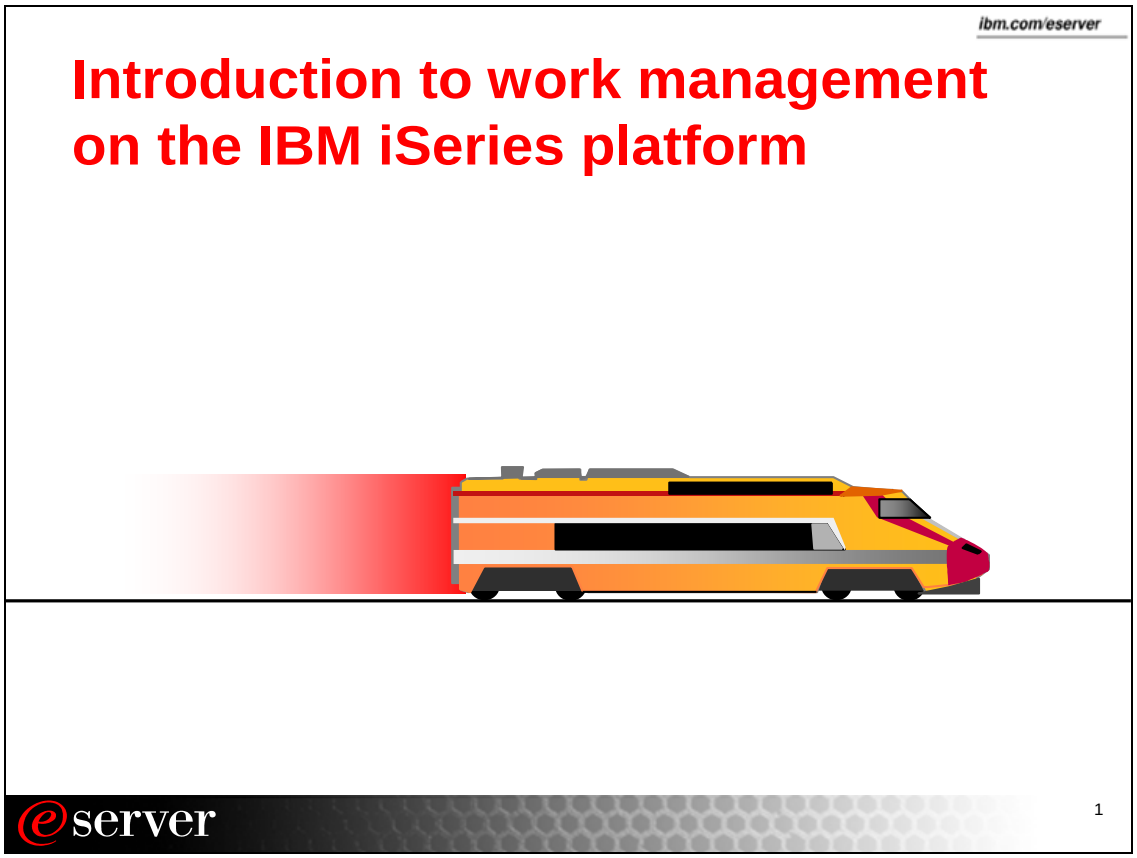
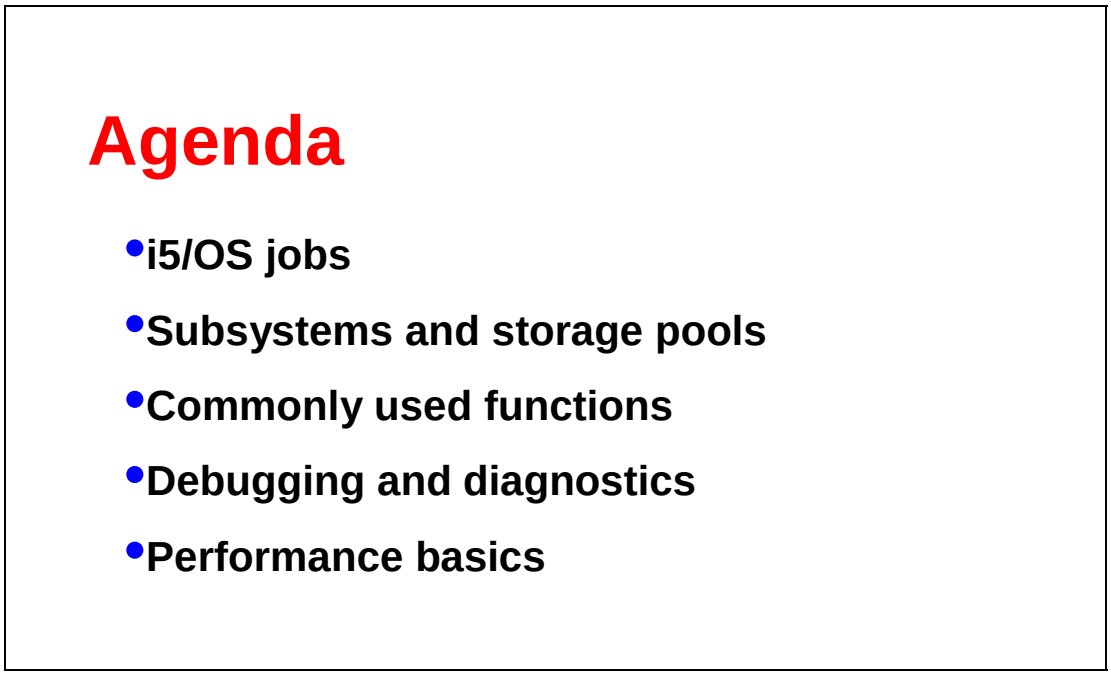
Table of contents



Introduction

All systems use different methods to manage their work and different terms to describe their various functions. The IBM® iSeries™ system is no exception. It is important to know how the iSeries environment manages its work, and the terminology used for its functions, to make better use of the system.

"Work management" is the term used to describe the functions that manage work on the iSeries system. This course will address terminology, functions, and diagnostic procedures that can be used to make the best use of your iSeries system.



Agenda

In this course, you will become familiar with work management concepts and terminology. We will discuss the two types of work (or processing) the iSeries system does (interactive and batch). We will also explain what a job is, some of its characteristics, and the types of jobs that can run on the iSeries system. You will come to understand what a subsystem is, the relationship between a job and subsystem, and basic debugging and diagnostic techniques related to work management.

Objective and importance

Objective:

- Manage tasks by:
  - Prioritizing
  - Improving response time
  - Improving system throughput
- Allocate resources:
  - Processor cycles
  - Memory

Importance:

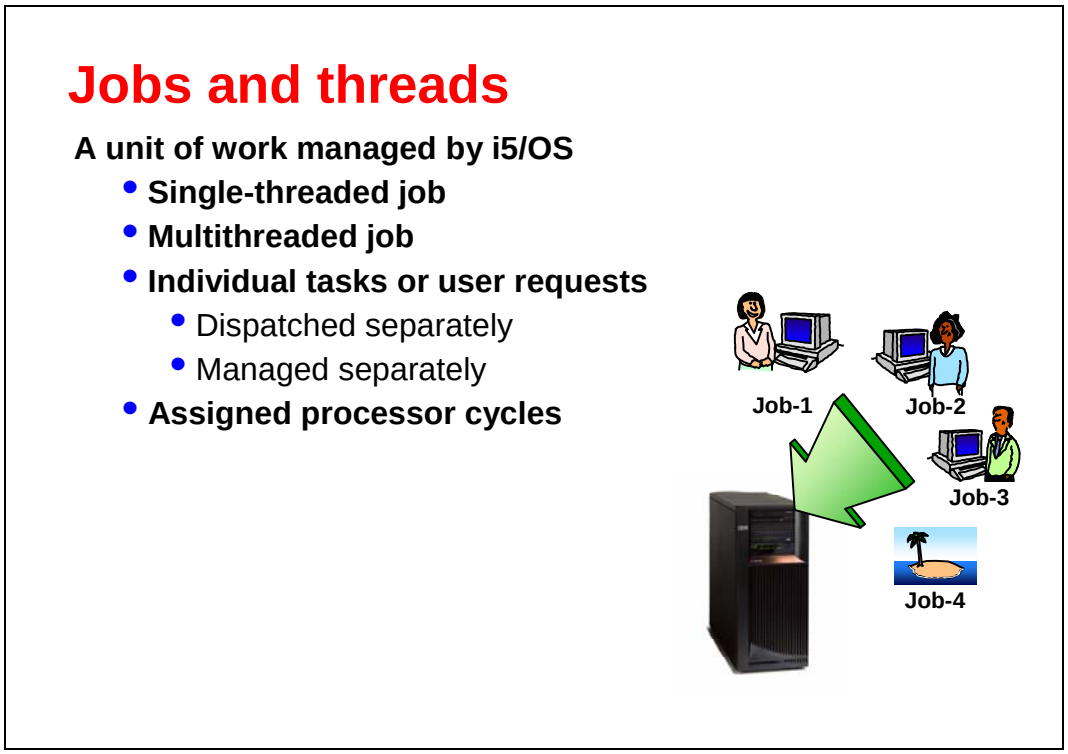
- Better manage iSeries system
- Quickly address problems

Objective and importance

Work management encompasses the fundamental issues of how work gets done within the iSeries system. Although the terminology might be unfamiliar, the concepts related to what is going on within the system will be apparent. Work management is also important so that you can optimally utilize the system's resources to gain the most efficient throughput of applications on the system. As you will see, a basic understanding of these concepts will be essential for procedures such as debugging an application and performance tuning.

i5/OS jobs

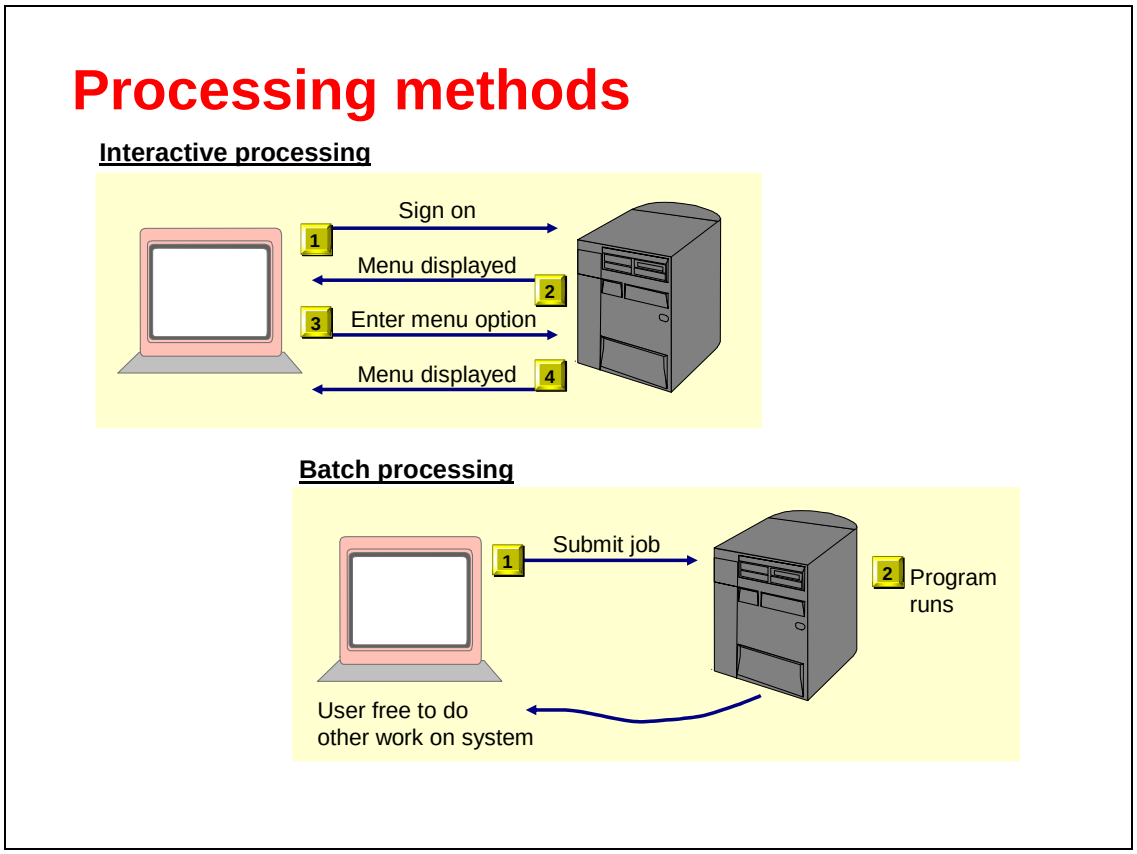
It is important to understand the nature of the various jobs that run in an IBM i5/OS® operating system on an iSeries system. This section of the course will establish these job concepts.



