

IBM i Performance Tools for Application Developers

Dean May - dean@ibm.com
[@DeanMayCom](https://twitter.com/DeanMayCom)



IBM



Performance Disclaimer

- "It depends"
- Performance information and recommendations in this presentation are based on measurements, analysis, and projections in a controlled environment for specific performance workloads.
- Your results may vary significantly and are dependent on the application and configuration.
- This information is provided along with general recommendations for you to better understand system performance.
- Information is provided "AS IS" without warranty of any kind.

Definitions

- **Interactive work** – Generally 5250 online transaction processing (OLTP)
- **Batch work** – non-interactive workload's
- **Commercial Processing Workload (CPW)** – Workload's which have a relatively large amount of I/O compared to computation
- **Disk arms** – generally one disk arm per disk drive. More disk arms generally results in improved I/O performance
- **Disk capacity** – the amount of disk storage space
- **Memory paging and faulting** – the movement of data in and out of memory
- **Wait Accounting** – the ability to determine what a job is doing when it is not running

Definitions...

- **Measurement** - The collection of performance metrics
- **Transaction** - A basic unit of work
- **Workload** - An application that can drive load on a system
- **Benchmark** - A specific workload with specific environment settings
- **Metric** - a value that is measured to gain insight into performance
- **Response Time** - The average observed time to complete a transaction
- **Utilization** - The percent of time that a resource is busy
- **Throughput** - The rate at which transactions are completed
- **Capacity** - The maximum throughput of a system

Agenda

- Brief Overview
- Introduction to IBM i Mail Accounting
- Performance Data Collectors
 - Collection Services
 - Job Watcher
 - Performance Explorer
- Performance Data Visualization and Diagnostics
 - Performance Data Investigator
 - Doctor
- Examples

Keep Current on PTFs



It's always good practice to keep current on the latest fixes from IBM

- PTFs address defects
- PTFs introduce new capabilities
 - IBM i Technology Refresh Updates
 - IBM i Group PTFs
 - Database
 - Performance Tools
 - Java
 - HTTP Server
 - HTTP Server Group PTF for Internet Navigator for i functionality
 - PTFs for performance data collectors
 - Collection Services, Job Watcher, Disk Watcher, Performance Explorer

IBM i Performance Tools

Doctor Suite

- Job Watcher
- PEX Analyzer
- Heat Analyzer
- VCB Investigator



Green Screen

- Performance Data Report
- System Configuration
- Performance Center

System Navigator

- Overview of the system
- Drill down into details
- Monitor job status
- View system health

PEX for Power Systems

MCE

Management Central

Third Party Tools



IBM i Performance Tools

Navigate Performance

- Investigate Data (PDI)
- Batch Monitor
- System Monitor

IBM Performance Advisor

Navigator for i

Performance Instrumentation and Data Collection

The Advantage



- IBM **develops the software stack**, top to bottom
 - Instruments the software with performance metrics
 - Performance metrics are component-specific
- IBM develops the **performance data collectors** that harvest those performance metrics
- IBM i has an **integrated database** – DB2
 - This is a **BIG DEAL**
 - Performance data is stored in the database automatically
 - No "add on" application is necessary – it's all in the Operating System
 - Applications mine the performance data in the DB2 files using SQL

IBM i has the best performance instrumentation and data collection capabilities in the industry!

Performance Instrumentation and Data Collection



Introduction to Wait Accounting

Performance Fact:

"All computers wait at the same speed"



What is Wait Accounting?

Wait Accounting = the ability to determine what a job is doing when it is not running



| Exclusive! Patented technology built into IBM i.

Wait Accounting Overview

When a job is not running (using CPU), it is waiting

- But what is it waiting for?

Waits may be normal, some waits are not normal

- Wait Accounting helps to determine what the wait is and if it is a problem

IBM i has instrumented most of the key wait conditions

- Wait information is automatically collected by **Collection Services** and **Job Watcher**



Wait States

Wait information is tracked for each job, thread and task on system
A job/thread/task is in one of three states:

Using CPU

- "Dispatched CPU"
Assigned to a virtual processor so it can begin execution of instructions

Waiting for CPU

- "CPU Queuing"
Ready to use processor, but waiting for it to become available

Waiting for something else...

- Idle waits
- **Blocked waits**

These waits are typically the most interesting waits to focus on

Wait Accounting - Buckets

Wait Buckets = "Wait condition groups" instrumented in the operating system.

- Buckets can then be **analyzed** to determine where a job is spending it's time (running or waiting)
- Categorized into **32** buckets
- Buckets found in both **Collection Services** and **Job Watcher** data
- Waits can be viewed at a **system-level** or at an **individual job/thread/task level**
 - Can also be grouped by generic job name, subsystem, user profile, pool ID, etc.



32 Wait Buckets (6.1 and beyond)

1. Time dispatched on a CPU
2. CPU queuing
3. Reversal
4. Other waits
5. Disk page faults
6. Disk non-fault reads
7. Disk space usage contention
8. Disk operation start contention
9. Disk writes
10. Disk other
11. Journaling
12. Semaphore contention
13. Mutex contention
14. Machine level gate serialization
15. Seize contention
16. Database record/lock contention
17. Object lock contention
18. Ineligible waits
19. Main storage pool contention
 20. Dbase Java™ user including locks (a to f)
 - (g) Journal user while active
 21. Dbase Java JVM (a to f)
 22. Dbase Java other (a to f)
 23. Reversal
 24. Bucket transfer
 25. Bucket receive
 26. Bucket other
 27. IFS
 28. NFS
 29. Data queue receive
 30. Interlocking for work
 31. Synchronization Token contention
32. Abnormal contention

Wait Accounting – “Run-wait” signature

Applying the concepts of wait accounting, we are now able to identify the amount of time the thread/task was running and the time the thread/task was waiting.

Consider the following:

Batch job with total run time of 6 hours

Run-wait signature



Wait Accounting – “Run-wait” signature



Now you can start asking questions such as:

- Are my pool sizes appropriate? What objects is the faulting occurring on?
- Is the write cache being overrun? Is the application forcing writes out synchronously?
- Are all the journals optimally configured? Are unnecessary objects being journaled?
- Am I locking records or objects unnecessarily?



Metrics related to components of wait time

Total count	Disk reads 3,523	Disk writes 17,772	Record Locks 305	Journal 5,741
Total time	42 sec	73 sec	45 sec	44 sec
Avg time per wait	0.012 sec	0.004 sec	0.126 sec	0.007 sec

- Tools capture this level detail
- Useful to know both counts and time

Wait Time Category	Total Wait Time (seconds)
...	...
...	...
...	...
...	...
...	...
...	...
...	...
...	...
...	...
...	...
...	...
...	...

