

Table of contents

About the author:	2
Introduction	3
Agenda	4
AS/400 models: The legacy.....	5
IBM System family	6
The family of IBM Systems.....	7
System i processor technology.....	8
System i i520, i 550 and i580 performance	9
System i family: Operating-system flexibility	10
iSCSI storage: System x and IBM BladeCenter	11
System i operating-system architecture	12
i5/OS: The primary System i operating system	13
i5/OS innovation: Single-level storage.....	14
i5/OS technology independence.....	15
System i longevity: Technology independence	16
i5/OS design: An integrated system	17
i5/OS application models.....	18
Checking installed software	19
System values	20
i5/OS work	21
User interfaces.....	22
iSeries Access for Windows	23
iSeries Navigator sample: Processor	24
iSeries Access for Web	25
Text-mode interface (terminal emulation).....	26
UNIX style shell interfaces.....	27
i5/OS information management.....	28
Automatic optimized storage	29
i5/OS storage: Integrated file system	30
Key file systems.....	31
PC and UNIX style file systems.....	32
Optical and remote file systems	33
i5/OS library and object space: /QSYS.LIB	34
DB2 UDB for i5/OS	35
DB2 file objects or stream files	36
IFS (root) view from Windows client	37
File-system access: Clients	38
Configure i5/OS file server (iSeries NetServer).....	39
i5/OS built-in security	40
Creating a user	41
Setting object permissions.....	42
Summary	43
Resources.....	44
Trademarks and special notices.....	45



About the author:

Bob Bittner
IBM System i Solution Enablement Specialist

Bob Bittner has been helping software developers become familiar with IBM System i platform and the IBM i5/OS operating environment for more than a decade. He is part of the IBM PartnerWorld group, providing education and technical assistance to software vendors who are migrating applications to the IBM System i platform.

Introduction to the IBM System i family



Introduction

This presentation is an introduction to the technology of IBM® System i™ family. You will learn the important technical elements of the System i platform and the many leading-edge technologies that have been embraced by this family of systems. This new knowledge will provide you with the foundation necessary to successfully begin working with this platform.

Agenda

- Background
- Architecture
- User interfaces
- Information management
- Resources



Agenda

First, this course provides a quick background (from a marketing perspective) of the System i family.

Next, the System i architecture is examined, both software and hardware.

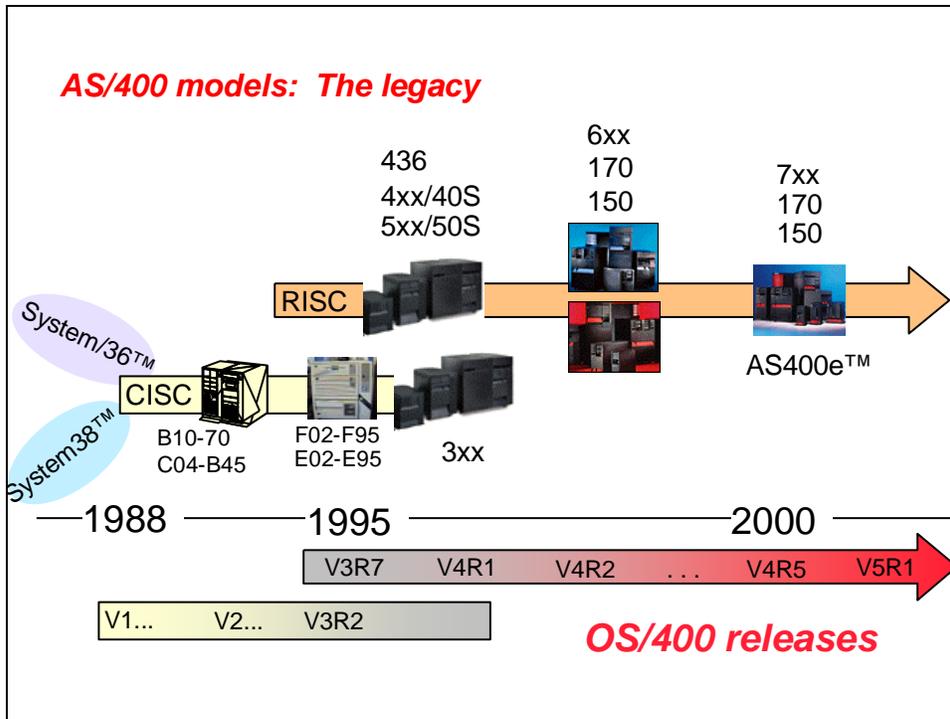
Lastly, you will learn how workloads are organized on this platform.

One of the most basic considerations for any systems environment is its user interfaces. In this course, you will learn how to interact with the System i platform.

This course will show you how data is stored on the System i family, including simple binary data, as well as data that is stored in the built-in relational database.

Next, you will see how work is organized on the System i platform.

Note: At the end of this course, you will find a listing of sources for additional information about the System i family.



AS/400 models: The legacy

In 1988, IBM introduced a revolutionary new business computer. Building on architectural advances in computer science, taking into consideration the UNIX® experience, advances in desktop computing and, most importantly, the vast experience of IBM in helping business users solve problems and enhance their productivity with computing, IBM created the AS/400® server line.

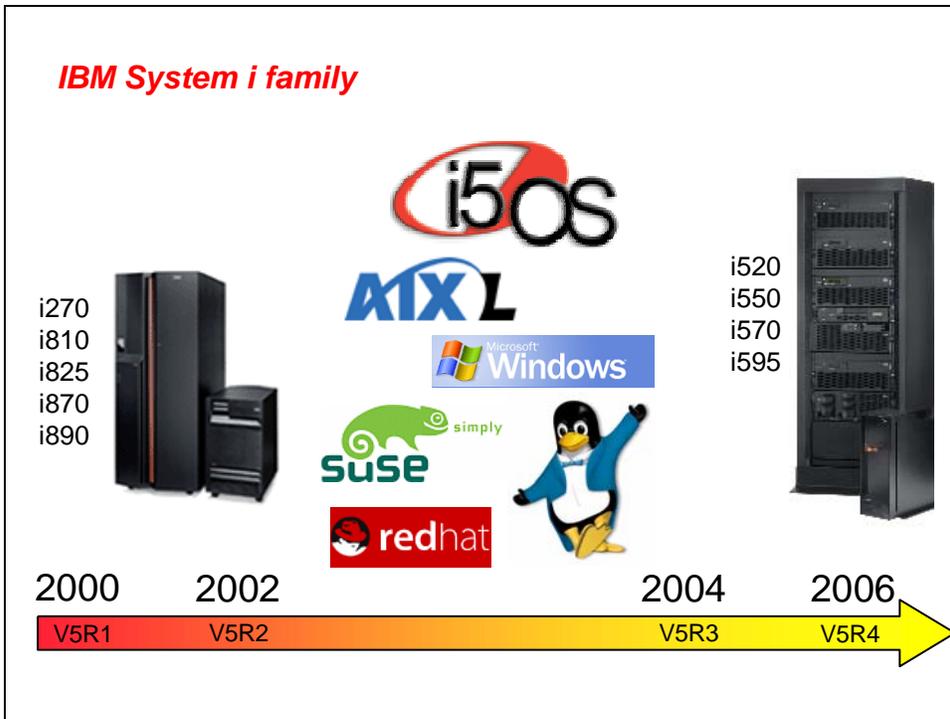
The AS/400 design principles were straightforward:

- It must be easy to use.
- It must be extremely reliable.
- It must integrate all the components that the enterprise needs.
- It must protect the enterprise's investment.
- It must be secure enough to trust with any kind of information.