Jobs and threads

As mentioned earlier, a job is the unit of work execution within the i5/OS operating system. A job is analogous to a process on a UNIX® system or a task in a Microsoft® Windows® operating system.

Jobs may be single-threaded or multithreaded. Examples of multithreaded jobs are those running Java or IBM Lotus® Domino® programs. Threads of execution, also called tasks, are processed independently of each other. On multiprocessor systems, separate threads within a job may be dispatched to different processors.



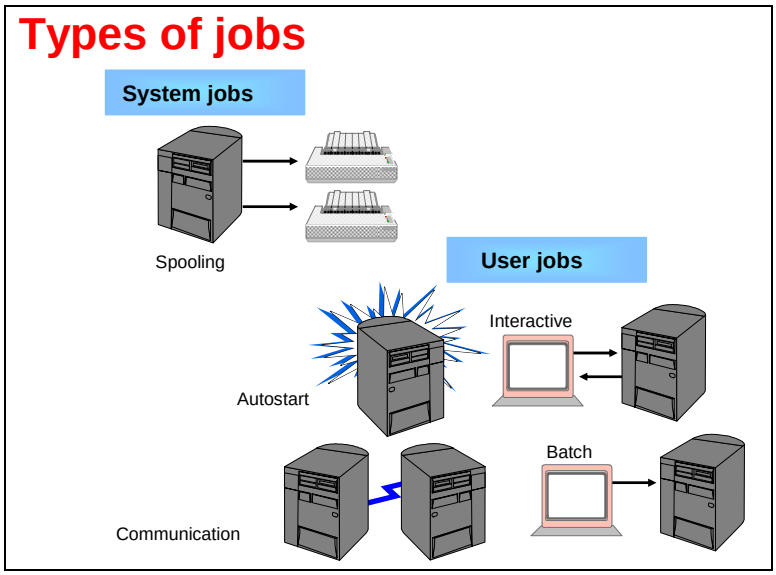
Processing methods

The i5/OS operating system does two types of processing: interactive and batch, as is illustrated in this diagram.

Interactive processing utilizes two-way communication between a display station user and the system. It is similar to a conversation. First, the user types something into the system, and then the system displays a response back to the user. In the example shown here, the user signs on to the system, and the feedback is a menu display. The user chooses an option from that menu, and the system responds accordingly.

Batch processing requires only that you send information to the i5/OS operating system for processing. There are numerous ways in which batch jobs are submitted to the system. For instance, you can submit a batch job via a command line. Client-side requests, such as those submitted via iSeries Navigator, are handled as batch jobs. Once a batch job is submitted, you are free to do other work.

The i5/OS operating system is designed to handle interactive and batch processing simultaneously.



Types of jobs

Jobs can be broadly characterized as either system jobs or user jobs. System jobs perform system-related functions, such as starting subsystems. (We will discuss subsystems later in this unit.) Conversely, user jobs perform work directly on behalf of a specified user.

An autostart job is one that is started automatically when its associated subsystem is started. Autostart jobs typically do such things as setting up for (or cleaning up after) an application, performing backups of data files, or starting devices.

A batch job is a predefined group of processing actions that is submitted to the system. Batch jobs run in the system background, freeing the user who submitted the job to do other work. A batch job is analogous to a UNIX daemon process. The job requires no interaction on the part of the user once it has been submitted. Batch jobs are typically lower priority jobs.

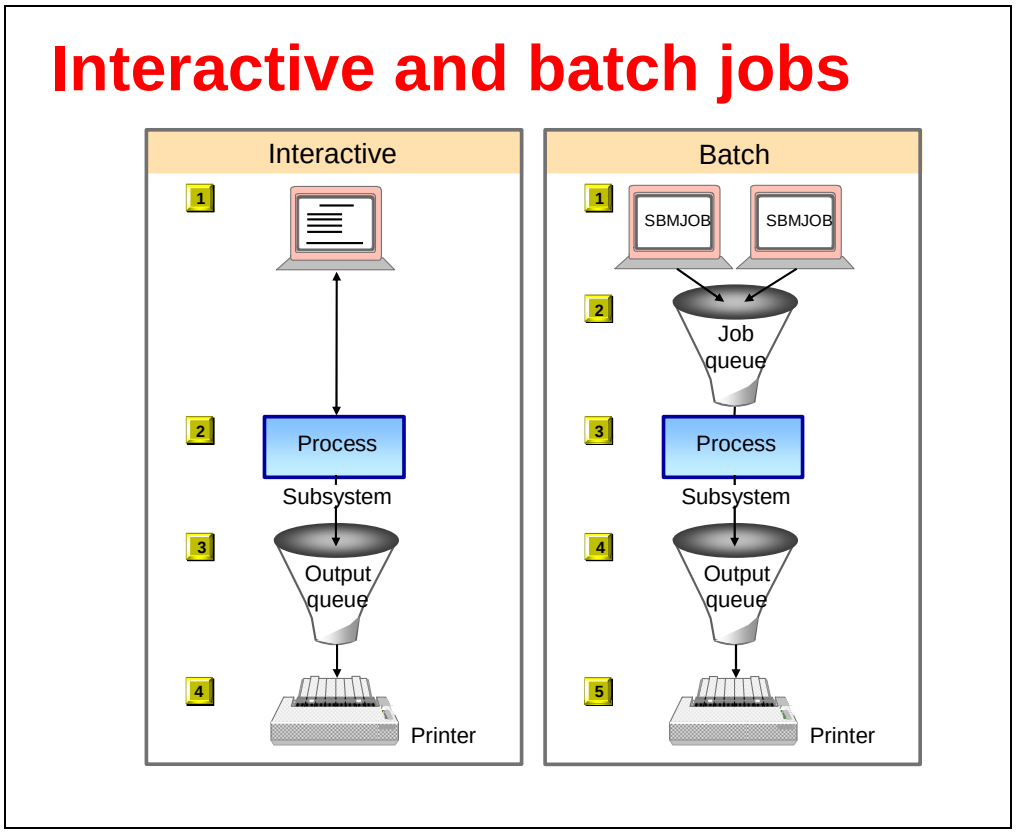
Interactive jobs are workstation jobs that start when a user signs on the workstation and end when the user signs off the workstation.

Communication jobs are started when a program start request is received from a remote system. For performance reasons, instead of starting a communications job each time a program start request is received, you can configure a prestart job to handle a program start request from a remote system.

A prestart job is a batch job that starts before a work request is received. The two types of prestart jobs:

- **Prestart communications:** The is a communications batch job that starts running before a remote system sends a program start request.
- **Prestart batch:** This is a batch job that starts before a work request is received.

Spooling jobs are system-provided print programs (writers) that run similar to batch jobs and print spooled printer output.



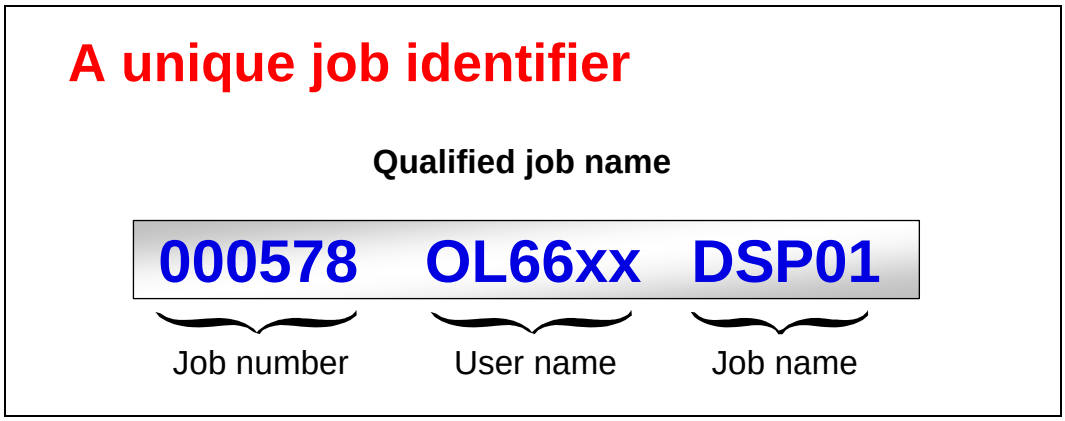
Interactive and batch jobs

An interactive job is active from the time a user signs on a display station until that person signs off. While signed on, an interactive job typically goes through the following steps:

1. A user signs on to a display station.
2. The i5/OS operating system runs the job and can generate printed output that is placed in a spooled file and put on an output queue. An output queue contains reports waiting to print.
3. This job waits on the output queue until prior jobs are printed.
4. The spooled output file is printed.
5. The user either signs off the system, which ends the interactive job, or stays signed on, in which case the job keeps running.

A batch job, on the other hand, runs in the background. It typically moves through the following steps:

1. A batch job can be initiated via the Submit Job (SBMJOB) command. Client-side requests, such as those submitted via iSeries Navigator, are also handled as batch jobs. A batch job is submitted to a job queue. The i5/OS operating system places the jobs on the job queue in the order in which they were received.
2. The job waits on the job queue until the subsystem that is monitoring the job queue has the resources needed to run the job. Then, the job runs and generates printed output that is placed in a spooled file and put on an output queue. The output queue is similar to the job queue, containing printed output waiting its turn to print. The spooled output can then be printed.



A unique job identifier

Each job has a system-unique identifier, called a qualified job name, that consists of three parts (or qualifiers):

- A unique six-digit job number assigned by the system
- A user name, that is the user ID under which the job was started
- A job name set by the user for submitted jobs or assigned as the name of the device used to connect for interactive login jobs

The qualified job name, comprised of these three parts, is analogous to a UNIX process ID. It can simplify searching for a job. For example, a complete qualified identifier will look similar to the following:

- WRKJOB JOB(000578/OL66xx/DSP01)

If all three qualifiers are not known, you can still search using only some of the qualifiers. For example, you can search by user name and job name, or by job name only:

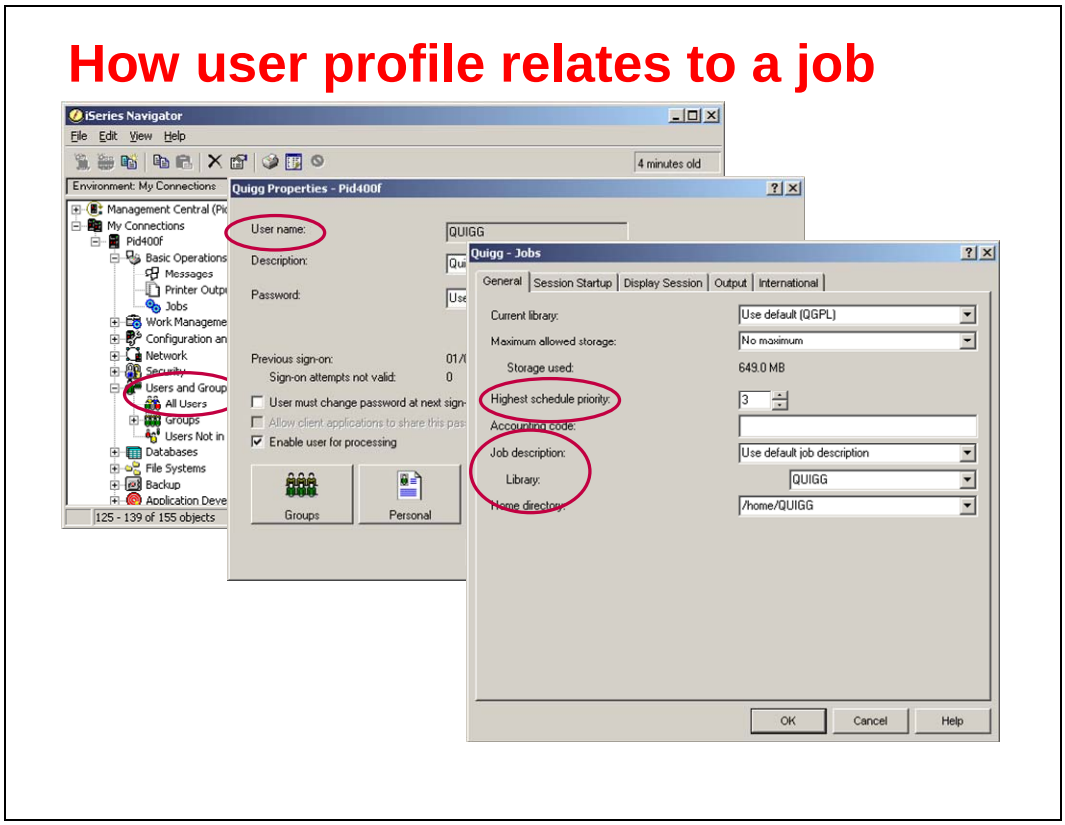
- WRKJOB JOB(OL66xx/DSP01)
- WRKJOB JOB(DSP01)

There are several commonly used commands for finding jobs, such as:

- Work with Active Job (WRKACTJOB)
- Work with User Job (WRKUSRJOB)
- Work with Submitted Job (WRKSBMJOB)

Jobs can be displayed with iSeries Navigator. (**Note:** Several iSeries Navigator screens will be shown later in this course, but for now, it will help to understand that it is a treelike device for drilling down through the iSeries system's resources.)

To view a job using iSeries Navigator, under **Basic Operations**, select **Jobs**. When this function is first opened, the default settings will dictate that active jobs and those on the job queue of the current user are displayed. It is important to note that iSeries Access jobs, including iSeries Navigator functions, are not started using the user's profile, but rather, the QUSER command is used. Therefore, unless a batch or interactive job is running, an empty panel might appear. By default, a user will only see the jobs specific to that user. To change this, select **View** > **Customize this view** > **Include** from the pulldown menu, and then change the **User** field to ***ALL**. Click **OK** to display a listing of all jobs. To display specific job details, double-click it.



How user profile relates to a job

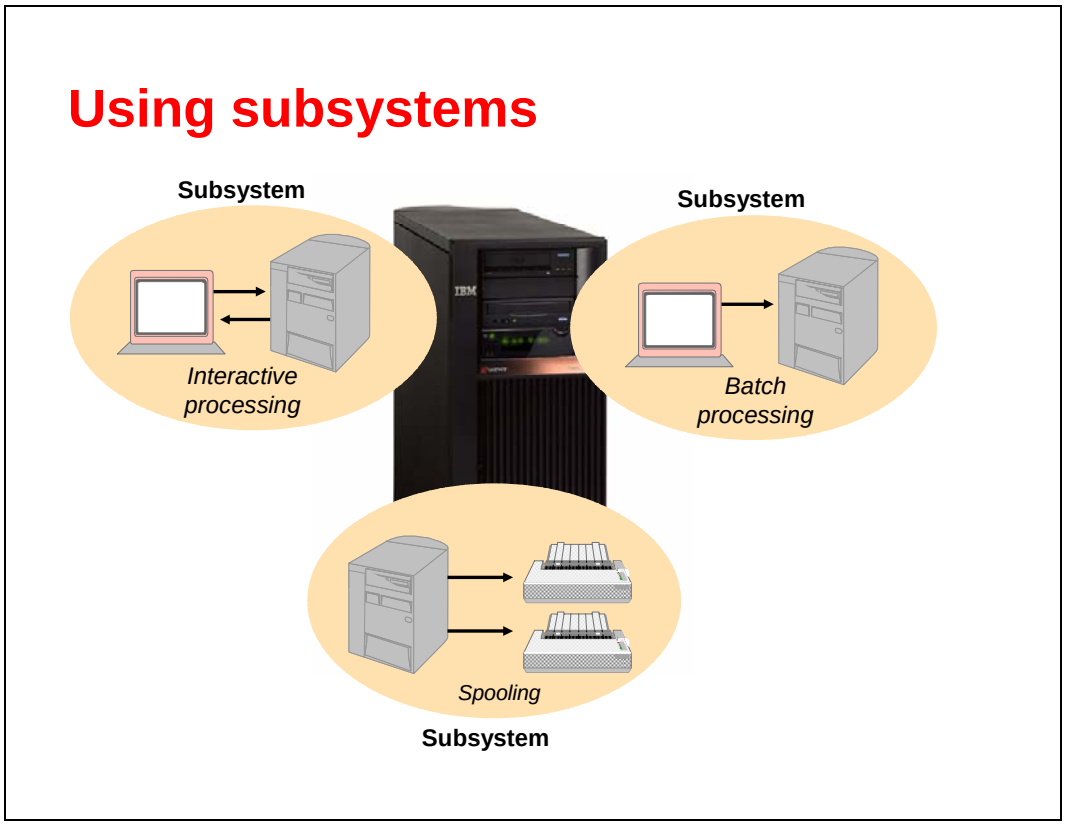
Every job run within the i5/OS operating system is associated with a specific user. A user's profile contains a great deal of information about the user and which types of jobs he or she can run. Likewise, system jobs are associated with system-supplied user profiles.

The user's initial menu and program are specified here. The job description, output queue, and printer for this user can also be specified in the profile.

Subsystems and storage pools

Subsystems and storage pools

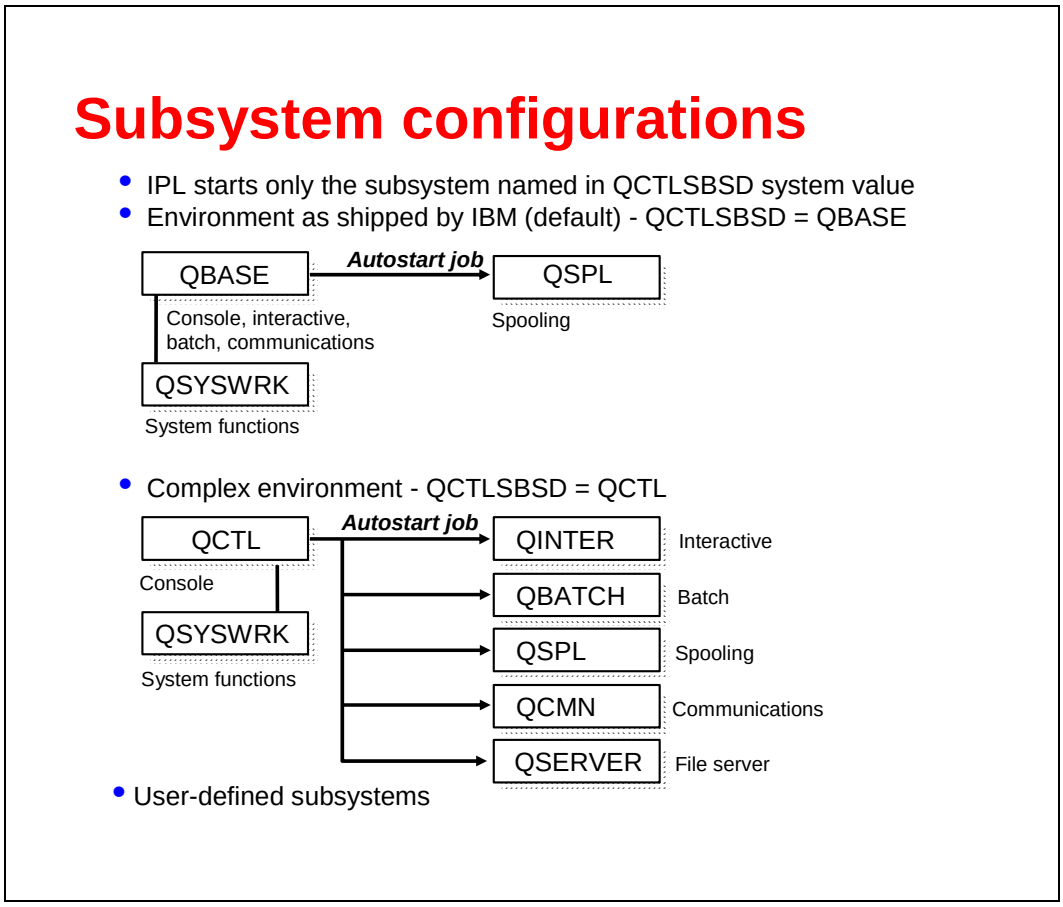
The i5/OS operating system utilizes subsystems and storage pools to execute jobs. These two topics will be discussed in this section of the course.



Using subsystems

From the earlier discussion about different types of jobs, you might correctly assume that any particular job will run most efficiently on a system designed to execute only that specific type of job. The i5/OS operating system can contain one or more single, predefined operating environments, called a subsystem, through which the system coordinates work flow and resources. As each job starts, it is assigned to a subsystem.

By using different subsystems, each with its own unique operating characteristics, i5/OS work management simulates an environment where it appears that each system was designed specifically to run only the type of job running in that subsystem.



Subsystem configurations

Predefined subsystem descriptions that can be used to provide the environments most needed are included when IBM ships the iSeries system.

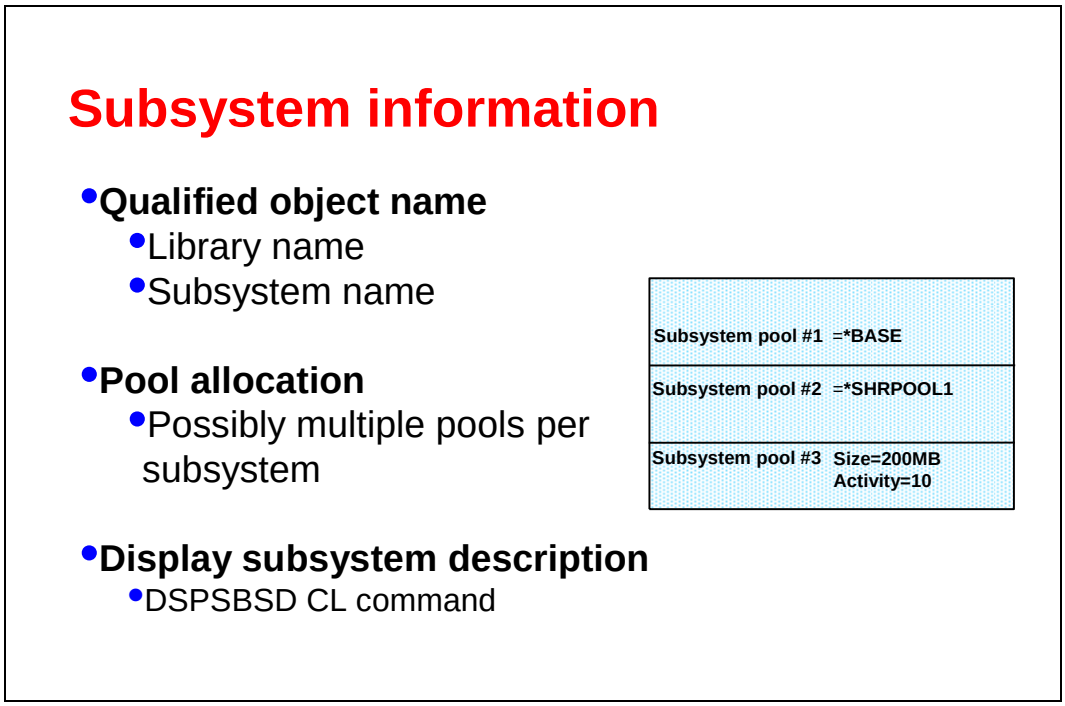
The default configuration consists of a controlling subsystem called QBASE, a system function subsystem called QSYSWRK, and a spooling subsystem called QSPL. In this environment, all jobs on the system run under the control of the QBASE subsystem. However, a few system and spooling jobs run under the control of QSYSWRK and QSPL, respectively.