Why Developers should leverage Wait Accounting?

- Helps you understand application characteristics
 - Is it CPU bound? IO bound?
- Helps you to understand where to focus your effort and investment
 - Is there a bottleneck on CPU, Memory, IO, Contention time?
 - Invest resources where greatest benefit will be
- Can offer insight into potential performance issues before end-users are affected
 - Can leverage aspects of wait accounting in test environment
 - Eliminate surprises
 - Identify bottlenecks that prevent testing
- Provides valuable clues to help analyze performance issues as they arise
- Instrumentation part of base IBM i operating system, IBM tools available to help you analyze



Common Waits that Applications use

- Disk Waits
- Semaphores, Mutexes, Synchronization Tokens
- Journaling
- Database record locks
- Object locks
- Sockets



A few other things to know about waits...

- Some waits are “expected” and others “unexpected”
- If waits can be reduced or eliminated, CPU can be used more efficiently
- One wait may be reduced/eliminated, only to have another wait surface
- Likely won't be able to remove all wait times
- What is a wait “tax”?
 - Is there a business impact? Are users complaining?
 - “It depends” but waits more than 25% of run time may need additional analysis

Tools for analyzing Wait Accounting Information



Wait Accounting - IBM i Collectors

- **Collection Services**

- Collects data automatically 24 X 7 at specified intervals (typically 5 or 15 minutes)
- System and job level data
- Starting point



- **Job Watcher**

- Needs to be started/stopped (typically 5 or 10 second intervals)
- Additional detailed data such as call stacks, object waited on, holder
- Frequently needed to solve performance issues



Performance Data Collection vs Visualization



- The performance data **collection** capability is built into the operating system
 - **Everyone** can collect all types of IBM i performance data

- The **visualization** of the data may require additional products
 - Visualization is the display of charts or tables of performance data
 - The Performance Data Investigator is part of the operating system
 - **Everyone** can visualize Collection Services data



Two Graphical Analysis Tools

- Performance Data Investigator – Job Watcher

- Requires STxx-PT1 – Job Watcher feature
- Granted to average user



The screenshot shows a window titled 'Job Watcher' with a table of performance data. The table has columns for Job ID, Job Name, Job Type, and Job Status. The data is organized into a tree view on the left and a main table area. The table contains several rows of job information, including job IDs and names.

Job ID	Job Name	Job Type	Job Status
...
...
...
...

- iDoctor – Job Watcher

- IBM i Service offering, yearly license by serial number
- Granted to advanced user

Wait Accounting IBM i Graphical Analysis Tools

- Two powerful graphical tools to help make your analysis more efficient and productive:

Performance Data Investigator (PDI)

- Component in IBM Navigator for i (browser-based)
- Nothing to install, can view Collection Services for free
- <http://www.ibm.com/coll-services/ibm-ibmnavi/pdi/index.html>

IBM iDoctor for IBM i

- Microsoft Windows based client
- Requires Job Watcher yearly license to see Collection Services data (IBM Service offering)
- <http://www-912.ibm.com/ibm/doctor/en/>

Wait Accounting IBM | Graphical Analysis Tools

- Both GUI tools sit on top of same rich IBM | instrumentation, but not equivalent in presentation and function



e- PDI

Director ->



Wait Accounting Analysis Strategy

- Understand the "big picture" first
 - Understand overall partition characteristics first and where system bottlenecks may be that affecting your application
 - Typically done using **Collection Services** data
 - Drill down to job level
 - Waits can be analyzed in various useful ways:
 - Waits by Job or Task
 - Waits by Service Job or Task
 - Waits by Job User Profile
 - Waits by Job Current User Profile
 - Waits by Task
 - Waits by Subsystem
 - Waits by Service Task
 - Waits by Job Priority
- Continue detailed analysis at a Job Level using **Job Watcher**
 - Narrow focus to interesting instances / jobs
 - Many more job level details available

Using Performance Data Investigator (PDI)

- IBM Navigator for i is the Web console for managing IBM i
 - Has much of the function as System i Navigator
 - Just with a browser user interface
 - Simply point your browser to <http://systemname:2001>



PDI Wait Accounting Perspectives - Where to start



Performance -> Investigate Data -> Collection Services:

Option 1: **CPU Utilization and Waits Overview**

- Combines related waits into higher-level buckets

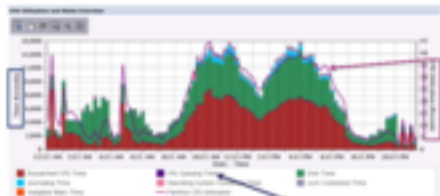
Option 2: **Waits Overview**

- All individual "blocked" wait buckets shown

Additional

IBM Performance Center
 Copyright © 2014 International Business Machines Corporation
 IBM, the IBM logo, and Power Systems are trademarks of International Business Machines Corporation in the United States, other countries, and internationally.

System 'run-wait' signature -> CPU Utilization and Waits Overview



- A chart that combines CPU utilization as well as the wait buckets can be very beneficial in assessing the health of your partition
- In this chart, we can see that the majority of the time, the jobs were spending time in CPU as well as in Disk. Minor amounts of Journal wait time and operating system contention time are also present.

CPU Utilization and Waits Overview – drilldown analysis

Because Disk wait time was fairly significant, drilldown to Disk Waits Overview to further examine the (detailed) waits contributing to this time:



- Can now see that Disk (Page Fault) time is the biggest contributor to Disk Time. (A job needed something in memory, it wasn't there, had to do an IO to bring it into memory before job could continue running).

Waits by Job or Task

The next question likely would be which job(s) are incurring this wait time. Drilling down further, we can see the list of jobs incurring this wait time:



• This type of chart can also be used to understand a job's run-wait signature.

Efficient System with Little Wasting

Primarily Dispatched CPU Time



Processor Bound System

Workload and System Overview

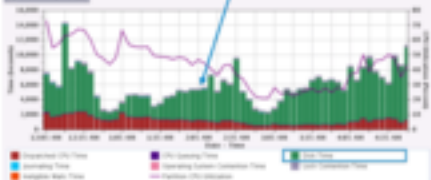
1 2 3 4 5 6



I/O Bound System

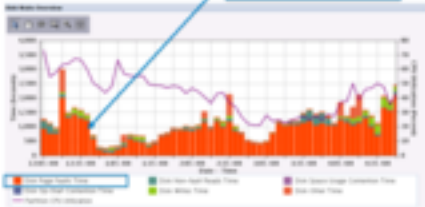
IBM iSeries and Blade Performance

1 2 3 4 5 6 7 8 9 10



IO – Further Investigation

Disk Page Faults Wait Time

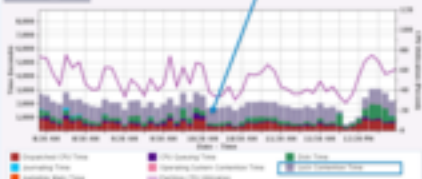


Lock Contention Time Bottleneck

Lock Contention wait time

Workload and Data Overview

1 2 3 4 5 6



Job Watcher data is typically needed to solve lock related issues.