With a steadily growing and extremely loyal user base, AS/400 systems became the world's most popular back-office business computers. By 1995, more than half a million installations worldwide dwarfed the next-most-popular, multiuser computing platform (many varieties of UNIX).

But the 1990s were tough on AS/400. Some shortsighted analysts predicted the death of the AS/400 platform. In a sense, they were right. What they understood AS/400 to be — a proprietary legacy system for order entry and batch processing — was not good enough.

That is why IBM has invested so much to revolutionize the AS/400 platform. In 1995, the core hardware technology was replaced in its entirety with the most modern reduced instruction set computer (RISC) processors and I/O technology. The database engine received innovations that still are unmatched. The software was revamped with modern services, the ability to run Java™ and UNIX applications, and a super-easy-to-use graphical interface — all the while holding true to the value proposition that it always has to work right, it has to be easy to use and it has to be affordable.



IBM System family

These systems truly bring a new level of systems integration. At the core of this family of systems remains the capabilities of its AS/400 predecessor, easy-to-use systems that integrate a world-class database (IBM DB2® for i5/OS), Java, high availability, and other important components. Adding to that foundation is the ability to run Linux®, Microsoft® Windows® and IBM AIX® applications. The System i platform also adds on-demand features such as processor Capacity On Demand and on-demand dynamic logical partitioning (LPAR).

Combining all of these environments might seem a bit much to comprehend at first. Therefore, in the following panels, you will see how it is actually accomplished on this platform.

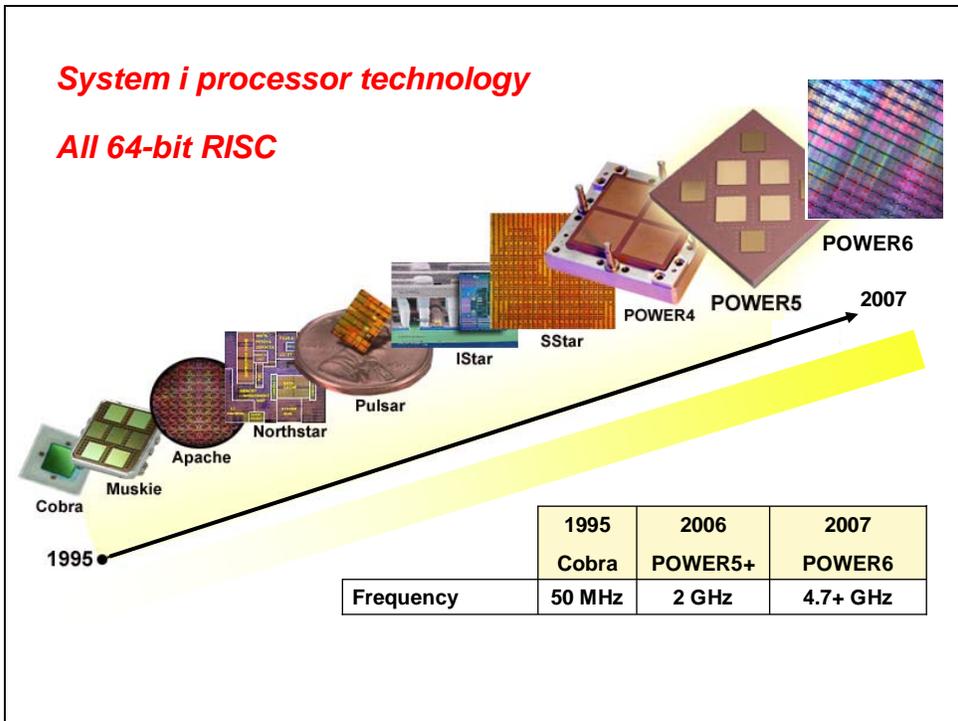
The family of IBM Systems

	System i <ul style="list-style-type: none">High-performance <i>integrated</i> systems built for business		System p <ul style="list-style-type: none">High-performance advanced <i>UNIX</i> servers
	System x <ul style="list-style-type: none"><i>Intel</i> processor-based servers with mainframe-inspired reliability technologies		System z <ul style="list-style-type: none"><i>Most reliable</i>, mission-critical database and transaction servers

The family of IBM Systems

In 2000, IBM brought together the best attributes of a diverse line of systems to create the IBM Systems family, which comprises four individual system platforms:

- The **IBM System z™** platform replaces mainframes that run the IBM OS/390® operating system, among other operating systems. The **z** stands for *zero downtime*. When absolute availability is the requirement, System z models provide some of the industry's best high-availability capabilities and run the IBM z/OS® operating system.
- The **IBM System x™** line of systems is designed for the entry-to-midsized enterprise environment where the IBM Enterprise X-Architecture™ and Intel® microprocessors play a key role. If a Microsoft or Linux operating system is adequate for you, System x models are an excellent choice.
- The **IBM System p™** platform provides UNIX® servers. These machines run the IBM AIX operating system, one of the leading UNIX operating systems, and Linux. System p models are ideal if UNIX is your requirement.
- The **IBM System i** platform is known for its high level of system integration. When building a business computer, many varied components are required. The family of System i models have these components built-in. These systems run applications from their IBM OS/400® heritage, an operating system that was renamed *IBM i5/OS* with Version 5, Release 3. These systems also support LPARs that run applications founded in the AIX and Microsoft Windows operating environments.