A more complex configuration is also provided that can be used instead of the default. It consists of QCTL as the controlling subsystem, QSYSWRK for certain system functions, QINTER for interactive jobs, QBATCH for batch jobs, and QSPL for spooling.

Additionally, the QCMN subsystem can be used for communications, and QSERVER can be used for file serving.

IBM also supplies application-specific subsystems, such as QHTTPSVR for the IBM HTTP Server.

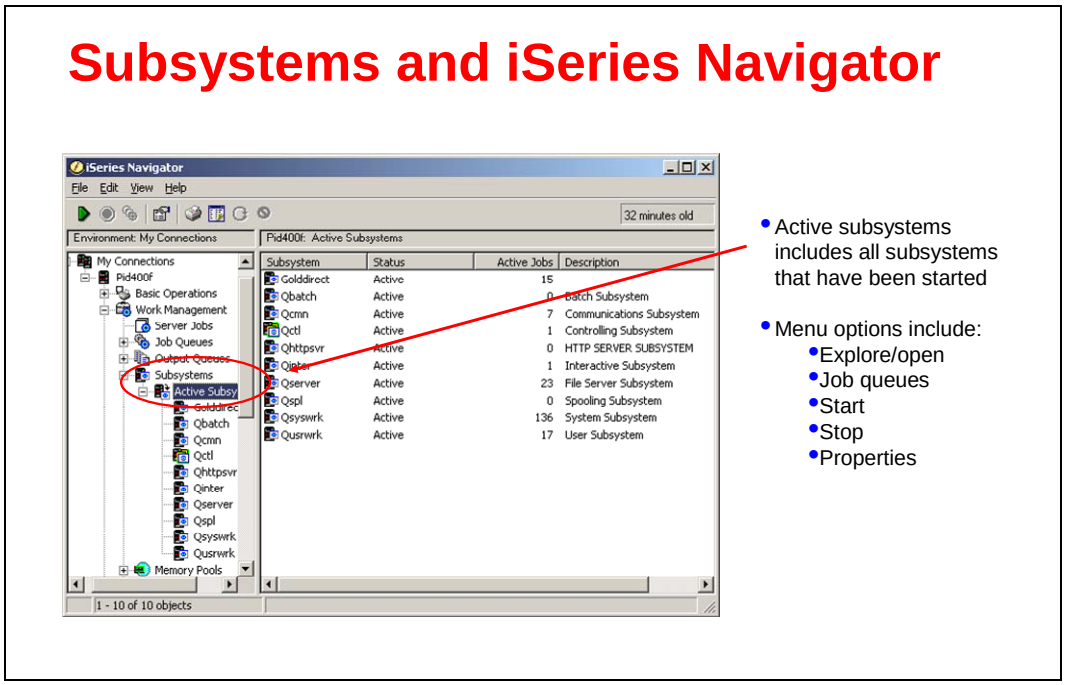
Finally, you can define a subsystem to run a specific application. This is a convenient way to group system resources that will be dedicated only to jobs related to specified application.



Subsystem information

Each subsystem is uniquely identified using a qualified object name.

The amount of work that can be done in a subsystem can be controlled by changing the number and size of the pools associated with it. Increasing the size of the pools will allow more work to be done in that particular subsystem. Using shared storage pools, which are discussed in greater detail later in this course, will allow the system to distribute jobs for interactive users across multiple subsystems, yet still allow the jobs to run in the same storage pool.

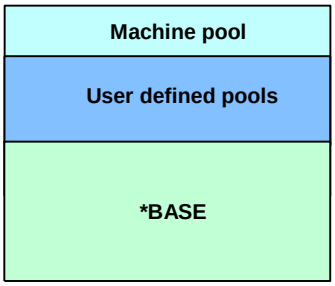


Subsystems and iSeries Navigator

The subsystem-related information can be displayed with the text-based interface or iSeries Navigator, as shown here. **Active Subsystems**, which is highlighted in the left pane, includes all subsystems that have been started. Using the menu options, you can explore the subsystem, view job queues, start and stop jobs, or look at various properties.

Main storage pool

Logical subdivisions of main storage
- **Shared pools**
  - *MACHINE
  - *BASE
  - *INTERACT
  - *SPOOL
  - *SHRPOOL1-n
- **Private pools**
  - Assigned to a specific subsystem
- **Up to 64 memory pools**



Activity level
- Maximum active tasks (threads) in subsystem
- "Throttle" mechanism

Main storage pool

A storage pool, also called a memory pool, is a logical division of memory that is reserved for processing a job or group of jobs. There are two types of pools in a system: shared and private. A shared storage pool is one in which multiple subsystems can run jobs, whereas a private storage pool allows jobs from only a single subsystem.

The machine pool (*MACHINE) and base pool (*BASE) are considered to be special shared pools. The machine storage pool is used for highly shared machine and i5/OS licensed programs; and it provides storage for jobs the system must run that do not require your attention. The size for the machine pool is specified in the system value QMCHPOOL. No user jobs run in this storage pool.

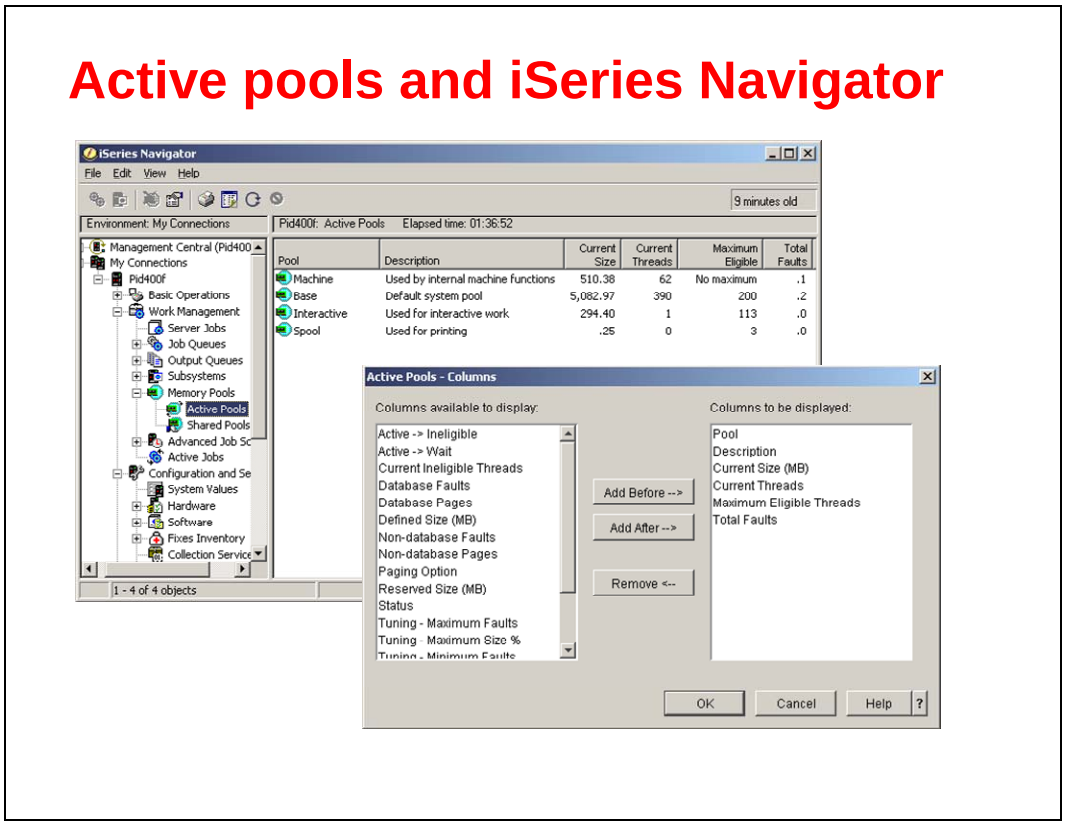
The base storage pool contains all unassigned memory on the system. In other words, it has all the storage that is not allocated to any other pool.

Activity levels determine the number of threads that can be active in a storage pool at the same time. Allowing more than one active thread at once can increase resource efficiency. This works by making use of downtime in thread processing. Occasionally, while handling a thread, a program waits for a system resource or a user response. During such waits, use of the storage pool is given up and another thread can take its place. When more threads are started than can run at the same time, based on activity level controls, the excess threads wait for the processing unit.

Commonly used functions

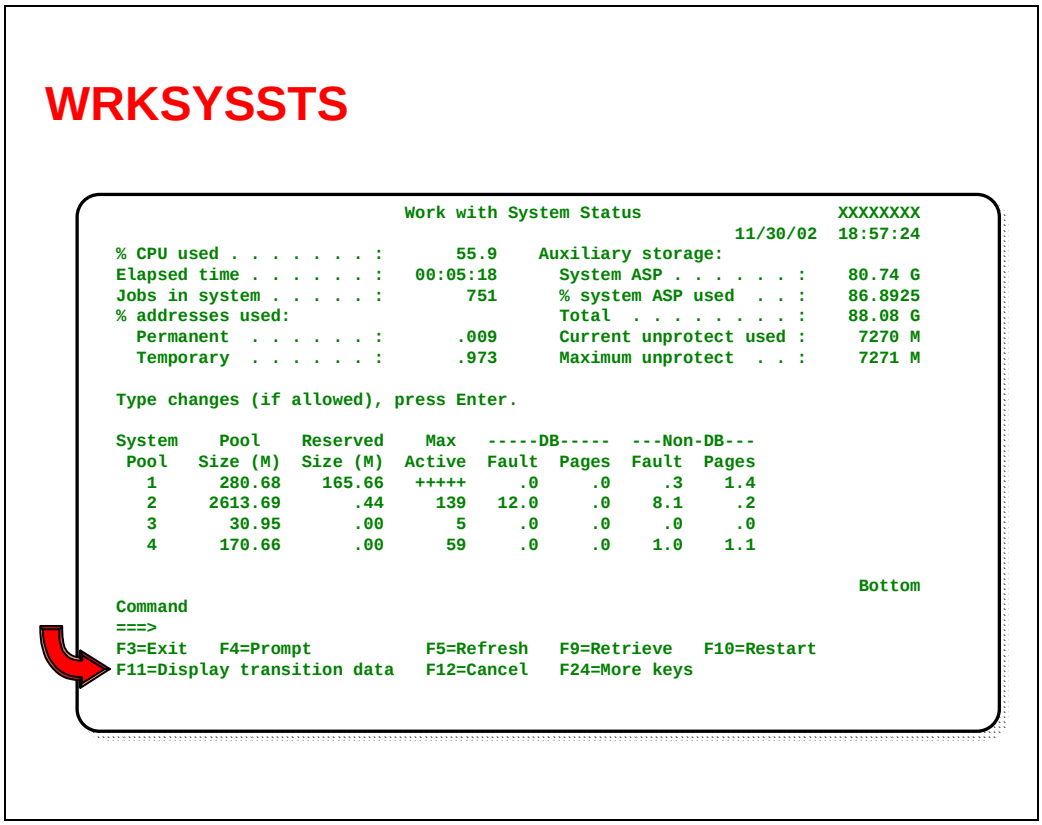
Commonly used functions

You will also need to understand the most frequently executed functions to manage jobs in an i5/OS operating system. This next section of the course will explain these functions to you.



Active pools and iSeries Navigator

iSeries Navigator can be used to view and display several attributes for each storage pool. Once you have displayed the active pools, select **View** > **Customize this view** > **Columns**. Next, simply select the attributes to display. Much of the information is the same as that shown in the text-based interface using the Work with System Status (WRKSSYSTS) CL command, as shown on the following screen.