Lock – Further investigation

Object Lock Contention time



Job Watcher data will show object waited on, the holder, and call stacks for both the waiter and the holder (examples shown later on...)

Wait Accounting at a Job Level



Would this job benefit from additional memory?



 Suspended CPU Time

 Wait Non-Real Memory Time

 Wait Write Time

 Page Commitment Time

 Shared Resource Time

 CPU Waiting Time

 Wait Space Usage Commitment Time

 Waiting Time

 Database Record Lock Commitment Time

 Wait Storage Pool Inaccessibility Time

 Wait Page Fault Time

 Wait I/O Wait Commitment Time

 Machine Level Cache Sensitivity Time

 Major Lock Commitment Time

 Abnormal Commitment Time

Wait Accounting at a Job Level



Would this job benefit from additional memory? CPU? Disk?



- Reserved CPU Time
- Wait Non-Real Memory Time
- Wait Write Time
- Wait Completion Time
- Serial I/O Time

- CPU Waiting Time
- Wait Space Usage Completion Time
- Waiting Time
- Database Record Lock Completion Time
- Wait Storage Pool Inaccessibility Time

- Wait Page Fault Time
- Wait Sp-Start Completion Time
- Machine Level Cache Sensitivity Time
- Mount Lock Completion Time
- Abnormal Completion Time

Wait Accounting at a Job Level



Would this job benefit from an improved I/O subsystem?



- Suspended I/O Time
- I/O Non-Read Ready Time
- I/O Write Time
- I/O Completion Time
- I/O Read Time

- I/O Getting Time
- I/O Space Usage Completion Time
- I/O Time
- Database Record Lock Completion Time
- Main Storage Pool Inaccessibility Time

- I/O Page Fault Time
- I/O Up-Start Completion Time
- Machine Level Cache Sensitivity Time
- Major Lock Completion Time
- Abnormal Completion Time

Job Watcher - Additional Benefits

- Collects **more detailed** performance data than Collection Services
 - Call Stacks
 - SQL Statements
 - Additional wait accounting information:
 - Objects being waited on
 - Holder of object
- More **frequent intervals** (seconds)
- Need to start/stop Job Watcher
 - Navigator for I, iDoctor, green screen commands
- To see charts in PDI, need Performance Tools LPP Job Watcher option (chargeable) or iDoctor Job Watcher license for viewing in iDoctor

Job Watcher - Holders versus Waiters

- IBM i keeps track of who is holding a resource, and if applicable, who is waiting to access that resource
 - A **Holder** is the job/thread/task that is holding the serialized resource
 - A **Waiter** is the job/thread/task that wants to access the serialized resource
- IBM i also maintains call stacks for every job/thread/task.
- The combination of
 - **Who** - holders and waiters ... who has it? who wants it?
 - **What** - object being waited on
 - **How** - call stacksprovides a very powerful solution for analyzing wait conditions



Job Watcher – Where to Start



Performance → Investigate Data → Job Watcher:

Option 1: CPU Utilization and Waits Overview

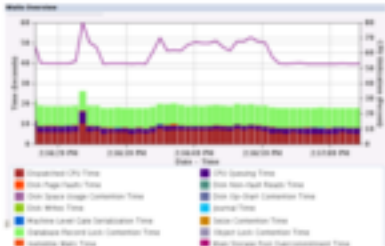
– Combines related waits into higher level buckets

Option 2: Waits Overview

– All individual “tracked” wait buckets shown

Notice similar perspectives available as Collection Services

Job Watcher – Waits Overview



Notice some wait buckets, but more granular intervals

Job Watcher – Additional Interval Details

Viewed on Job Details

Job Information: [XXXXXXXXXXXXXXXXXXXX](#)
XXXXXXXXXXXXXXXXXXXX

Job Name: [XXXXXXXXXXXXXXXXXXXX](#)

Start Date: [XXXXXXXXXXXXXXXXXXXX](#)

End Date: [XXXXXXXXXXXXXXXXXXXX](#)

Created on Job: [XXXXXXXXXXXXXXXXXXXX](#)

Job ID: [XXXXXXXXXXXXXXXXXXXX](#)

Job Status: [XXXXXXXXXXXXXXXXXXXX](#)

Jobs: [XXXXXXXXXXXXXXXXXXXX](#)

Job: [XXXXXXXXXXXXXXXXXXXX](#)

Job Description: [XXXXXXXXXXXXXXXXXXXX](#)

Report Job Details: [XXXXXXXXXXXXXXXXXXXX](#)

Job Start Date: [XXXXXXXXXXXXXXXXXXXX](#)

Job End Date: [XXXXXXXXXXXXXXXXXXXX](#)

Job ID: [XXXXXXXXXXXXXXXXXXXX](#)

[Job Details](#)

Job Watch

Job Name	Program	Module	Resource
1			XXXXXXXXXXXXXXXXXXXX
2			XXXXXXXXXXXXXXXXXXXX
3			XXXXXXXXXXXXXXXXXXXX
4			XXXXXXXXXXXXXXXXXXXX
5			XXXXXXXXXXXXXXXXXXXX
6			XXXXXXXXXXXXXXXXXXXX
7			XXXXXXXXXXXXXXXXXXXX
8			XXXXXXXXXXXXXXXXXXXX
9			XXXXXXXXXXXXXXXXXXXX
10			XXXXXXXXXXXXXXXXXXXX
11			XXXXXXXXXXXXXXXXXXXX
12			XXXXXXXXXXXXXXXXXXXX
13			XXXXXXXXXXXXXXXXXXXX
14			XXXXXXXXXXXXXXXXXXXX
15			XXXXXXXXXXXXXXXXXXXX
16			XXXXXXXXXXXXXXXXXXXX
17			XXXXXXXXXXXXXXXXXXXX
18			XXXXXXXXXXXXXXXXXXXX
19			XXXXXXXXXXXXXXXXXXXX
20			XXXXXXXXXXXXXXXXXXXX

Job Watched

Job Watched

Job Watched

Wait Accounting - Recommendations: Be proactive!

- Use the rich IBM i wait accounting instrumentation found in
 - Collection Services & Job Watcher
 - Use POI or iDoctor to view/analyze
- Understand your particular's "run-wait" signature and normal patterns



- Identify bottlenecks

Recommendations: Be proactive!