System i processor technology

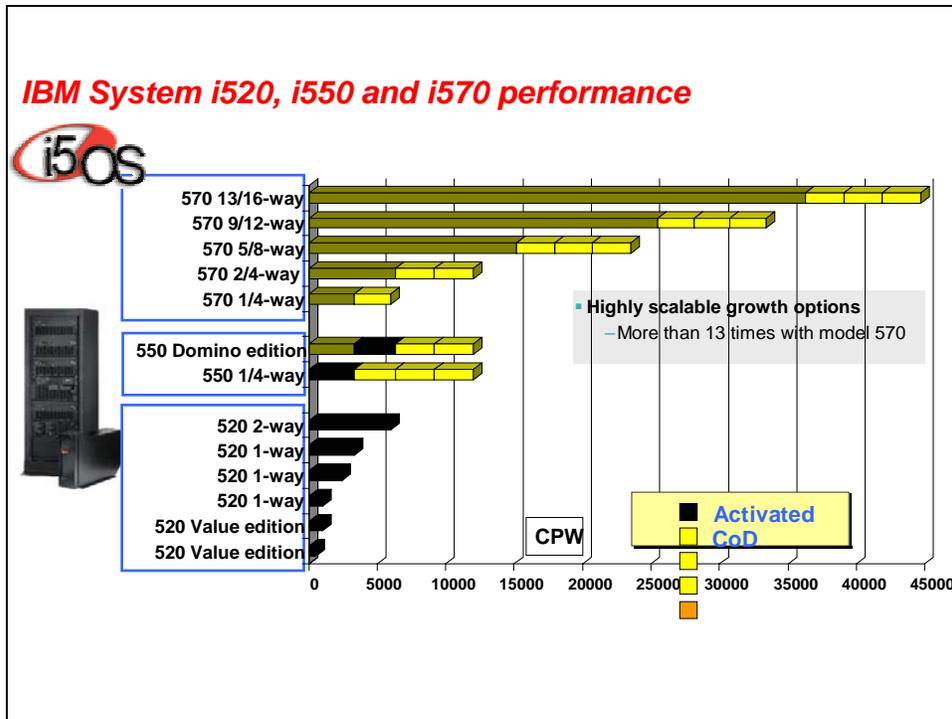
The System i models use the IBM POWER™ microarchitecture. By a significant margin, this is the most advanced mainstream, 64-bit computing architecture available today. Although other vendors' 64-bit processors are largely prototypes, demonstrations and hype, IBM has been delivering solid 64-bit benefits to the marketplace since 1995.

IBM POWER5™ technology is much more than just an improvement in processor design. It is a complete architectural change, creating a much more efficient, superscalar-processor complex. POWER5 processors have two load and store units, two arithmetic units and one branch unit. The processor complex is built in such a way that it can most efficiently run multiple-instruction streams concurrently. With simultaneous multithreading (SMT), it is possible to issue instructions to two different threads per single processor cycle.

In general, multithreading allows a single processor to process multiple threads in a different fashion, thus minimizing the processor-wait or -idle time as compared to a single processor without this capability.

The POWER5 concept is also a step further into autonomic computing. A number of enhanced reliability and availability enhancements are implemented. Along with increased redundant components, it incorporates new technologically high standards, such as special ways to reduce junction temperatures to reach a very high level of availability. A full system-design approach is required to maintain balanced utilization of hardware resources as well as high availability of the System i models.

The IBM POWER6™ chip is also one of the first widely sold that uses the IBM 65-nanometer manufacturing process. It offers a host of improvements, including very high (near 5 GHz) frequencies, additional built-in virtualization features, 50 percent power savings compared to the POWER5 processor, and much faster internal bandwidth.

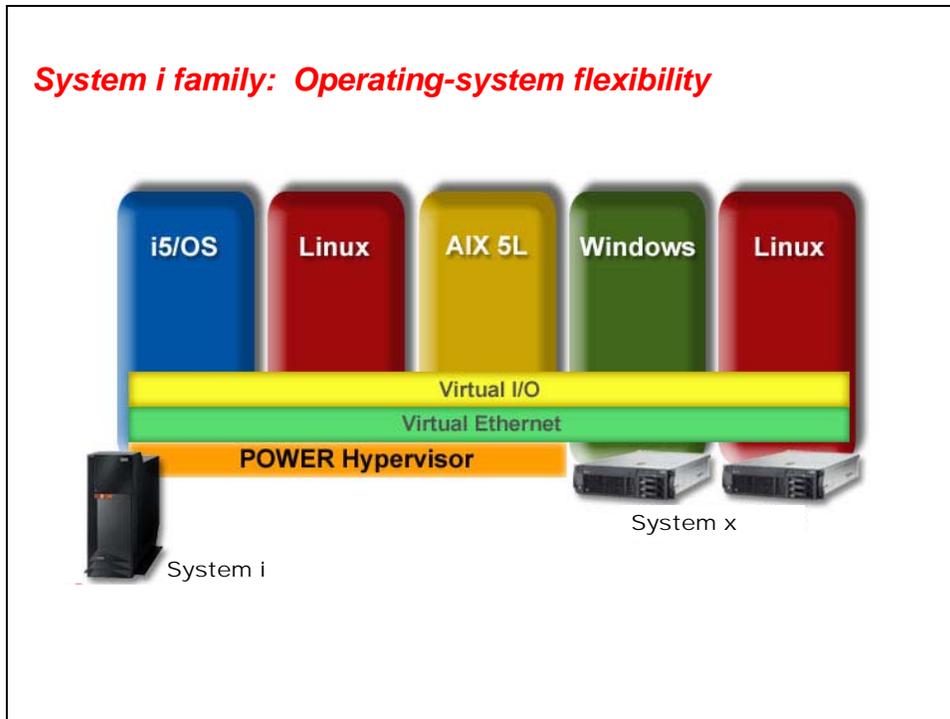


System i i520, i 550 and i580 performance

This slide shows a portion of the IBM System i product line. The chart compares the relative performance of System i models, measured in units of commercial-processing workload (CPW).

For an extensive reference of all System i models and features related to performance, see the System i Performance Capabilities Reference at ibm.com/eserver/series/perfmgmt.

Note: Performance in enterprise environments might vary. The CPW value is measured on maximum configurations.



System i family: Operating-system flexibility

The System i family can simultaneously run multiple operating systems. i5/OS is the primary operating system, which is always present. Additional operating systems run in dynamic LPARs and Integrated xSeries Servers (IXS).

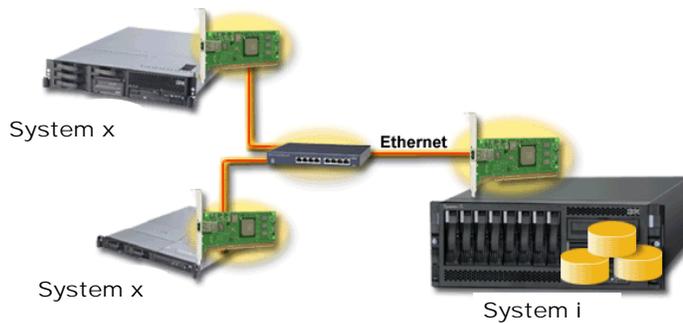
i5/OS, AIX and Linux use the POWER5 processors that are at the heart of the System i platform. A transparent manager called IBM POWER Hypervisor™ dynamically instructs the hardware to allocate processor, memory and I/O resources to the dynamic LPARs (which are running simultaneously). **Very** high-speed and flexible networking is provided between DLPARs.

When Linux and AIX operating systems are added to a System i model, they run their respective kernels on their assigned partition. Thus, the respective environments that users and applications see is exactly what is expected of a traditional Linux or AIX system.

Windows and Linux x86 operating systems also run on System i models using auxiliary Intel processors. These plug-in processor boards have dedicated RAM and PCI slots. However, they share disk storage with the rest of the system. The Intel processor-based System x servers are the technology source for these blade-type servers. Thus, the name of this add-in is the Integrated xSeries Server (IXS).