WRKSYSSTS

To access the Work with System Status display, enter **WRKSYSSTS** on a command line. Alternately, you can:

1. Enter option 3 **General system tasks** on the OS/400® main menu
2. Choose option 2 **Status** on the general system tasks menu
3. Select option 6 **Work with system status** on the status menu

The Work with System Status display is a powerful workload tuning tool. The following are tips for tuning your system:

- If CPU utilization is continually less than 80%, you might need to tune the system if no batch jobs are running. (**Note:** This percentage is meaningful only if the elapsed time is at least five minutes.)
- Failure to manage spooled files, which are typically job logs, can cause the number of jobs in the system to range into the thousands. The more jobs the system has to manage, the less productive work it can do.
- The percentage of permanent and temporary addresses used must never exceed 90%.
- For optimum performance, the auxiliary storage percentage used must be kept below 70%.

Work with active jobs

- **Active jobs by subsystem identification**
  - Job name
  - User
  - Job number
- **Type**
  - Interactive
  - Batch
  - Batch immediate
- **Runtime information**
  - Pool
  - Priority
  - CPU usage
  - Threads information

Working with active jobs

You can view, modify, and refresh information about currently running jobs using either iSeries Navigator or the text-based CL command Work with Active Jobs (WRKACTJOB).

The Work with Active Jobs display can be used to find information about any job on the system. When only a small number of jobs is running after an elapsed time of more than one minute, and the **CPU %** column for any one job is significantly higher than other jobs, then it is clear which job is consuming the CPU. It might mean that the job is in a loop or is logging many errors. On a lightly loaded system, high CPU utilization might not be abnormal as some jobs typically use a great deal of processor time.

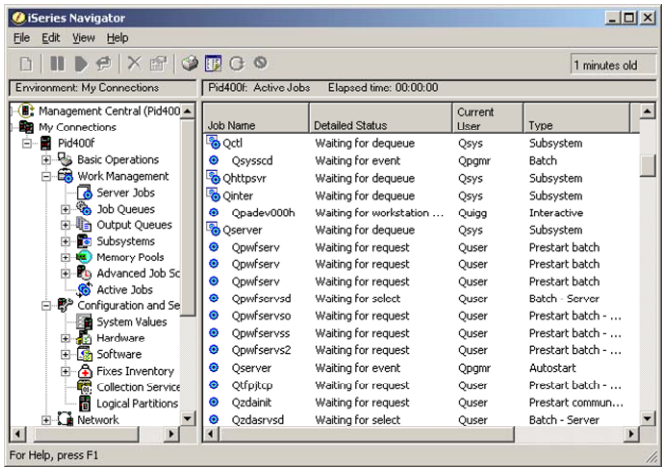
If you suspect that a job is having problems, put it on hold by using option 3 on this display while the problem is investigated further. Depending on the results, you can then either release or end the job with this display.

To access the Work with Active Jobs display:

1. Enter option 3 **General system tasks** on the OS/400 main menu
2. Choose option 1 **Jobs** on the general system tasks menu
3. Select option 2 **Work with all active job statistics** on the Job menu.

You can also enter **WRKACTJOB** on a command line.

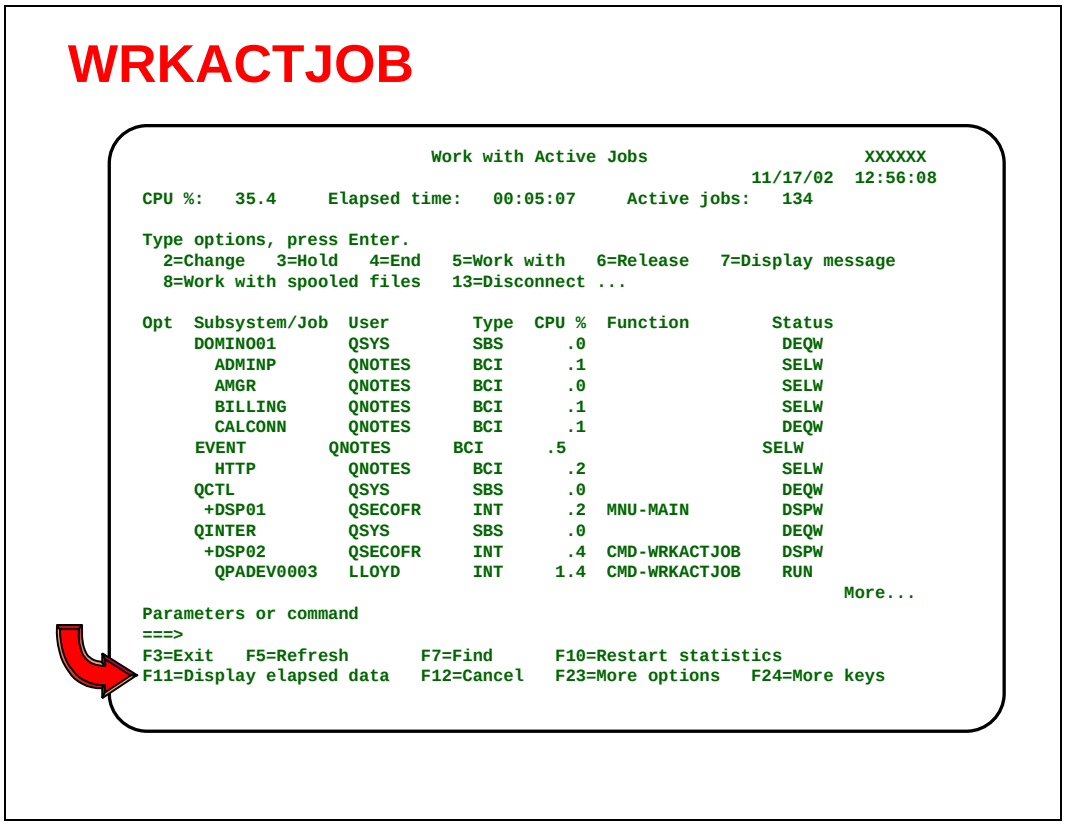
Active jobs



- More function and flexibility than text interface
- Customizable
  - Sort
  - Include
  - Columns
  - Shortcuts
- Grouped by subsystem
- Menu options
  - Call stack
  - Locked objects
  - Open files
  - Library list

Active jobs

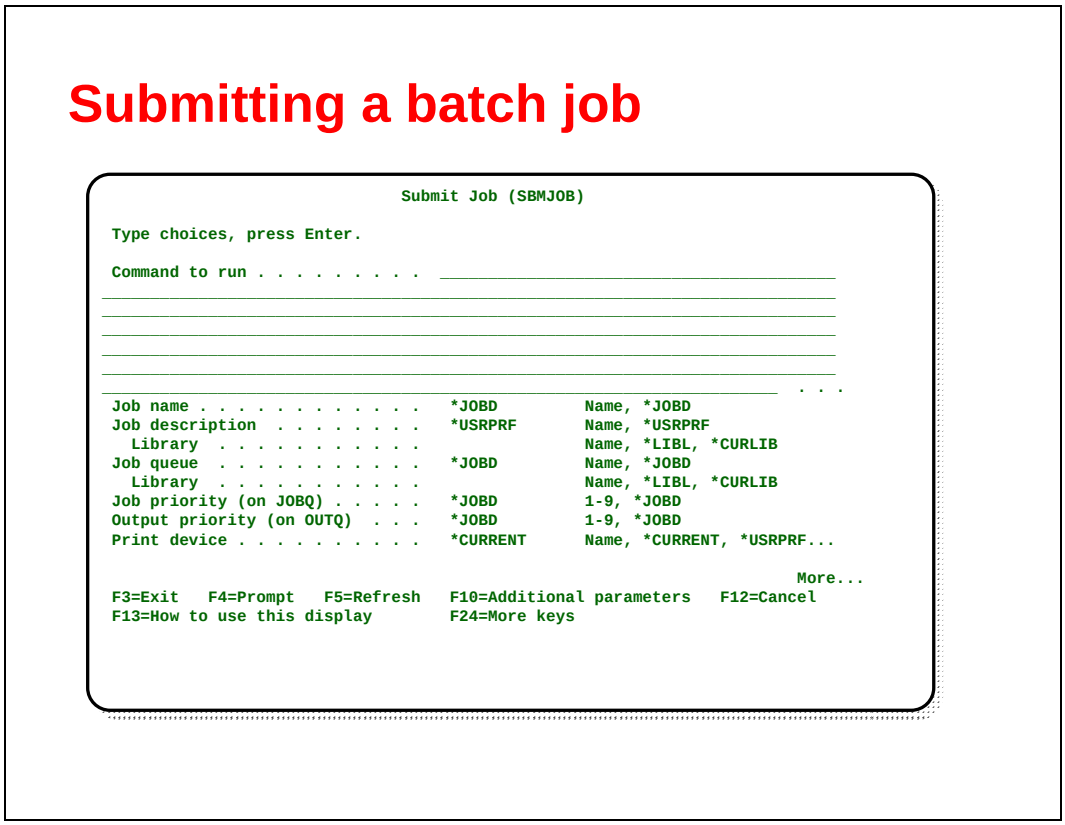
Active jobs can be viewed using iSeries Navigator. From the left side of iSeries Navigator, expand **Work Management** and select **Active jobs**. Once you have displayed the active jobs, select **View** > **Customize this view** > **Columns**. You can now select the attributes to display.



WRKACTJOB

The Work with Active Jobs display can be used to find information about any job on the system. By default, the jobs are sorted by subsystem, but there are numerous other ways that they can be sorted. Type **WRKACTJOB** and prompt using F4.

Within the display, repeatedly pressing F11 will display various columns of data.



Submitting a batch job

As previously discussed, a job queue is an object that contains a list of batch jobs waiting to be processed. Each job has a priority (1 = highest; 9 = lowest) when it is submitted to a job queue, and they are processed in first-in, first-out order according to their priority.

To submit a batch job from an interactive display session, use the Submit Job (SBMJOB) command.

A batch job has many characteristics that can be assigned, such as:

- The queue to which it is submitted
- The priority at which it is to run
- The output queue to which its associated files will go
- The priority at which the files are placed in the output queue
- The print device that will be used

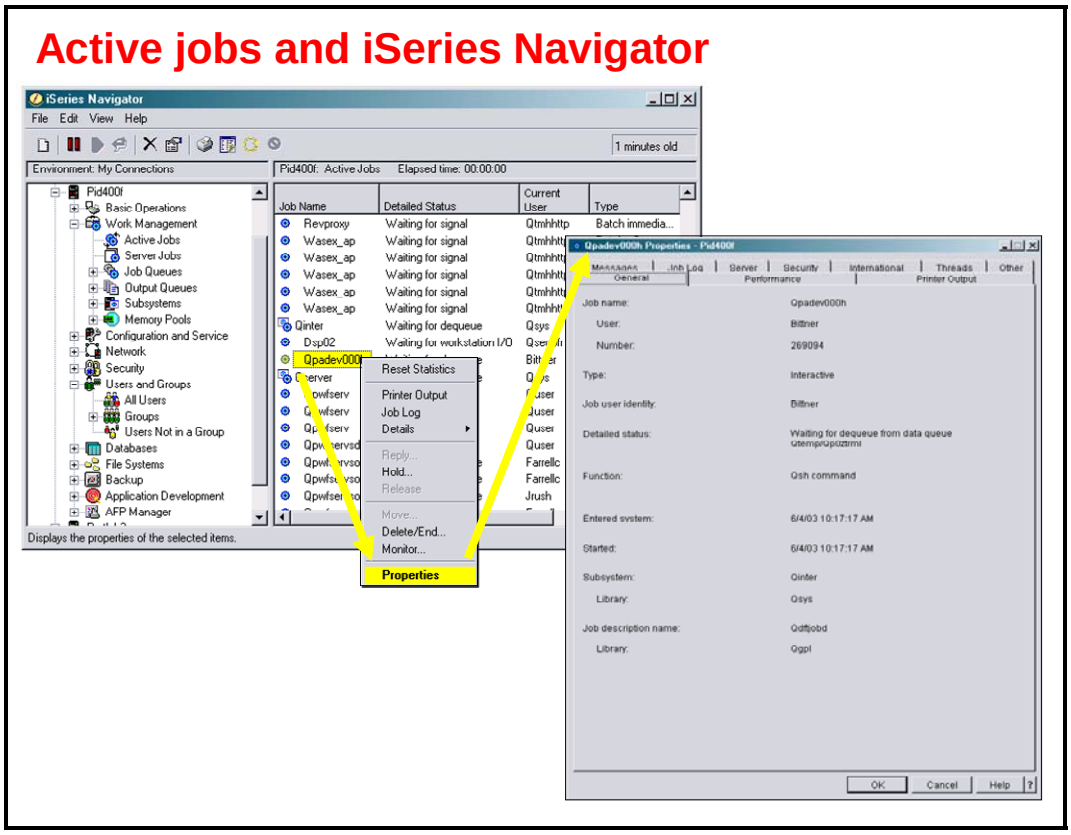
If you use the SBMJOB command, there are parameters for each of these characteristics. There is also a parameter to specify which job description to use. Unless specified explicitly on the command, the submitted job gets its characteristics from the job description.