- **Keep a baseline**

- Collection Services (Job Watcher data is also nice to have)
 - Weekly, end-of-month, and/or year
 - Prior to any hardware, software, configuration related change

- **A baseline provides a reference point**

- It is the expected performance characteristics over a defined period of time
- Having one makes it easier to recognize changes and its effect



- **Wait bucket information can make it easier to determine what has changed?**
 Both at a partition level as well as an [individual job level](#)

IBM i Performance Data Collectors

Performance Data Collection Architecture

- Collection Services
- Job Watcher
- Disk Watcher
- Performance Explorer



Collect System-wide Performance Data



Collect Performance Data 24/7



- If something goes wrong, you have data that will help **analyze** the problem, **fix** it, and **prevent** it from happening in the future
- If you can't solve the problem, you have information that makes it easier for IBM Support to **solve the problem faster**
- To provide a **reliable baseline** so you can understand the **impact** that a software, network, or environmental **change** had on the performance of your system
- To provide historical information that enables you to **plan for future growth** based on real trends, not guesses.

Patterns in Performance Data



- Performance data typically has patterns

- Daily, weekly, monthly, yearly



- Understand your typical patterns

- Recognize change

Job Watcher

- Job Watcher returns real-time information about a selected set of jobs, threads, or LIC tasks
- Job Watcher collects additional types of data that Collection Services does not, as well as more frequent intervals
 - Job Watcher has more overhead than Collection Services
- Data collected by Job Watcher includes
 - Wait times
 - CPU
 - IO activity
 - Call stacks
 - SQL statements
 - Communications statistics
 - Activation Group statistics



Run Job Watcher when you need detailed performance data for diagnostic purposes.

There are clients that run Job Watcher 24x7 to always have diagnostic data available.

Need to manage the data carefully. 

Job Watcher

- Job Watcher collects **more detailed** performance data than Collection Services and at **more frequent intervals**
 - CPU and IO (like Collection Services)
 - Call Stacks
 - SQL Statements
 - Detailed Wait information
 - Objects being evaluated, even records number of files
 - Number of objects
- Job Watcher **does not collect everything** that Collection Services collects.
 - It does not always collect information about every thread
 - Thread must use CPU during interval
 - Thread must exist for entire interval
 - It does not collect memory pool or detailed IO statistics
- Data is written to OED files

Job Watcher Usage Tips

- Use Job Watcher when you need detailed performance data to resolve a problem
 - Typically problem has been scoped first by Collection Services
- For problem determination Job Watcher can be run on **specific jobs**
 - **Caution:** When using Job Watcher on specific job(s), you may not get detailed HADR information
- Multiple collections can be run at the same time
- Need to manage the amount of data collected



Basic Job Watcher Data Collection Steps

1. Create the Job Watcher definition
 - Or use one of the IBM-supplied definitions
2. Start the Job Watcher collection
3. Let it run until the problem has occurred
4. Stop the Job Watcher collection
5. Analyze the data

There are times when you may want to run Job Watcher continuously

How Do I Run Job Watcher with the Commands?

• CL Commands

- Add Job Watcher Definition (ADDJWDFN) to define the collection
 - Identifies the performance data that is to be collected

- Remove Job Watcher Definition (RMVJWDFN) to remove a definition
(Note: Job Watcher Definitions can only be deployed through the DAE)

- Start Job Watcher (STRJW) to start the collection

- End Job Watcher (ENDJW) to end the collection (optionally)

IBM-Supplied Job Watcher Definitions

- Several pre-defined Job Watcher definitions are available
 - The main difference is the sample intervals
 - o QJWB01* - 1 second intervals with call stacks, another to include I/O.
 - o QJWB02* - 10 second intervals with call stacks, another to include I/O.
 - o QJWB03* - 5 second intervals with call stacks, another to include I/O.
- Recommendations:
 - Collect with Call Stacks and I/O.
 - Use 10 second intervals for general analysis (QJWB02).
 - Use 5 second intervals for complex or intermittent issues, or for contention related problems (QJWB03).



 A screenshot of the Job Watcher configuration interface showing a table of job watcher definitions. The table has columns for 'Job Watcher Name', 'Sample Interval', and 'Include I/O'. The data is as follows:

Job Watcher Name	Sample Interval	Include I/O
QJWB01*	1 second	Yes
QJWB02*	10 second	Yes
QJWB03*	5 second	Yes

Job Watcher Authority Requirements

- **Commands:**

1. You must have service (*SERVICE) special authority
 - Change User Profile to add *SERVICE authority to create Job Watcher Definitions or to Start Job Watcher
2. **-OR-** Be authorized to the Job Watcher function of the operating system
 - Change Function Usage (CHFCNFUSG) command, with a function ID of QSM_SERVICE_JOB_WATCHER can be used to change the list of users that are allowed to use this command.

```
CHFCNFUSG PWD(QSM_SERVICE_JOB_WATCHER) USER(*userprofile) USAGE(*ALL) CMD(*)
```

- **Definitions:**

- Additional authority is needed to see the definitions for each as they are shipped *PUBLIC *EXCLUDE. To see the definitions shipped in Job Watcher, users will need authority to the QAPYJWDPM file in QUSRC012

<https://www.ibm.com/developerworks/community/wiki/home?lang=en&wiki=IBM%20z%20TechCenter%20zSeries%20Open%20Authority>

Performance Explorer

- Performance Explorer helps identify the causes of performance problems that cannot be resolved using one of the other performance data collectors
 - Collects more detailed information about a specific application, program, or resource
- Performance Explorer is typically used for two main reasons:
 - Detailed performance trace data is needed to identify the performance problem
 - Analyzing the performance of applications
- Performance Data Investigator supports [profile](#) collections only
- iExecor is required for advanced PEX Analysis

Performance Explorer

- Performance Explorer is the most sophisticated IBM i performance tool
 - Can collect the details of every I/O operation, every task switch
 - Hundreds of events collected
 - Thus, most complex to use
 - More overhead

- Typically, problem has been scoped by other tools first

- Generally used by IBM performance analysis experts

Except.....