Consolidating multiple systems onto a single, secure footprint provides numerous cost and server-management benefits, including the integration of users and backup tasks. There are also unique enhancements to the recoverability and system-replication functions.

iSCSI Storage: System x and BladeCenter



- **iSCSI (Internet SCSI)**
 - International standard for TCP/IP-connected storage
- **iSCSI Host Bus Adapters exploit...**
 - System i virtual storage
 - System i tape and optical devices

iSCSI storage: System x and IBM BladeCenter

The System i platform is built on the IBM X-Architecture® technology for Intel processors. One of these servers, the IBM IXS is a plug-in card that integrates into System i models.

Depending on the System i model, you can install several IXS cards, each of which uses storage provided by the hosting i5/OS file system (IFS). As such, the disk drives seen by the Windows applications that run on the IXS cards are, in reality, files created on the i5/OS single-level storage. (Single-level storage is explained in a moment.) The use of IXS cards provides a way to easily scale Windows environments beyond the usual limitations of stand-alone PC servers. As needed, more storage can be allocated to these servers without rebooting either the Windows or i5/OS operating systems.

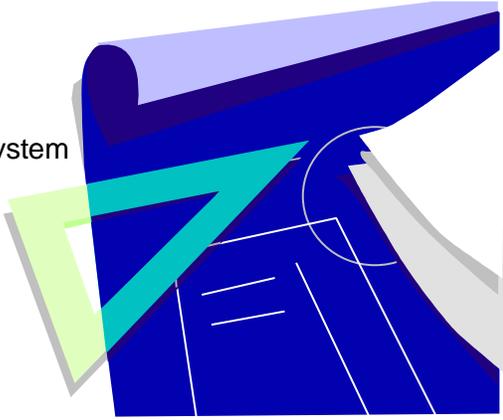
Several aspects of managing a Windows server improve when using IXS. You can automate your backup procedures as a normal part of backing up i5/OS to offline media. Additionally, i5/OS monitors the status of Windows and can reboot Windows as conditions warrant. It is possible to replicate servers. Additionally, you can swap whole servers in just a few minutes.

One consultant referred to the IXS as "adult supervision for Windows."

**System i
operating-system
architecture**



Single-level storage
Technology independence
Integrated, complete operating system
Application models
Jobs
System values (policy settings)



System i operating-system architecture

The next few panels examine the hardware and software architecture for the System i platform. Again, these will be talking about integration.

i5/OS: The primary System i operating system



Version:
V5R4M0

"V5R4"



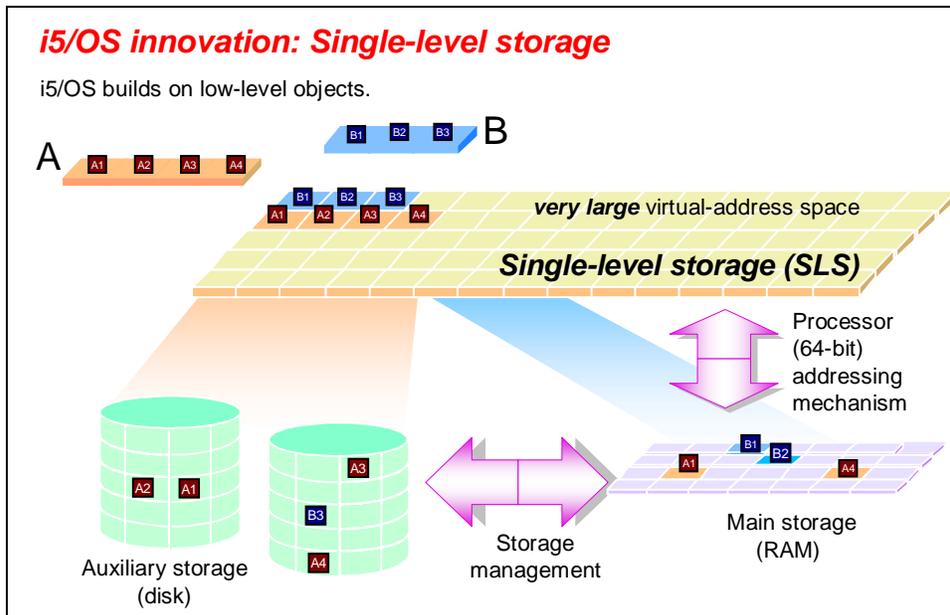
i5/OS: The primary System i operating system

The primary operating system that runs on every System i model is i5/OS (formerly named OS/400). Among other functions, this operating system provides the following:

- The integrated version of IBM DB2 (the IBM database engine)
- The fundamentally integrated Java virtual machine (JVM)
- The primary Internet services needed for e-business
- The transparent-management layer, POWER Hypervisor, which allocates resources across the entire system, including all LPARs

i5/OS releases are shipped on a regular basis. The current shipping release of i5/OS is Version 5 Release 4 or V5R4.

Note: As is the nature of software evolution, some capabilities described in the rest of this presentation are only available at i5/OS V5R4.



i5/OS innovation: Single-level storage

Here is a glance at how storage works on i5/OS. In this context, *storage* refers to both memory and disk. This fundamentally different architecture enables scalability, flexibility and security advantages that no other computing platform has. A traditional UNIX file system, as well as the advanced i5/OS object-based structure, are both built on top of the very large common address space known as single-level storage. Single-level storage is the foundation on which all i5/OS objects exist.

Single-level storage allows the system to access all storage in a uniform manner, within a single, very large virtual-address space. The operating system and applications do not consider an object to be in memory or on disk. All objects are simply in the single-level storage. The storage-management kernel component, which resides in the system-licensed internal code (SLIC), manages everything that is physically in memory and physically on disk.

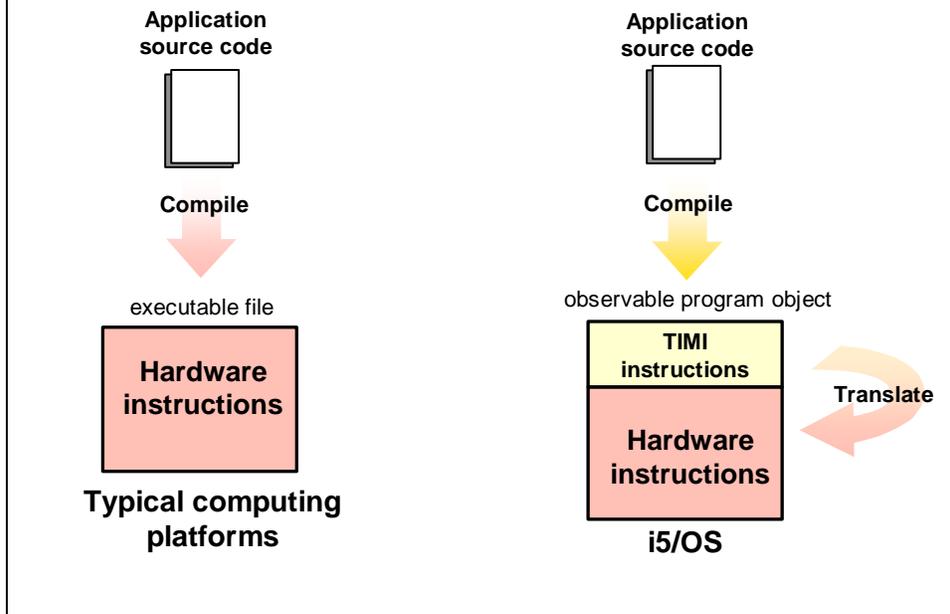
In this example, two objects are created, A and B. Object A is designated as a permanent object. This instructs storage management to ensure that the object is maintained on some nonvolatile disk storage. Object B is designated as temporary. Thus, the pages of this object are only sent to auxiliary storage to make the best use of the real physical memory.

