are all DB2 tasks not mapped to Linux WLM?

Agents and system entities are only mapped when they begin work in a DB2 service class. Only database level agents and system entities are mapped to Linux WLM.

### Why is a DB2 thread still mapped to a Linux WLM cgroup after it leaves its DB2 service class?

We do not unmap a DB2 task from its Linux WLM cgroup when it leaves its DB2 service class. The task remaps to a new Linux WLM cgroup when it begins work in a new DB2 service class. This remapping is done for performance reasons.

---

## Linux WLM further reading

Additional information about the Linux WLM is provided here and can be used for both introductory and supplemental purposes with respect to this best practices paper.

### RHEL 6 Resource Management Guide

[http://docs.redhat.com/docs/en-US/Red\\_Hat\\_Enterprise\\_Linux/6/pdf/Resource\\_Management\\_Guide/Red\\_Hat\\_Enterprise\\_Linux-6-Resource\\_Management\\_Guide-en-US.pdf](http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/pdf/Resource_Management_Guide/Red_Hat_Enterprise_Linux-6-Resource_Management_Guide-en-US.pdf)

**SLES 11 System Analysis and Tuning Guide**

[http://www.suse.com/documentation/sles11/pdfdoc/book\\_sle\\_tuning/  
book\\_sle\\_tuning.pdf](http://www.suse.com/documentation/sles11/pdfdoc/book_sle_tuning/book_sle_tuning.pdf)

**Linux cgroups Documentation**

[/usr/src/linux/Documentation/cgroups/cgroups.txt](#)

**Linux CPU Accounting Subsystem Documentation**

[/usr/src/linux/Documentation/cgroups/cpuacct.txt](#)



---

# Index

## A

AIX WLM  
    commands summary 16

## F

FAQ  
    AIX WLM integration 15  
    Linux WLM integration 22  
further reading  
    AIX WLM 16  
    Linux WLM 22

## I

implementing  
    AIX WLM integration 9  
    Linux WLM integration 17

## M

monitoring CPU consumption  
    AIX WLM integration 15  
    Linux WLM integration 22

## N

notices 27

## P

planning 3  
prerequisites  
    integrating AIX WLM 9  
    integrating Linux WLM 17

## W

workload management best practices  
    monitoring CPU consumption  
        AIX WLM integration 15  
        Linux WLM integration 22  
    supplement  
        AIX WLM commands summary 16  
        AIX WLM further reading 16  
        configuring AIX WLM and DB2 workload management  
            integration 9  
        configuring Linux WLM and DB2 workload  
            management integration 17  
        FAQ for AIX WLM integration 15  
        FAQ for Linux WLM integration 22  
        implementing AIX WLM integration 9  
        implementing Linux WLM integration 17  
        introduction 1  
        Linux WLM further reading 22  
        planning 3  
        prerequisites for AIX WLM integration 9  
        prerequisites for Linux WLM integration 17





---

## Notices

This information was developed for products and services offered in the U.S.A. Information about non-IBM® products is based on information available at the time of first publication of this document and is subject to change.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information about 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.  
1623-14, Shimotsuruma, Yamato-shi  
Kanagawa 242-8502 Japan

**The following paragraph does not apply to the United Kingdom or any other country/region 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, changes, or both in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to websites not owned by IBM are provided for convenience only and do not in any manner serve as an endorsement of those

websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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.

Licenses 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 that has been exchanged, should contact:

IBM Canada Limited  
U59/3600  
3600 Steeles Avenue East  
Markham, Ontario L3R 9Z7  
CANADA

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 may contain 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. 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\_*. All rights reserved.

## Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)<sup>®</sup> are trademarks or registered 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 [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

The following terms are trademarks or registered trademarks of other companies

- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
- Java<sup>™</sup> and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle, its affiliates, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Intel, Intel logo, Intel Inside, Intel Inside logo, Celeron, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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







Printed in USA