Performance Explorer – “TPROF” usage

- “Trace-profiler” is a fairly easy, and fairly “light-weight” PEX collection that can be useful to application developers, especially when trying to diagnose high CPU issues
 - Provides CPU usage at a program/procedure level
 - Make sure you have latest PEX.PTF's applied
 - <http://www.ibm.com/support/ctgsec.do?catid=951021020>
 - Can be run over subset of jobs

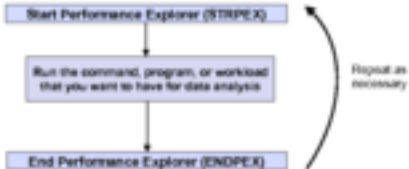
Steps:

1. Add a PEX definition:

```
ADDPEX DEFINITION
  TRACPROF 100000
  PEXNAME TRACPROF
  PEXID TRACPROF
  PEXTYPE TPROF
  PEXPROP TRACPROF
  PEXSTRT TRACPROF
  PEXSTOP TRACPROF
```


Performance Explorer – “TPROF” usage

2. Collect data



3. Analyze data



Performance Explorer TPROF reports – PDI

The screenshot displays four overlapping Performance Explorer TPROF report windows. The primary window shows a hierarchical tree of system components with columns for 'Component Name', 'Module Name', 'Description Name', 'Component', and 'CPU usage'. A green box highlights the 'CPU usage' value for the 'PDI' component, which is 100.00%.

The 'CPU usage' column values for the main report are as follows:

Component Name	Module Name	Description Name	Component	CPU usage
OS/390		OS/390	OS/390 (System)	100.00%
OS/390		OS/390	OS/390 (System)	100.00%
OS/390		OS/390	OS/390 (System)	100.00%
OS/390	OS/390	OS/390	OS/390 (System)	100.00%
OS/390	OS/390	OS/390	OS/390 (System)	100.00%
OS/390	OS/390	OS/390	OS/390 (System)	100.00%
OS/390	OS/390	OS/390	OS/390 (System)	100.00%
OS/390	OS/390	OS/390	OS/390 (System)	100.00%
OS/390	OS/390	OS/390	OS/390 (System)	100.00%
OS/390	OS/390	OS/390	OS/390 (System)	100.00%

The 'PDI' component is highlighted in green in the original image, showing a CPU usage of 100.00%.

The other windows show detailed views of the 'PDI' component, including its 'CPU usage' and 'CPU usage' values, and a 'Detailed CPU usage' table.

The 'Detailed CPU usage' table shows the following data:

Component Name	Module Name	Description Name	Component	CPU usage
OS/390		OS/390	OS/390 (System)	100.00%
OS/390		OS/390	OS/390 (System)	100.00%
OS/390		OS/390	OS/390 (System)	100.00%
OS/390		OS/390	OS/390 (System)	100.00%
OS/390		OS/390	OS/390 (System)	100.00%
OS/390		OS/390	OS/390 (System)	100.00%
OS/390		OS/390	OS/390 (System)	100.00%
OS/390		OS/390	OS/390 (System)	100.00%
OS/390		OS/390	OS/390 (System)	100.00%
OS/390		OS/390	OS/390 (System)	100.00%

Watches

- Watches provide a programmatic interface to be notified when the following occur:
 - Message
 - Licensed Internal Code Log (LIC Log)
 - Problem Activity Log Entry (PAL entry)
- Start Watch (STRWCH) command or API (QSCSWCH)
- End Watch (ENDWCH) command or API (QSCREWCH)
- When the condition being watched occurs, your program gets control and you can take any action you want



<http://www.ibm.com/eserver/zseries/techref/zos/monitoring/monitoring.html>

Examples

What is causing disk wait time?

Let's Look at the Disk Waits



We see it's faulting.... let's find out who did it

Drill-down into "Units by Job or Task"



Let's find out who the user is

We now have several clues:

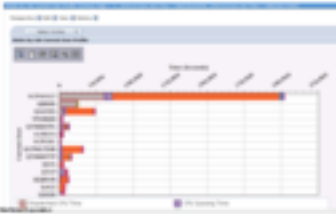
We know the job - `QMAILD02U` - `QMAILD02U` server jobs

We know the time - early afternoon

We know the user profile - `QMAIL`

The `QMAIL` will help! We need the job's current user profile

Walls by All Current User Profiles shows us `QMAILD02U` is the guilty party



Viewing Waits with Job Watcher

Example of Object Lock Contention

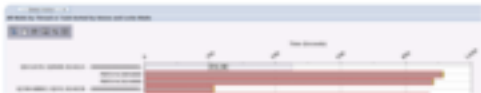




Seizes and Lock Waits Overview → All Waits by Thread or Task...



All Waits by Thread or Task ↕
All Waits for One Thread or Task



Select the job with the object lock contention time. Look at all waits for that one thread or task



All Waits by Thread or Task -> Interval Details



7.2 ...More Information

Properties | Job | View |

Collection

Task	Start	End	Status
Task1	Mon, May 12, 2008 00:00:00 AM	Mon, 05/12/2008	Completed
Task2	Mon, May 12, 2008 00:00:00 AM	Mon, 05/12/2008	Running

Type: Job Monitor File-based Collection
No task: 0

Interval or Task Details

Job information:	spoolmgr/psadm/psadm -	Priority:	0
Current user profile:	psadm/psadm	Pool:	0
Object monitor on:	psadm/psadm	Type description:	IBM Monitoring Data
Max function:	00 milliseconds	Segment type description:	Monitoring Data (Metric)
Current interval width:	00:00:00	Max object size:	None (default) No interval
Rolling job or task:	None (default) No interval	Interval description:	Mon, 05, 2008 0:00:00 AM
Max. interval:	None (default) No interval	Interval ID in TMS:	1234567

[View Details](#)

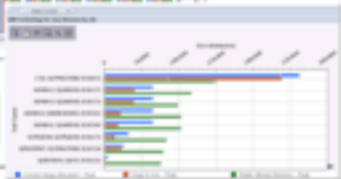
More "Interval or Task Details"

Details

Easily navigate from one interval to the next

More PDI Examples

Java Perspectives in Collection Services



Find that job using a lot of heap...

Java Perspectives

Breakdown for one job -

Look at the heap and memory usage over time for one selected job.



IBM Technology for Java Memory for One Job

Database Full Opens

Full Opens are expensive resource-wise

A screenshot of a database configuration interface. A blue box highlights the 'Full Opens' section, which includes a list of settings for different types of full opens (e.g., 'Full Opens for Native', 'Full Opens for SQL', etc.).



General recommendation is to keep Native Full Opens per second < 1000

Next, find jobs
doing full opens.

Database Full Opens



In an RPL program, full opens are caused by the use of SET OPEN FULL command of SET OPEN. Based on the priority, the setting on the SET OPEN program will keep the program in memory, keep the open and perform all, restore, archive, release, etc. Also, each table is locked by the RPL program.