This makes system tuning easier because there is no concept of worrying about disk-cache size, paging size or journaling size. This is also easy for developers, because programs do not need to manage memory for optimal performance. If an object is used, it is kept in memory (main store). If not, it is paged out. The distinction is that there is no wasted space that is otherwise required when copying from one to the other — as is the case on systems that distinguish, above the operating system, a difference between disk and memory.

The architecture affords different addressing models. Traditionally, pointers within code are 128 bits and directly address the entire single-level storage. Furthermore, a program can run within large private-memory segments using either 64- or 32-bit pointers. These large private segments are referred to as *teraspaces* because they are up to one terabyte in size.

The system automatically balances (stripes) data across disk units. If you need more auxiliary storage, disk units are added and the system rebalances the data across the pool of disk drives.

i5/OS technology independence

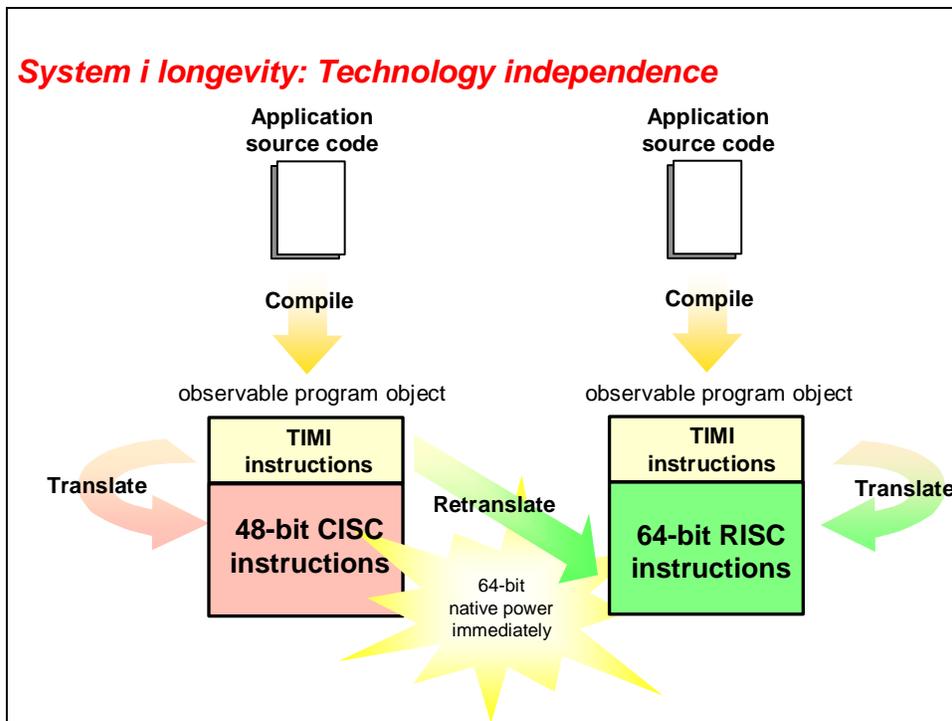


i5/OS technology independence

A traditional compiler generates executables for a specific type of processor — as shown on the left side of this diagram.

The i5/OS compilers generate executables for a high-level machine interface — as shown on the right side of this diagram. The technology-independent machine interface (TIMI) translates (only one time) the intermediate representation to real hardware instructions to run on the processor.

The benefit is investment protection, both for enterprises and application providers. Applications are not tied to a specific hardware architecture.

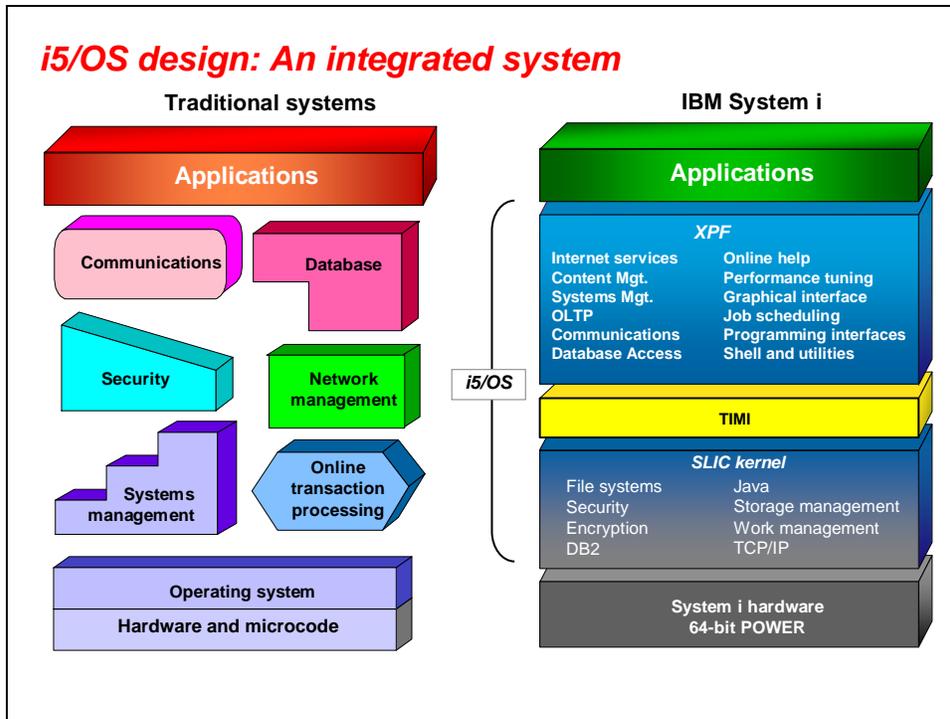


System i longevity: Technology independence

The coupling of the object-based architecture, the TIMI and single-level storage produces an extraordinary level of upward and backward compatibility for applications running on i5/OS. In the early 1990s, AS/400 applications used 48-bit processing on CISC hardware. In 1995, IBM introduced 64-bit RISC AS/400 systems. Because IBM rewrote the SLIC kernel (below the TIMI) and the optimizing translator (OX), applications written years ago are instantly fully operational on, and optimized for, POWER processor-based 64-bit systems — without intervention from the original developer of the applications.

With retranslation, the SLIC kernel (below the TIMI) adapts applications so that they are optimized for the latest hardware. Regardless of when the application was built, the same code is optimized for whatever hardware implements the system.