If nothing is specified, the system will use the job description in the user profile. The default is QDFTJOBD, which specifies that batch jobs are to be submitted to QBATCH in library QGPL. This job queue runs its jobs in subsystem QBATCH or QBASE, depending on how your system is set up.

Debugging and diagnostics

Debugging and diagnostics

Of course, debugging and diagnostics are critical to all work management efforts. The tools available for this purpose are explained next.

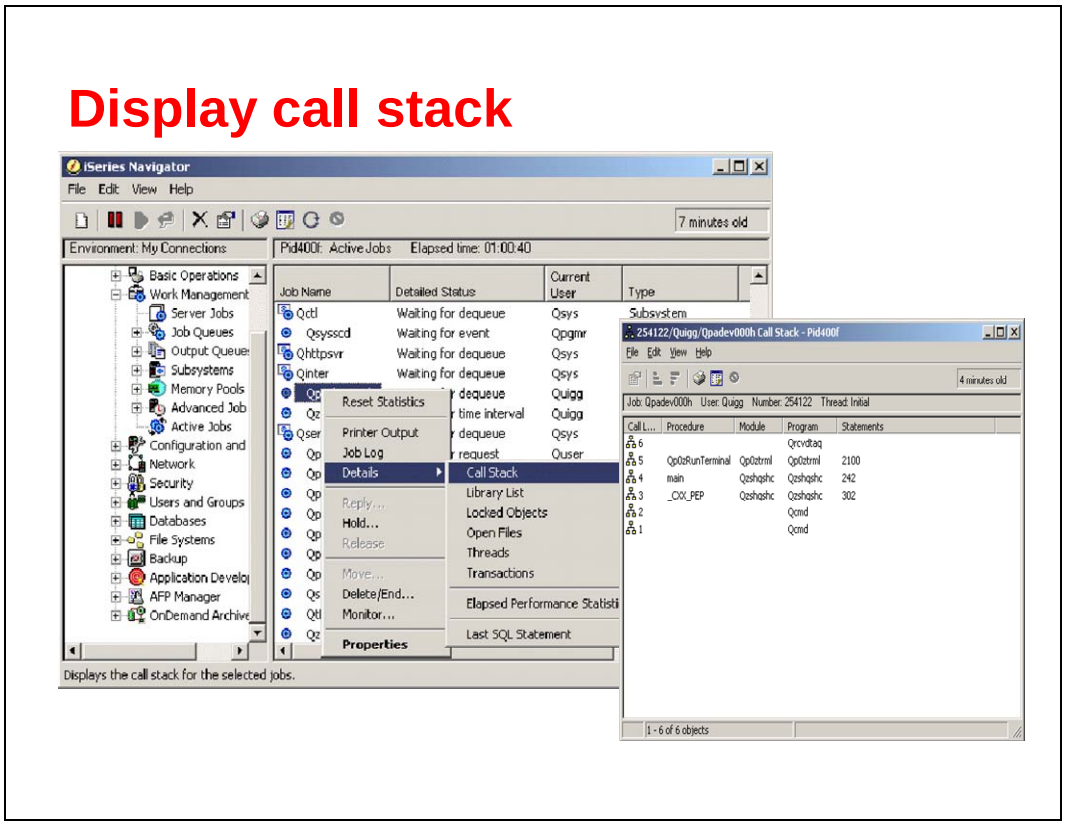


Active jobs and iSeries Navigator

From the left side of iSeries Navigator, expand **Work Management** and select **Active jobs**. On the right side of the iSeries Navigator screen, when you right-click a job, a context menu will be shown. From this menu, you can change the attributes and hold, release, or end the job. You can also select **Properties**, as shown in this screen shot. Other functions are also available. There is extensive information about a job to help determine what it is currently doing or what it has done. The fully qualified job name (containing the job name, user, and number) appears near the top. Nearly everything that the system knows about this job can be accessed from this context menu.

The Work with Job menu, accessed via the WRKJOB CL command, presents similar information. In addition to the CL command, the WRKJOB menu can be accessed from the OS/400 main menu.

1. Enter option 1 **User tasks**.
2. From the **User Tasks** menu, enter option 1 **Display or change your job**.

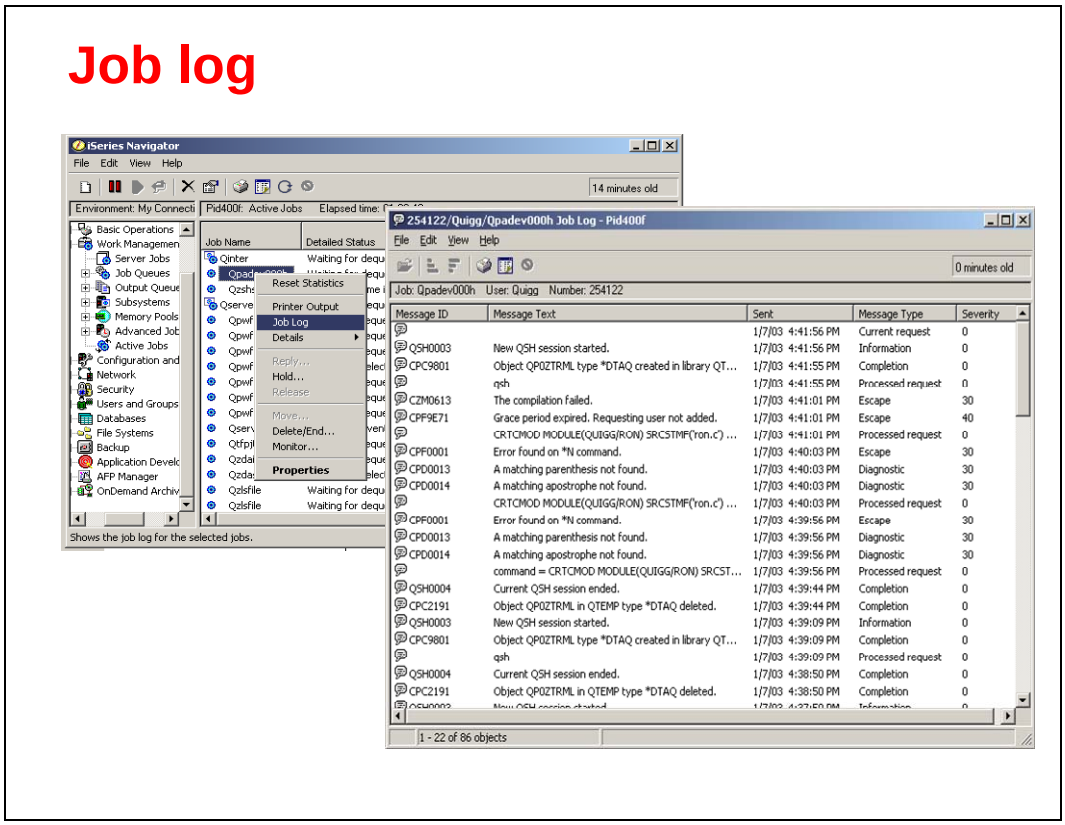


Display call stack

The call stack includes information for all programs and procedures for the specified job. If the call stack changed during processing, the displayed information might not be correct. Press **Refresh** to update the call stack.

The default view includes columns showing the procedure name and the program name in the call stack. For integrated language environment (ILE) programs, it can be useful to show additional information so that the view can be customized to include activation group name, module name, and various other information.

The high-level language statement identifier is included if the program or procedure contains debug information. For ILE procedures, more than one statement ID can exist.

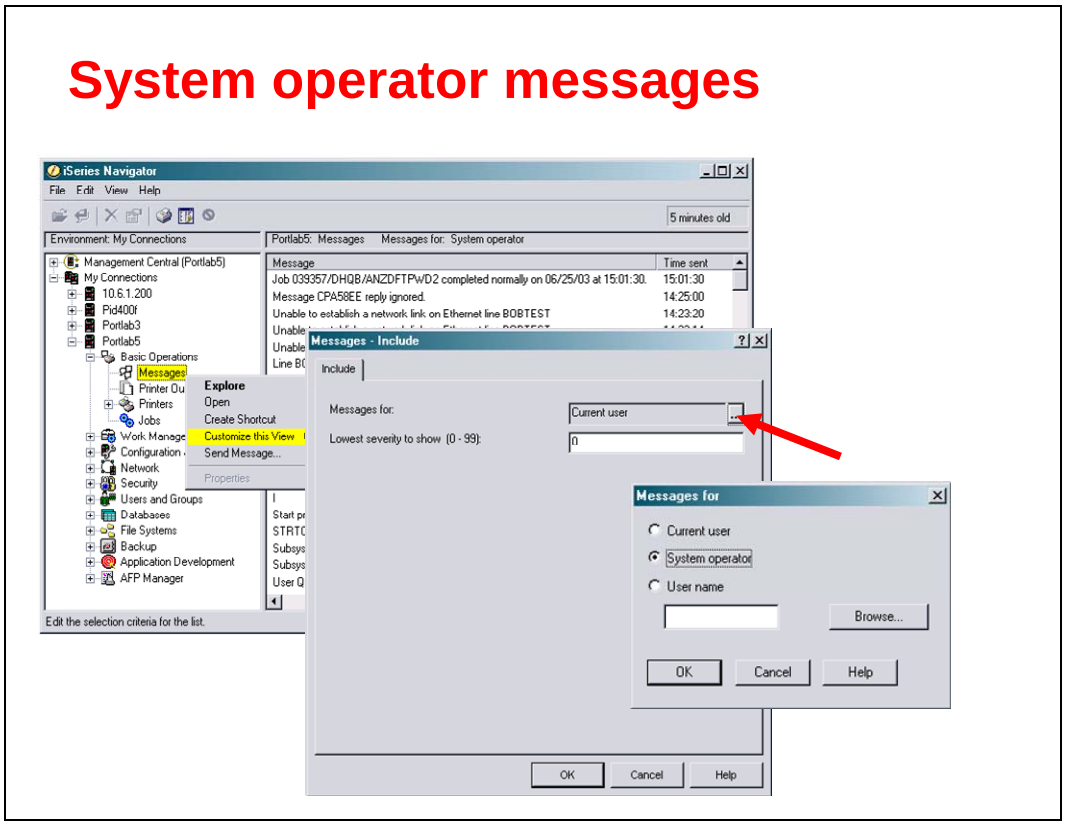


Job log

The i5/OS operating system keeps a detailed record of the activity of each job. This trace (of the commands run and the system messages a job receives) is called a job log, and it can be a useful aid in problem determination. From iSeries Navigator, simply double-clicking the job will allow you to view its job log. A log can be viewed as long as the job is active in the system. Regardless of the method used, if insufficient detail is shown to determine precisely what the job has done, right-clicking will display all available details.

The overlaying screen shot seen here shows the job log details.

The log for a batch job is created when the job is placed on a queue. It contains a trace of all of the job's activity for as long as it is active. By default, if a batch job completes successfully, its log is deleted from the system. If the job fails, the job log is spooled to an output queue.