Normal full opens are for data expansion than full opens. This consumes less CPU, less storage and use faster than full opens. There are some trade-offs of changing full opens to shared full opens, but maintenance of shared opens is usually easier to manage and maintain than changing all full opens to full opens in RPL programs. The following table provides information on open and maintenance of any shared table.

```

SET OPEN FULL table-name open-mode open-mode open-mode open-mode open-mode open-mode open-mode open-mode open-mode
SET OPEN FULL table-name open-mode open-mode open-mode open-mode open-mode open-mode open-mode open-mode open-mode
SET OPEN FULL table-name open-mode open-mode open-mode open-mode open-mode open-mode open-mode open-mode open-mode
  
```

```

SET OPEN FULL table-name open-mode open-mode open-mode open-mode open-mode open-mode open-mode open-mode open-mode
SET OPEN FULL table-name open-mode open-mode open-mode open-mode open-mode open-mode open-mode open-mode open-mode
SET OPEN FULL table-name open-mode open-mode open-mode open-mode open-mode open-mode open-mode open-mode open-mode
  
```

Temporary Storage Allocation/Deallocation perspectives

Storage Allocation Perspectives

Where is my temporary storage going?

Expand Collection Services

- Storage Allocation
 - Storage Allocation/Deallocation Overview
 - Storage Allocation/Deallocation by Thread or Task
- Temporary Storage
 - Temporary Storage Allocation Accounting
 - Temporary Storage Allocation/Deallocation Overview
 - Temporary Storage Allocation/Deallocation by Job or Task
 - Temporary Storage Allocation/Deallocation by Thread or Task
 - Temporary Storage Allocation/Deallocation by General Job or Task
 - Temporary Storage Allocation/Deallocation by Job User Profile
 - Temporary Storage Allocation/Deallocation by Job Current User Profile
 - Temporary Storage Allocation/Deallocation by Subsystem
 - Temporary Storage Allocation/Deallocation by System Task

Overview

Notes

Temporary Storage Allocation Accounting

Description

This tool shows the amount of temporary storage created by jobs and users on the system. The amount of job temporary storage and the amount of temporary storage used by individual users is broken down by job. The tool also shows the amount of temporary storage used by the system and the amount of temporary storage used by the system.

How to use

Temporary Storage Allocation Accounting Overview

Temporary Storage Allocation / Deallocation Overview

Generally, allocations and deallocations following a similar pattern



From an overview perspective, drill down to more detail



What has the performance adjuster been doing to my pools?

- Collection Services allows you to look backward in time



System Trend Data

1



What does the faulting look like when I was testing?



IBM iDoctor for i

- Product developed by the IBM Rochester Support Center for deep, detailed performance analysis
- Major components
 - Job Watcher
 - Job Watcher
 - Collection Services Investigator
 - Disk Watcher
 - Plus Cache Analyzer - graphical analysis of the system's DCL, Plus Cache
 - PEX Analyzer
 - Heap Analyzer—Classic JVM heap analysis
 - VIOS Investigator
- http://www-912.ibm.com/i_doctor.net



IBM iDoctor for i



- Windows client application
https://www-912.ibm.com/i_dinodoctor.nsf
- Detailed performance analysis and diagnostics

iDoctor versus Performance Data Investigator

- You have two graphical interfaces for performance data analysis...
 - Which should you use? It depends....

Feature	iDoctor	PDI
Interface	Windows Java	Browser
Host Access	Yes	Yes
Collection Services	Yes	Yes
Job Monitor	Yes	Yes
Task Monitor	Yes	Yes
Performance Profiles	Yes	Profile collection only
Containers	Yes	Yes
Job Monitor Monitor	Yes	Yes
Containerize	Yes	Yes
User Defined graphs and queries	Yes	Yes
Update Frequency	Monthly	Real Time
Support	Customer	Support Center
Prerequisites	None to install	- Collector Services at no additional charge with 1 Job Monitor, Database, and Performance Profiles included with base PDI product Job Monitor is an additional option at PDI and has an additional charge
Exporting Features	Yes (e.g. CSV, HTML, PDF)	Yes
	Yes	Yes

IBM i Performance FAQ a MUST read!

[Go to the IBM i Performance FAQ](#)

[or visit the IBM i Performance FAQ on the IBM i Performance website](#)

in Power Systems Performance



IBM i on Power - Performance FAQ
October 12, 2011

Now it all makes sense!



IPM for Power Systems



WorkLoad Estimator



IBM Navigator for i



Green Screen

Performance Explorer
Performance Snap Reports
System commands



Collection Services
Health Indicators
Monitors
Dashboards
Job Monitor
Disk Monitor
Net Explorer
Event Manager





ithankyou

www.ibm.com/power/i

References



PTF Enhancements via PTFs

1. [IBM i developerWorks](#)
2. [Technology Modern](#)
3. [Performance Tools](#)
4. [Performance on the Web](#)



You will find a list of enhancements by time frame with links to the details.

- [developerWorks](#)

- [Performance Tools](#)

- [Additional performance tools resources](#)
- [Performance on the Web](#)
- [Performance Data Collector](#)



Discover the latest enhancements for IBM operating system and middleware products and frequently enhanced tool sets with our new site.

[View all performance tools](#)

- [Contact](#)

- [IBM | Performance Data Investigator](#)

- [IBM | Performance Data Investigator – Edit Perspectives](#)

- [IBM | Web Accessing](#)

- [How to use the IBM® Model Performance tool](#)

IBM i Web Sites with Performance Information



[IBM Knowledge Center](#)

- [CPU](#)
- [I/O](#)



[IBM i Performance Management](#)

This web site has a lot of GREAT references and papers -- see the [IBM i 7.5.0](#)

- [Performance Management for Power Systems](#)
- [IBM Workload Estimator](#)
- [Docx](#)
- [Job Workload Estimator](#)



You and i

You and i only need one language

<http://www.ibm.com/systems/mag.com/Blogs/You-and-i/>



i Can

i Can do what you can't

<http://www.ibm.com/systems/mag.com/Blogs/i-Can/>



Mr. Modern-i-zation Rowe

<https://www.systemsdeveloper.com/blogs/TheBlog/>



DB2 for i

DB2 for i

<http://www.ibm.com/db2fori/>

iCan

For a simple list of all
blogs on one page -
[iCan Blog of Blogs](#)