It is often a surprise to find that the largest number of 64-bit applications in the world are written in RPG and run on i5/OS.



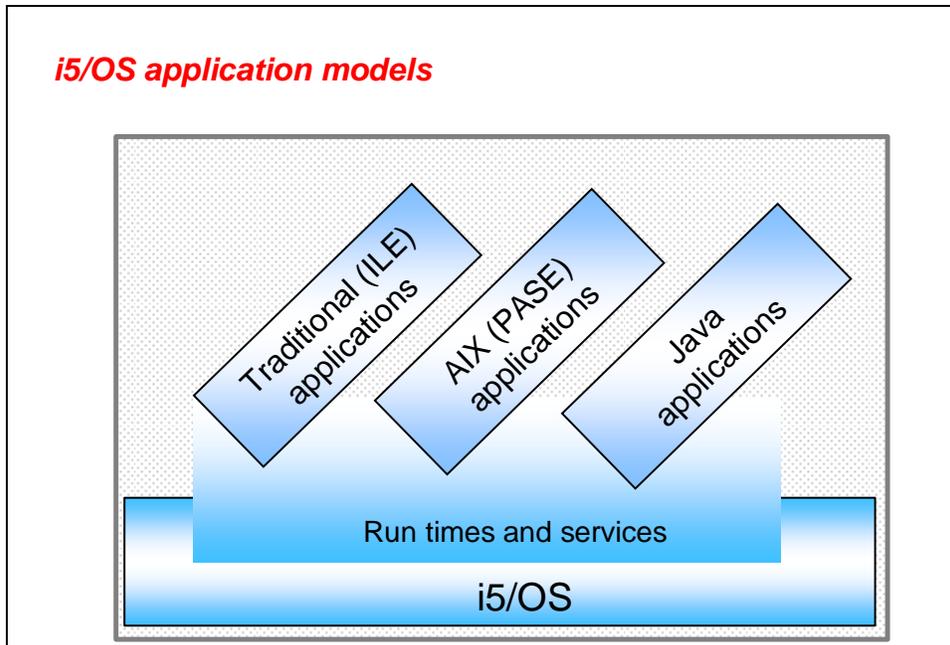
i5/OS design: An integrated system

i5/OS, the primary operating system that runs on System i, integrates many of the components required on a modern business-computing platform.

These services and features can be easily divided into two categories: *high-level services* and *core functions*. The i5/OS high-level services are often optional, and they are typically the things with which a user would interact. These high-level functions are above the TIMI. Below the TIMI, the SLIC kernel provides the core functions that serve as the foundation on which the system operates.

The advantages to this approach are many. First, users do not need to purchase additional componentry, such as a database engine. Second, all of these features were developed and tested together. Therefore, there is no worrying about version conflicts between database and operating system or systems-management software and security layer. The expensive integration services required on other platforms are unnecessary with System i.

Notice also that the software stack isolates applications and much of the operating system from the underlying hardware. In this way, investments in applications are protected — even as processor technology (and other technologies) advance. Several times in the history of the System i platform, IBM has changed the system's basic processor architecture, without inconveniencing users.



i5/OS application models

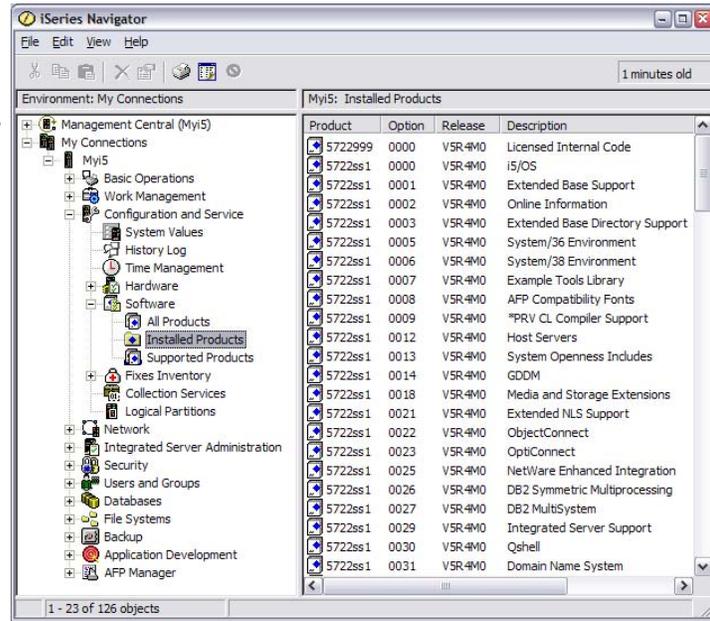
The rest of this course focuses on i5/OS, the primary operating system for the System i platform. Within a single i5/OS partition, there are three models of applications.

- Traditional System i applications run under a model known as *IBM Integrated Language Environment® (ILE)*. This model allows C/C++, RPG, CL and COBOL applications to easily call functions from, and share data with, each other through interaction that occurs at the TIMI level. Above the TIMI, all programs compile to an intermediate form that is automatically translated into actual hardware instructions by the SLIC kernel prior to execution. ILE applications are not interpreted.
- Java applications can use the *JVM* that is built into the i5/OS kernel. The i5/OS JVM takes advantage of the object-oriented i5/OS architecture to enable a uniquely scalable and high-performance Java machine. Java applications often run with IBM WebSphere® Application Server, although they can also run with IBM HTTP Server, which is the Apache Tomcat server.
- The newest addition to the lineup of application models is the *i5/OS Portable Application Solutions Environment (PASE)*. Applications using this model comply with the AIX binary interface. Executables from AIX applications can run on i5/OS because both operating systems use the same processors. PASE is the enablement method for running the AIX executables; PASE includes the AIX runtime libraries. Applications using this model see a view of the server that is similar to UNIX in its function, yet also similar to i5/OS in its ease of use.

The i5/OS service that is provided in all three of these application models is standardized and available by default. Applications run directly on the processors. Further, each of these application models can interact directly with each other, and, because they all rely on the same kernel, they can also interact indirectly, through the integrated file system (IFS) or sockets (with user IDs and passwords).

Checking installed software

- **Product ID**
 - 7 characters
 - 5722SS1
 - 5722JV1
 - 5722DG1
 - 5733W61
- **Options**
 - For each ID
 - i5/OS components
 - JDK levels
 - WebSphere Application Server subversions



Checking installed software

It is easy to determine which IBM software products are installed and available on a particular System i model. From iSeries Navigator (discussed later), navigate as follows:

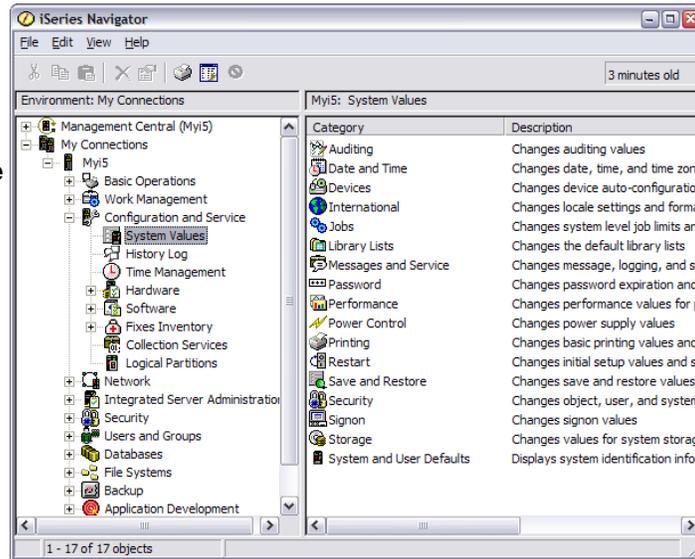
Management Central → My Connections → Configuration and Service → Software → Installed Products

Notice that all IBM product IDs are seven characters long.