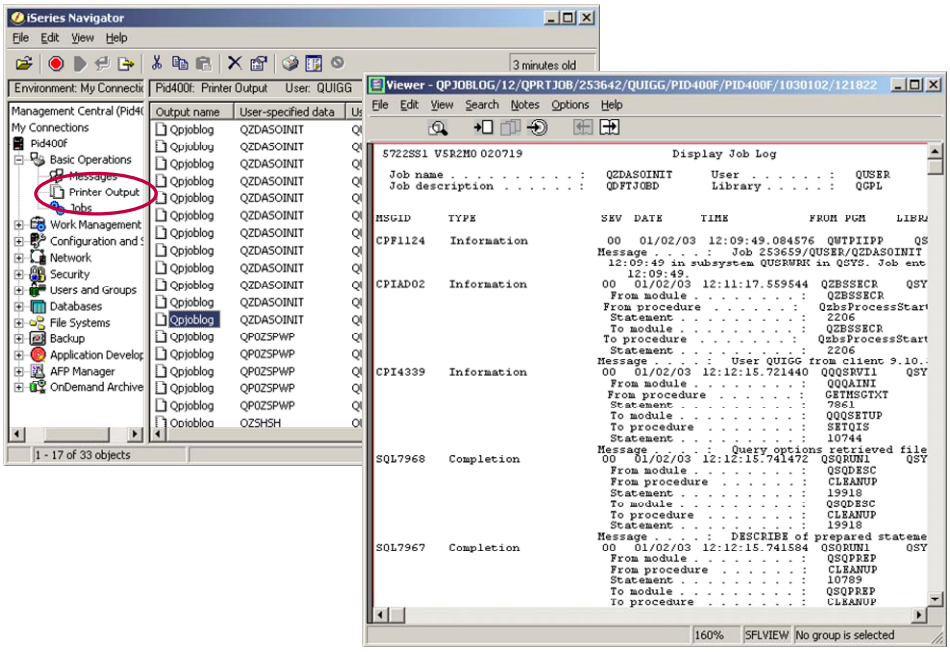
System operator messages

Often, the only thing wrong is that there is an unanswered message in one of the system message queues. When in doubt, check for messages here.

The system operator message queue (QSYS/QSYSOPR) is typically used to collect system-management-related messages.

This screen demonstrates using iSeries Navigator to display these messages.

Displaying and printing spooled files



Displaying and printing spooled files

Spooled output is a listing that is queued to be printed. Spooled files associated with multiple output queues can be shown. The status of each, the number of pages per copy, and the number of copies to be printed are displayed. Although you can control which spooled files are displayed, the default is for the current user.

When you double-click one of the items in the right-hand pane, iSeries Navigator will activate the advanced function presentation (AFP) viewer. This will give you a view of the contents of the spooled file on the screen.

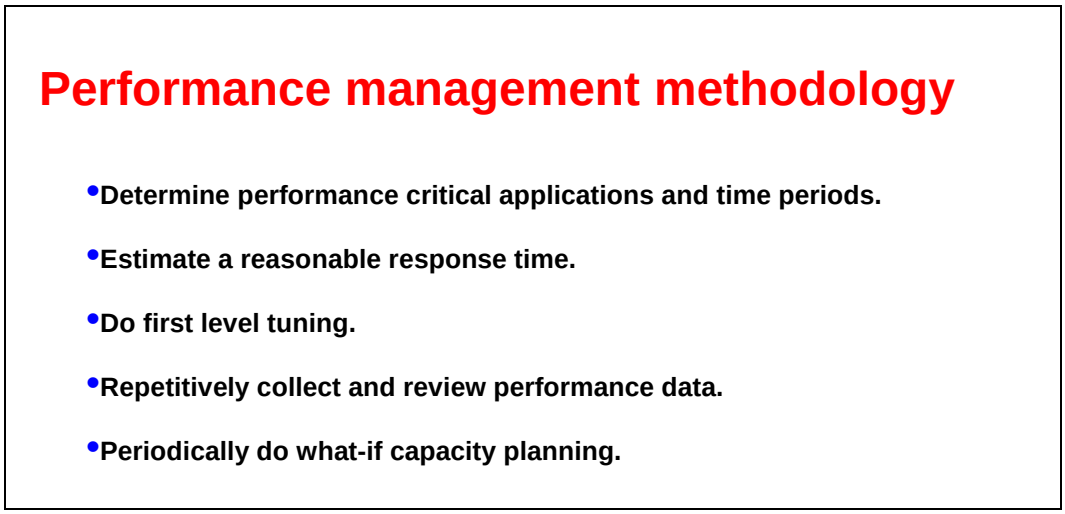
By right-clicking one of the spooled files, you will see a context menu that allows numerous options. These options include the ability to perform the following functions: open or display the file, hold, release, print next, send, copy, cut, move, and delete the spool files.

Selecting the **Properties** option from the context menu allows you to change many of the properties of the spooled files. For example, some of the options that can be changed include the output queue, printer, starting and ending page to print, form type, and number of copies to print. As a practical matter, to print a spooled file (or, as another example, if you want to send it as an attachment) it is easy to simply drag-and-drop it onto your desktop or into a folder.

Performance management methodology and tooling

Performance management methodology and tooling

Particular methodologies and related tooling will help you to better manage the flow of jobs through the i5/OS operating system and to optimize the performance of an application within a single job. You will be learning about these as you proceed through this last section of the course.



Performance management methodology

If a mixed workload is running within the i5/OS operating system, it is not likely that all applications require optimal performance. For example, the performance of overnight batch jobs might not be critical, provided they complete sometime between midnight and 6:00 a.m. However, interactive applications run by users during the business day might require millisecond response time.

The runtime characteristics of various applications within a mixed workload are more than likely different from one another. Disk-intensive applications will merit a review of disk arm utilization. The percent busy value ought not exceed 40%. Disk arm utilization can be viewed using the command WRKDSKSTS, or through iSeries Navigator by clicking: **My Connections** > **server-name** > **Configuration and Service** > **Hardware** > **Disk Units**. At this point you will be prompted for service tools sign-on. Then choose **All Disk Units**.

The information gathered here will show the average total system CPU utilization, the average total system interactive job CPU utilization, and the response time disk arm utilization.

First-level tuning

Check the following:

- **Memory pool size and activity level**
- **Number of subsystems within a memory pool**
- **Memory pool paging option (Expert cache system value)**
- **Minimize pool page fault rates**

First-level tuning

First-level refers to the first series of items that are to be reviewed when diagnosing a performance problem. Check each of the following and adjust as needed:

- **Memory pool size**: A high rate of faulting in a pool indicates that there is not enough memory, or that too many jobs are in the pool competing for memory. Adjusting the memory pool size in one of the following ways can prove beneficial:
  - Use System Tuner (which is a QPFRADJ system value) to automatically adjust memory pools and activity levels.
  - Manually change memory pool sizes; be sure to turn off the system tuner.

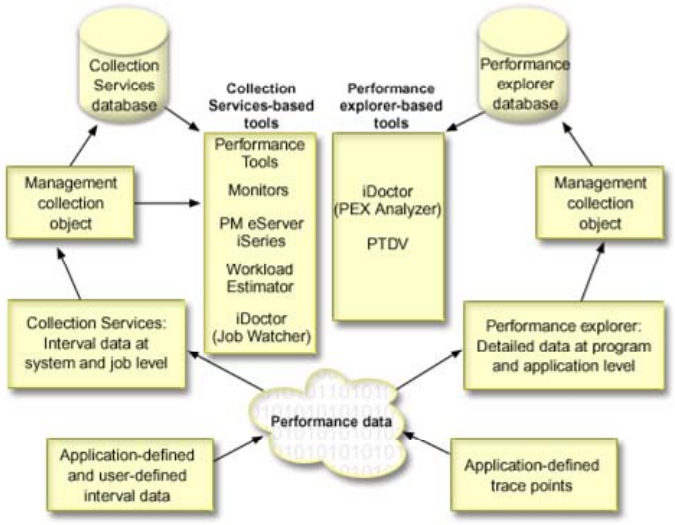
  Special considerations are made for the Base pool, which is the only memory pool that does not have a defined amount of memory. It contains whatever is not allocated elsewhere and has a minimum amount of memory required to run.
- **Number of subsystems within a memory pool**: Too many subsystems in a pool can have a negative effect on performance. It can be beneficial to create a separate pool for one or more subsystems.
- **Number of jobs (and threads) in a memory pool**: Too many jobs (or threads within those jobs) in a memory pool can negatively impact system performance.
- **Activity level too low**: If a job is **Waiting for activity level**, review memory pool usage as mentioned above. You can also increase the activity level for the pool.
- **Insufficient CPU resource**: View the CPU% column for active jobs. If the system is busy, the job might not be getting enough CPU resource. Consider changing the run priority of the job (**1** being the highest priority).
- **Memory pool paging option**: If the application is disk-intensive (that is, low CPU utilization and sufficient memory), consider turning on **Expert Cache**. To do this in iSeries Navigator, click **Work Management** > **Memory pool** > **Shared Pools**. Then, from the right iSeries Navigator panel, right-click the pool. Then click **Properties** > **Configuration** to set the paging option to **calculated**.
- **Low job run priority**: See *Insufficient CPU resource* above. Also, on a system with high CPU% and a job with low priority, consider a dynamic adjustment of job priorities (QDYNPTYSCD and QDYNPTYADJ system values).
- **Page fault rates are too high**: Fault rates between 20 and 30 per second are acceptable. Higher rates can be reduced by increasing the memory pool size. In some cases when storage is available, reducing this value even further can improve performance as well.

For more information on first-level tuning, such as performance issues, refer to the **Resources** section of this course.

Performance management tools

Performance collection functions:

- Collection services: collects interval data at the system and job level.
- Performance explorer: collects detailed data at the program and application level.



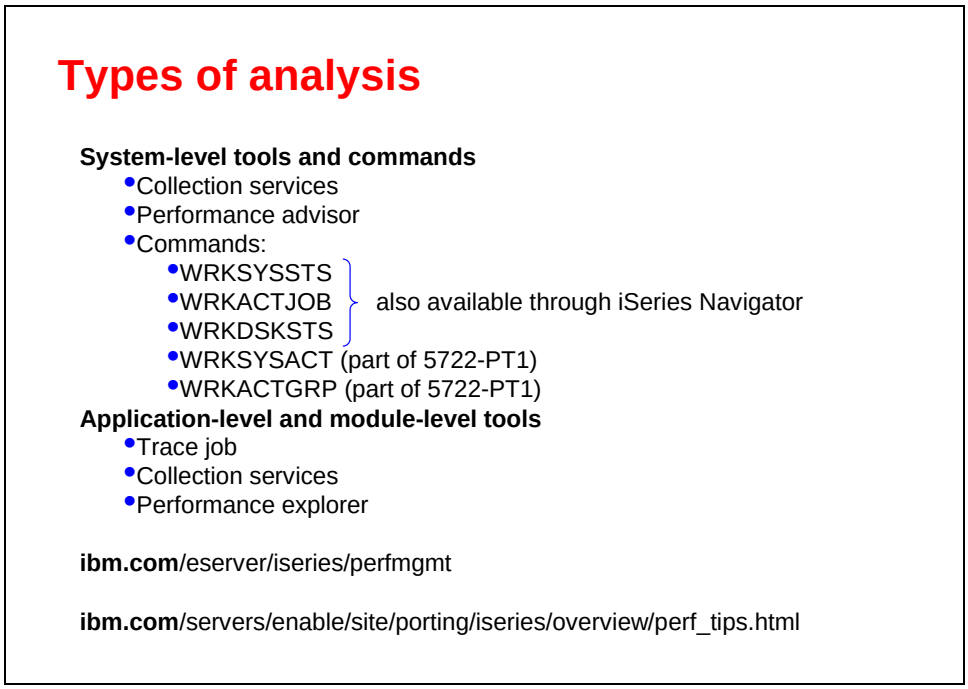
http://publib.boulder.ibm.com/infocenter/iseries/v5r3/topic/rzahx/rzahxapsparent.htm

Performance management tools

The following tools are available to assist in performance management:

- **Collection Services**: This tool gathers performance data at user-defined intervals and stores this information in collection objects on the system. Other tools, discussed below, rely on these collection objects for their data.
- **Performance database files**: Database files can be generated from the collection objects maintained by collection services.
- **Monitors**: These display current information about system performance. Additionally, they can be used to carry out predefined actions when a specific event occurs.
- **Graph history**: This tool provides a graphical display of performance data collected by collection services over a specified period of time.
- **PM iSeries**: Formerly known as PM/400, this tool automates the collection, archival, and analysis of system performance data and returns reports to help manage system resources and capacity.