iCan *Technical Tips for i*

[New Books: 2006 & 07](#)
[2007 New Books: 2007 & 08](#)
[2007 & 08 Power Systems: 2006, 2007 & 08](#)
[2007 & 08 Power: 2006](#)
[2007 & 08 Power: 2007](#)
[2007 & 08 Power: 2008](#)
[2007 & 08 Power: 2009](#)
[2007 & 08 Power: 2010](#)
[2007 & 08 Power: 2011](#)
[2007 & 08 Power: 2012](#)
[2007 & 08 Power: 2013](#)
[2007 & 08 Power: 2014](#)
[2007 & 08 Power: 2015](#)
[2007 & 08 Power: 2016](#)
[2007 & 08 Power: 2017](#)
[2007 & 08 Power: 2018](#)
[2007 & 08 Power: 2019](#)
[2007 & 08 Power: 2020](#)
[2007 & 08 Power: 2021](#)
[2007 & 08 Power: 2022](#)
[2007 & 08 Power: 2023](#)
[2007 & 08 Power: 2024](#)
[2007 & 08 Power: 2025](#)
[2007 & 08 Power: 2026](#)
[2007 & 08 Power: 2027](#)
[2007 & 08 Power: 2028](#)
[2007 & 08 Power: 2029](#)
[2007 & 08 Power: 2030](#)
[2007 & 08 Power: 2031](#)
[2007 & 08 Power: 2032](#)
[2007 & 08 Power: 2033](#)
[2007 & 08 Power: 2034](#)
[2007 & 08 Power: 2035](#)
[2007 & 08 Power: 2036](#)
[2007 & 08 Power: 2037](#)
[2007 & 08 Power: 2038](#)
[2007 & 08 Power: 2039](#)
[2007 & 08 Power: 2040](#)
[2007 & 08 Power: 2041](#)
[2007 & 08 Power: 2042](#)
[2007 & 08 Power: 2043](#)
[2007 & 08 Power: 2044](#)
[2007 & 08 Power: 2045](#)
[2007 & 08 Power: 2046](#)
[2007 & 08 Power: 2047](#)
[2007 & 08 Power: 2048](#)
[2007 & 08 Power: 2049](#)
[2007 & 08 Power: 2050](#)
[2007 & 08 Power: 2051](#)
[2007 & 08 Power: 2052](#)
[2007 & 08 Power: 2053](#)
[2007 & 08 Power: 2054](#)
[2007 & 08 Power: 2055](#)
[2007 & 08 Power: 2056](#)
[2007 & 08 Power: 2057](#)
[2007 & 08 Power: 2058](#)
[2007 & 08 Power: 2059](#)
[2007 & 08 Power: 2060](#)
[2007 & 08 Power: 2061](#)
[2007 & 08 Power: 2062](#)
[2007 & 08 Power: 2063](#)
[2007 & 08 Power: 2064](#)
[2007 & 08 Power: 2065](#)
[2007 & 08 Power: 2066](#)
[2007 & 08 Power: 2067](#)
[2007 & 08 Power: 2068](#)
[2007 & 08 Power: 2069](#)
[2007 & 08 Power: 2070](#)
[2007 & 08 Power: 2071](#)
[2007 & 08 Power: 2072](#)
[2007 & 08 Power: 2073](#)
[2007 & 08 Power: 2074](#)
[2007 & 08 Power: 2075](#)
[2007 & 08 Power: 2076](#)
[2007 & 08 Power: 2077](#)
[2007 & 08 Power: 2078](#)
[2007 & 08 Power: 2079](#)
[2007 & 08 Power: 2080](#)
[2007 & 08 Power: 2081](#)
[2007 & 08 Power: 2082](#)
[2007 & 08 Power: 2083](#)
[2007 & 08 Power: 2084](#)
[2007 & 08 Power: 2085](#)
[2007 & 08 Power: 2086](#)
[2007 & 08 Power: 2087](#)
[2007 & 08 Power: 2088](#)
[2007 & 08 Power: 2089](#)
[2007 & 08 Power: 2090](#)
[2007 & 08 Power: 2091](#)
[2007 & 08 Power: 2092](#)
[2007 & 08 Power: 2093](#)
[2007 & 08 Power: 2094](#)
[2007 & 08 Power: 2095](#)
[2007 & 08 Power: 2096](#)
[2007 & 08 Power: 2097](#)
[2007 & 08 Power: 2098](#)
[2007 & 08 Power: 2099](#)
[2007 & 08 Power: 2100](#)

Performance Management on IBM i Web Site

<http://www-03.ibm.com/systems/power/software/management/performance/index.html>

Performance management on IBM i

Overview Tools Performance Navigator Resources

Plan what you need

[Performance Data Collectors](#)

There are four collectors on IBM i that collect performance related data and store the information in database files, each having their own unique characteristics: Collector Services, IBM Job Watcher, IBM Data Watcher, and Performance Explorer.

[Performance Data Investigator \(PDI\)](#)

Use the Investigator Data tool feature in the web-based IBM Systems Director Navigator for i to view and analyze the data collected from any of the four data collectors found on IBM i. The investigator allows you the ability to work with the data interactively in chart or table form.

[Monitor for IBM i](#)

A family of products including Job Watcher, PDI Analytics, and Job Watcher Tools for i are focused on increasing the overall health of a system by providing automated analysis of a variety of performance related data.

[MIB for Power Systems](#)

It can that can automatically collect system utilization information and can produce regular reports which show the utilization and growth trends of your system.

[Performance and Stability Services](#)

This web program for changes in the data center when using the IBM operating system on Power Systems hardware will help from IBM Systems, all Services and Training. [What's New](#)

What Happened to the PCRM?

- Performance Capabilities Reference Manual – “PCRM”
- Was THE reference manual for all things related to IBM i performance considerations
 - Content was carried forward but not always updated
- Beginning in 2014, the PCRM only covers CPW information
 - Updates for new hardware models and CPW ratings
 - Older versions are still available for download
- Use other sources for IBM i performance information:
 - The IBM i Performance FAQ
 - Papers under the resources section on the Performance Management site
 - Knowledge Center
 - developerWorks

End to End Performance Management on IBM i

- Understand the end-to-end performance management process
- Monitor performance using the graphical interface in IBM i
- Use the command line interface



<http://www.redbooks.ibm.com/redbooks/pdfs/rd01901.pdf>

Power Systems 

IBM i 7.1 Technical Overview, with Technology Refresh Updates

Covers the 7.1 content through
Technology Refresh 7

Chapter 6 – Performance Tools

Chapter 17, Section 6 –
Performance in Navigator for i

IBM

IBM i 7.1 Technical Overview with Technology Refresh Updates



IBM i 7.2 Technical Overview with Technology Refresh Updates

Covers the 7.2 content through
Technology Refresh 1

Section 2.6 – Performance

Section 8.6.7 – Job level SQL
stats in Collection Services

IBM

IBM i 7.2 Technical Overview with Technology Refresh Updates

- IBM i 7.2 Technical Overview with Technology Refresh Updates
- IBM i 7.2 Technical Overview with Technology Refresh Updates
- IBM i 7.2 Technical Overview with Technology Refresh Updates



Redbooks and Redpapers on IBM i Performance Tools

- [IBM i 7.5 Technical Overview with Technology Refresh updates](#)
- [IBM i 7.4 Technical Overview with Technology Refresh updates](#)
- [Application and System Performance Analysis Using TSM Statistics](#)
- [Best Practices for Monitoring IBM i Jobs and Objects and a Few other useful links](#)
- [IBM Operating Tools for System Administrators: An IBM i Reference for System Administrators](#)

The following redbooks are a bit dated but still have some useful information.