Also, you will be able to notice the installed options for each software component.

System values

- Control overall operation
- Provide information
- Grouped by type
- Named
- Examples
 - Date
 - Audit level
 - Library list
 - Message logging
 - Password length
 - Storage limit
 - Security level



System values

System values are settings that provide information for the system's use. Specifically, system values govern the behavior of one i5/OS LPAR.

You can view and change system values from iSeries Navigator, as this screen capture illustrates.

Alternatively, you can use the i5/OS Work with System Values (WRKSYSVAL) command in the text interface to change the system values. When using the text interface, you see the coded names of these values. For example, the QLMTSECOFR value limits the access for privileged users.

i5/OS work



- **Job**
 - Process

- **Threads**
 - Stream of work inside a job
 - All jobs have at least one

- **Spooled files**
 - Job output

- **Job status**
 - Job queue
 - Running, active
 - Wait

The screenshot shows the iSeries Navigator interface. The left pane displays a tree view of system components, with 'My5' expanded to show 'Jobs'. The right pane displays a table of active jobs for user Qsecofr.

Job Name	User	Status	Type
Qgldpuba	Qdirsv	Running	Autostart - Server
Qgldpube	Qdirsv	Running	Autostart - Server
Qibmhelp	Qibmhelp	Running	Batch - Server
Qjvacmdsrv	Qibmhelp	Running	Batch immediate
Qmsf	Qmsf	Running	Batch
Qtotntp	Qntp	Running	Batch - Server
Qqqttemp1	Qpgmr	Running	Batch
Qqqttemp1	Qpgmr	Running	Batch
Qqqttemp2	Qpgmr	Running	Batch
Qqqttemp2	Qpgmr	Running	Batch
Qserver	Qpgmr	Running	Autostart
Qsyscd	Qpgmr	Running	Batch
Qzssserver	Qpgmr	Running	Batch - Server
Srvmon0000	Qsecofr	Running	Batch - Server
Srvmon0001	Qsecofr	Running	Batch - Server
Srvmon0002	Qsecofr	Running	Batch - Server
Srvmon0003	Qsecofr	Running	Batch - Server
Srvmon0004	Qsecofr	Running	Batch - Server
Srvmon0005	Qsecofr	Running	Batch - Server
Srvmon0006	Qsecofr	Running	Batch - Server

i5/OS work

Each job on i5/OS is identified by a job ID and consists of three parts:

- A name
- The user who started the job
- A job number

This screen capture shows an iSeries Navigator listing of the jobs for a given user. You can see the three items in the job ID for each job.

In addition to the Job ID, which all jobs have, a job might be assigned a process ID. The process ID (PID) is simply a number that is used on UNIX types of APIs to identify a process.

i5/OS user interfaces

Graphical
Browser
Wireless
Text-mode



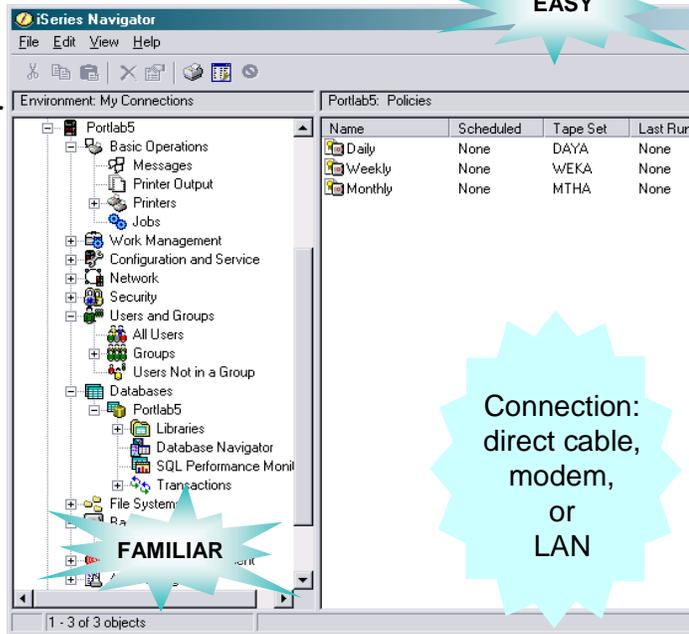
User interfaces

You have a choice of four main user interfaces to interact with the System i operating system, i5/OS:

- The primary interface is the *IBM iSeries Access for Web* GUI. It operates in a client-server mode, where the client runs on a PC, and the server is, obviously, i5/OS.
- IBM iSeries Access for Web can also be run from a browser to interact with i5/OS.
- IBM iSeries Access for Wireless provides PDA and mobile-phone browser-based access to i5/OS.
- The iSeries platform also supports a text-mode user interface.

iSeries Access for Windows

- **Windows client Connectivity**
- **iSeries Navigator**
 - Free
 - Ships with every system
- **Manage and monitor**
 - Users
 - Files
 - Database
 - Networks
 - Jobs
 - Performance
 - Backup
 - Servers
 - Much more

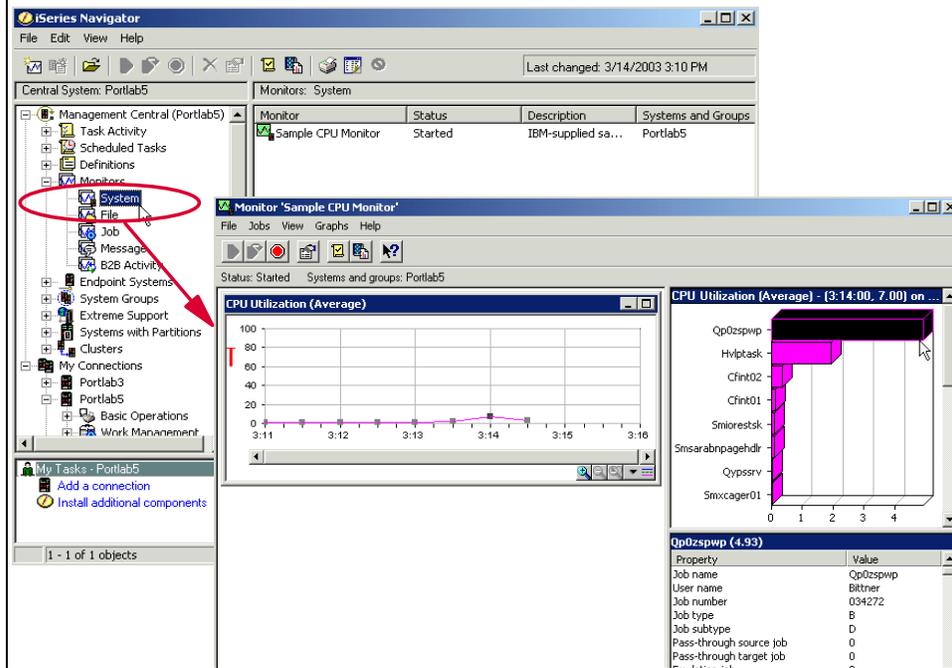


iSeries Access for Windows

The graphical interface to the System i platform occurs through a Windows client, which communicates with several servers running on i5/OS. Most system functions are available using this interface. The client package that is installed on Windows based PCs is delivered, ready to download, on i5/OS and OS/400 releases.