You can learn more about these tools at the Web site shown on this chart. A link to this site is also provided in the **Resources** section of this course.



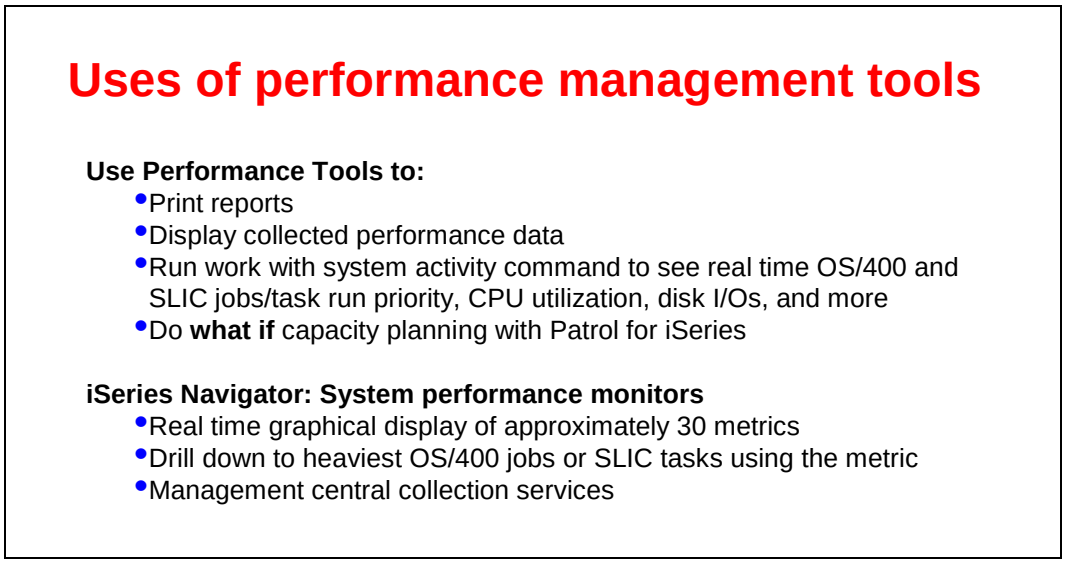
Types of analysis

Various types of performance analysis are provided by the previously discussed tools. These tools are listed here as a review and are grouped by the type of analysis they offer. These categories are as follows:

- **System-level analysis**: This includes surveying hardware utilization, response times, and batch run times. Tools in this category identify such things as bottlenecks, project workload growth, and system upgrade alternatives. System-level performance optimization can be at the expense of individual applications.
- **Application-level analysis**: This includes job traces, profiles, and statistics measured at a transaction or batch job level. Tools in this category can identify modules or subroutines that are consuming the most resources.
- **Module/subroutine-level analysis**: This is the most detailed analysis. Tools in this category identify specific areas in code that are causing errors, reducing resources, or generally not allowing the best performance. These areas can then be redesigned or recoded to improve efficiency.

Both system-level and application-level data can be collected by collection services. The base operating system (i5/OS) contains the commands and programs necessary to collect data. To analyze the collected data, the Performance Tools for iSeries licensed program must be installed.

You can learn more about these tools at the Web site shown on this chart. A link to these site is also provided in the **Resources** section of this course.

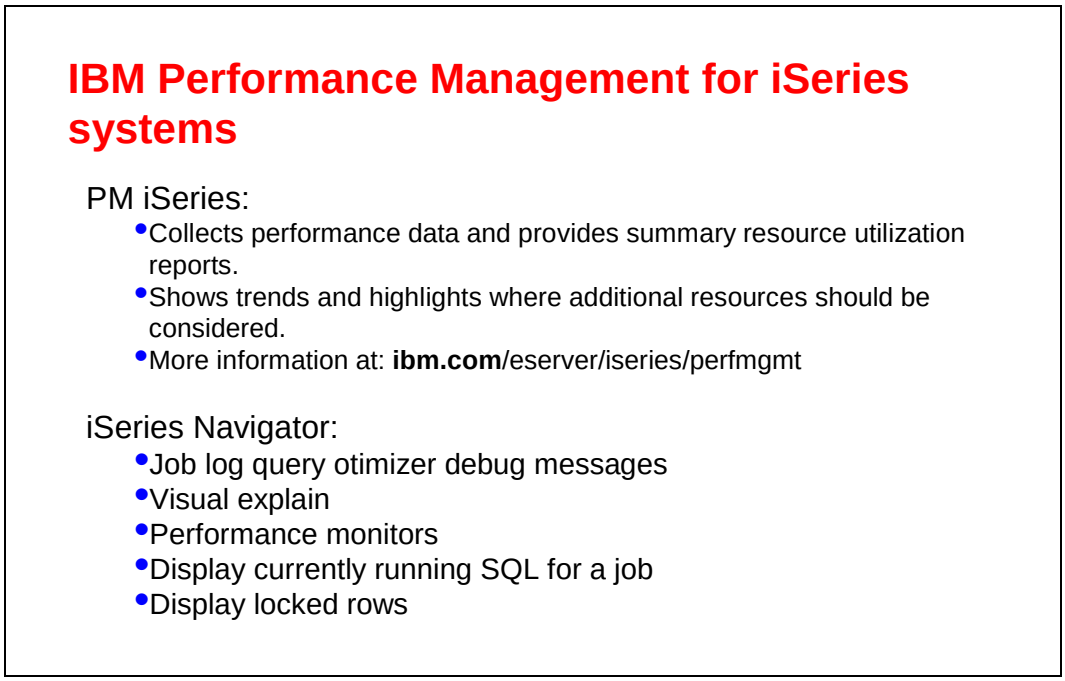


Uses of performance management tools

Performance Tools for iSeries, 5722-PT1, is a licensed program required to do the back-end analysis of data collected with the Performance Explorer (PEX). The licensed program is not required to collect the data, it is only for analysis. Therefore, information that is collected can be restored onto a different iSeries machine where the licensed program is installed.

iSeries Navigator provides interfaces to view system monitor performance data, either in real time or as Management Central Collection Services data. Collection Services allows you to gather data with little or no observable impact on system performance.

You can use iSeries Navigator to configure Collection Services to collect the needed data as frequently as you want. Once Collection Services has been configured and initiated, performance data is continuously collected. When you need to work with the information that has been gathered, it can be copied to a set of performance database files.



IBM Performance Management for iSeries systems

In i5/OS V5R3, the Performance Management/400 name was changed to IBM Performance Management for eServer iSeries (abbreviated as PM iSeries).

PM iSeries is a dynamic tool shipped with the i5/OS operating system that automates many of the functions associated with capacity planning and performance analysis. When activated, the i5/OS Performance Monitor automatically collects system utilization information. This information can include CPU utilization, disk capacity, response time, and throughput, as well as application and user information related to usage statistics.

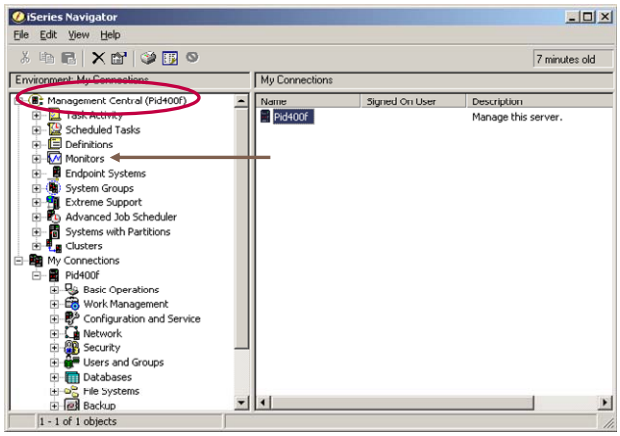
The data collected can be transmitted to IBM and analyzed (for a fee). You will then receive capacity planning and performance analysis reports and graphs that demonstrate your current operating system efficiencies. Based on the trends, the reports will let you know when you need to consider rectifying an approaching problem.

Alternately, iSeries Navigator provides a visual means to analyze database performance data. This includes processes such as job log query optimizer, debug messages, visual explanations, summaries, detailed performance monitors, display of current running SQL for a specific job, and display of locked rows.

Some third-party tools, such as those available from Centerfield Technology, provide similar, but richer, performance analysis functions. For more information on Centerfield Technology, refer to the link provided in the **Resources** section of this course.

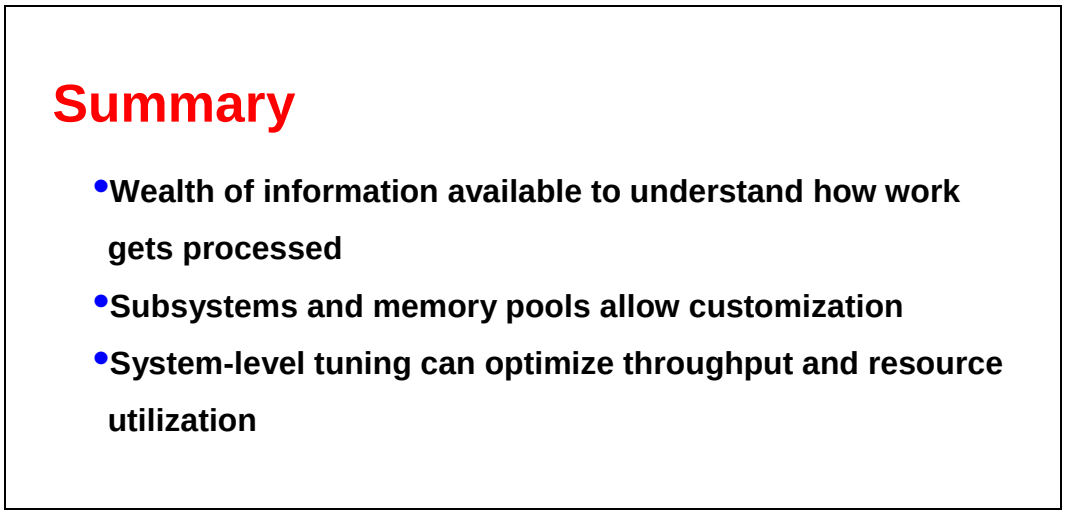
Management Central

An optional component of iSeries Navigator



Management Central

Management Central provides a set of functions that enable i5/OS administrators to manage multiple endpoint systems connected across a TCP/IP network, all from one central iSeries system. Management Central affords a wide range of functions, including monitors to view real-time performance data, and Collection Services to manage performance monitoring and specify the performance data to be collected.



Summary

The iSeries system offers a wealth of information to understand how work is processed within the i5/OS operating system. The system architecture supports constructs, such as subsystems and memory pools, that permit you to customize how work is processed. Based on operational characteristics of the workload, system resources can be adjusted for maximum efficiency. In addition, applications can be optimized for the fastest performance. If jobs fail or performance is suboptimal, the i5/OS operating system offers diagnostic messages and tools.

Resources

The following Web sites offer additional information in regards to the topics discussed in this course:

iSeries Information Center performance topics:

- Managing performance on an iSeries system:
  publib.boulder.ibm.com/infocenter/iseries/v5r3/topic/rzahx/rzahx1.htm
- Performance issues:
  publib.boulder.ibm.com/infocenter/iseries/v5r3/topic/rzaks/rzakspoorperformance.htm
- Work management issues:
  publib.boulder.ibm.com/infocenter/iseries/v5r3/topic/rzaks/rzaksdailyworkmanage.htm
  - As well as other related links on topics such as troubleshooting Work Management, manuals, and experience reports.

- Performance management for IBM iSeries
  ibm.com/eserver/iseries/perfmgmt

- Performance tips
  ibm.com/servers/enable/site/porting/iseries/overview/perf_tips.html

- Centerfield Technology
  www.centerfieldtechnology.com

Trademarks and special notices