- [IBM Systems Director Overview for IBM i \(October 05\)](#)
- [IBM eServer Series Performance Management Tools](#)
- [A Systems Management Guide to Performance Management for Systems and System a series](#)
- [Using IBM Tools Work on IBM System i7 Systems](#)
- [Managing iSeries with Operations Navigator: iSeries Volume 6 Performance Management](#)
- [IBM eServer Series Job Monitor: Advanced Performance Tool](#)
- [IBM eServer Series Systems Management ToolBook](#)

Articles

- IBM Systems Magazine, IBM i – [“Power Systems: Working With Power?”](#), January 2014
- IBM Systems Magazine, IBM i – [“Customizing a Performance Data Investigator”](#), August 2011
- IBM Systems Magazine, IBM i – [“Investigating the Investigator”](#), May 2010
- IBM Systems Magazine, IBM i – [“It’s All High Performance”](#), Aug 2009
- SystemNetwork – [“Performance Data Investigator: Consolidate Functions in One Place”](#), June 2009
- SystemNetwork – [“IBM Systems Director: Navigate to i Performance Tools Overview”](#), June 2009
- IBM Systems Magazine, IBM i – [“A Different Performance”](#), Nov 2008
- IBM Systems Magazine, IBM i – [“Introducing IBM Systems Director: Navigate to i”](#), Aug 2008
- IBM Systems Magazine, IBM i – [“A Different Effect”](#), Nov 2007
- IBM Systems Magazine, IBM i – [“Monitor, Performance Manager”](#), Nov 2006

Articles on Job Watcher

- ["Web Power"](#)
- [Introduction to Job Watcher Green Screen Commands](#)
- [Top 10 Hidden iDoctor Gems](#)
- [Using iDoctor for iSeries Job Watcher to Determine Why Jobs Wait](#)

Articles on Disk Performance

- **A New Way to Look at Disk Performance**
[http://www.ibm.com/systems/power/infocenter/docguide/infocenter/v5r5/New_Way_to_Look_at_Disk_Performance.html](#)
- **Analyzing Disk Watcher Data**
[http://www.ibm.com/systems/power/infocenter/docguide/infocenter/v5r5/Analyzing_Disk_Watcher_Data.html](#)
- **Using Wait State Accounting to Determine Disk Performance**
[http://www.ibm.com/systems/power/infocenter/docguide/infocenter/v5r5/Using_Wait_State_Accounting_to_Determine_Disk_Performance.html](#)
- **Understanding Disk Performance, Part 2: Disk Operation on I/OG**
[http://www.ibm.com/systems/power/infocenter/docguide/infocenter/v5r5/Understanding_Disk_Performance_Part_2_Disk_Operation_on_IOG.html](#)
- **Understanding Disk Performance Metrics**
[http://www.ibm.com/systems/power/infocenter/docguide/infocenter/v5r5/Understanding_Disk_Performance_Metrics.html](#)
- **Planning for Solid State Drives**
[http://www.ibm.com/systems/power/infocenter/docguide/infocenter/v5r5/Planning_for_Solid_State_Drives.html](#)
- **Moving Data to Solid State Drives**
[http://www.ibm.com/systems/power/infocenter/docguide/infocenter/v5r5/Moving_Data_to_Solid_State_Drives.html](#)
[http://www.ibm.com/systems/power/infocenter/docguide/infocenter/v5r5/](#)
- **Customer use of SSDs**
[http://www.ibm.com/systems/power/infocenter/docguide/infocenter/v5r5/Customer_Use_of_SSDs.html](#)
- **A Look at System i Integrated DACC Configuration and Performance under I/OG**
• Redpaper REDP-3019-08

Systems Management References

- Navigator for I on developerWorks
<http://www.ibm.com/developerWorks/tech/asm/asm.html>
- IBM Application Runtime Expert
<http://www.ibm.com/developerWorks/tech/asm/asm.html>
- Uncovering Application Runtime Expert - IBM i 5.1
<http://www.ibm.com/developerWorks/tech/asm/asm.html>
- Web Performance Advisor
<http://www.ibm.com/developerWorks/tech/asm/asm.html>
- IBM Systems Director
<http://www.ibm.com/developerWorks/tech/asm/asm.html>
- IBM Trust Monitoring
<http://www.ibm.com/developerWorks/tech/asm/asm.html>
- IBM Trust Monitoring Agent for IBM i
<http://www.ibm.com/developerWorks/tech/asm/asm.html>

Special notices

The document developed by IBM offerings in the United States as of the specific situation. IBM may not make these offerings available in other countries, provide information or subject to legal or other restrictions. Contact your local IBM business contact for information on the IBM offerings available in your area.

Information in this document concerning non-IBM products was obtained from the suppliers of these products or other public sources. Specific to the capabilities of non-IBM products should be addressed to the suppliers of these products.

IBM may have patents or pending patent applications covering subject matter in this document. The funding of this document does not give anyone license to these patents. Your business requires a policy. In IBM business of covering IBM Corporation may handle these, results, or IBM may not.

Statements regarding IBM future direction and identification subject to change or withdrawal without notice, and represent goals and objectives only.

The information contained in this document has not been validated by any formal IBM test and is provided "AS IS" without warranty or guarantee unless expressed in writing.

Information not included in this document are considered as "Out of Scope" of the release in which such IBM products can be used and the results may be inconsistent. Actual results may vary and performance may differ from any. Warning of individual user configurations and conditions.

IBM Global Financing offerings are provided through IBM Credit Corporation in the United States and other IBM subsidiaries and may not be available in certain countries and government clients. Rates are subject to creditworthiness, financing terms, offering type, equipment type and options, and may vary by country. Other restrictions may apply. Rates and offerings are subject to change without or withdrawal without notice.

IBM is not responsible for printing errors in this document that result in giving or information inaccurate.

IBM products may differ from those suggested by your subject to change without notice. Results may vary.

IBM hardware products are manufactured from materials or may be non-renewable and costly. Regardless, no warranty from IBM.

An IBM representative contacted in this document may be contacted in a published environment. Actual results may vary significantly and are dependent on how they are being used. Software is provided as software and without warranty. IBM representatives contact in this document may have been made in a published environment. There is no guarantee these representatives will be the same or generally available outside. Some representatives contact in this document may have been introduced through an application. Users of this document should verify the availability of the IBM specific environment.

Revised September 28, 2006