IBM iSeries Access for Windows is the name of the client product. It includes several components, but the main one of interest is IBM iSeries Navigator, often referred to as *iNav*. IBM adds new i5/OS features to this interface on a priority basis (as compared to adding functions to the other interface options). Some other components of iSeries Access for Windows are a data-transfer tool, a 5250 terminal emulator, an ODBC driver and a print viewer.

iSeries Navigator sample: Processor



iSeries Navigator sample: Processor

This screen capture is an example of what iSeries Navigator looks like. Notice the familiar Microsoft Windows Explorer type of navigation, which is also referred to as *tree access*. Here, you can see that the user wants to look at the system to determine what kind of processor utilization is occurring.



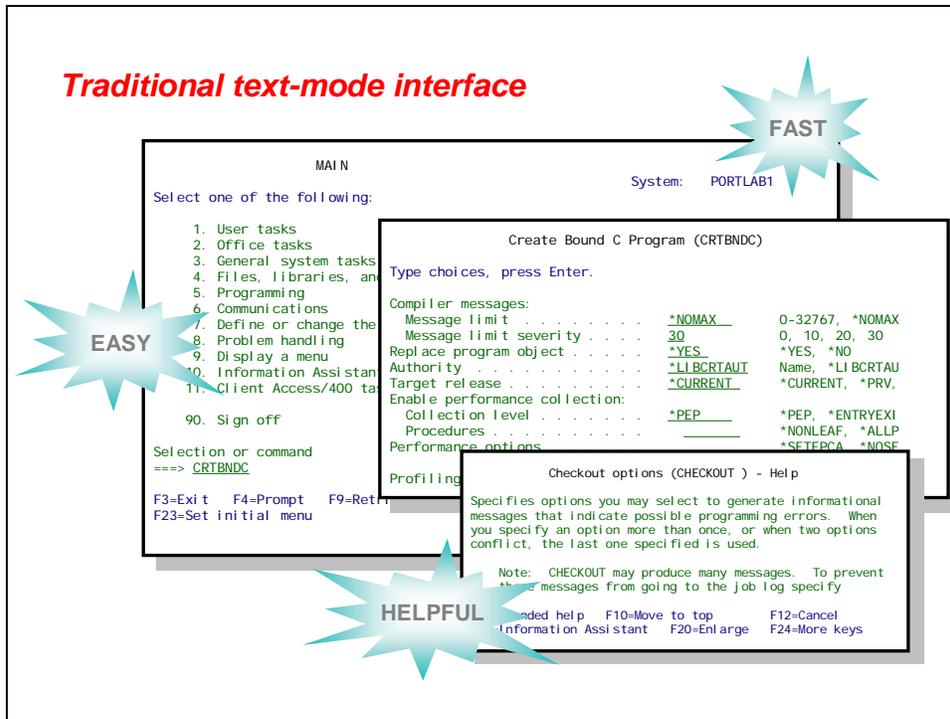
iSeries Access for Web

Another interface to i5/OS is possible through any of three standard Web browsers:

- Mozilla Firefox
- Microsoft Internet Explorer
- Opera (from Opera Software)

Many services and i5/OS functions are available using iSeries Access for Web. This chart lists some of the main categories.

iSeries Access for Web runs within a Java Web application server. The two supported Web application servers are Apache Software Foundation (ASF) Tomcat (IBM HTTP Server) and WebSphere Application Server (Express or Enterprise edition).



Text-mode interface (terminal emulation)

The original interface to the AS/400 system is known as the *text-mode interface*. This interface is also referred to as a *5250 interface* because of the model number for the terminal that was originally connected to these computers. Alternatively, it is called the *green-screen interface*, in deference to its green-on-black style of displaying information.

Today, you can connect to an i5/OS instance through text-mode emulation by using a lightweight Telnet connection. This presents you with a text-based interface. The main text interface occurs through a command-line (CL) language and includes access to nearly all i5/OS commands. As seen here, this interface is more than just a CL. Interactive prompting menus, several kinds of screens and extensive help are quickly available. This offers unmatched consistency that makes the text interface user-friendly and predictable. You only need to understand three function keys:

- **F1 - Help:** This key, which is always available to the user, displays specific help instructions that are related to the current position of the cursor.
- **F4 - Prompt:** When you are not sure what to do, press this function key to be shown step-by-step options.
- **F12 - Cancel:** This key gets you out of the current display.

UNIX style shell interfaces

- Shell scripts
- Familiar commands
- I/O redirection
- Text-mode applications
- Secure shell (OpenSSH) connectivity with 5733-SC1

```

/00openSys/usr/bin/-sh
> pwd
/
$
> ls -ld Q*
drwx---rwx 10 qdoc 0 8192 Jan 01 1970 QDLS
drwxrwsrwx 8 qsys 0 8192 Jan 24 20:29 QFPNWSSTG
d-----r-x 2 qsys 0 2176 Feb 17 22:41 QFileSvr.400
drwxr-sr-x 7 qsys 0 8192 Oct 04 16:39 QIBM
drwxrwsrwx 2 qdftown 0 4096 Jan 01 1970 QOPT
drwxrwsrwx 8 qsys 0 143360 Feb 28 19:34 QOpenSys
drwx---r-x 983 qsys 0 1296768 Mar 04 21:37 QSYS.LIB
drwxrws--- 9 qtcp 0 8192 Oct 04 16:46 QTCPTMM
$

===>
F3=Exit F6=Print F9=Retrieve F11=Truncate/Wrap
F13=Clear F17=Top F18=Bottom F21=CL command entry

xterm
$ cd /
$ ls -ld Q*
drwx---rwx 10 qdoc 0 8192 Dec 31 1969 QDLS
drwxrwsrwx 8 qsys 0 8192 Jan 24 14:29 QFPNWSSTG
d-----r-x 2 qsys 0 2176 May 06 10:12 QFileSvr.400
drwxr-sr-x 7 qsys 0 8192 Oct 04 2002 QIBM
d-----r-x 2 qdftown 0 1152 May 06 10:12 QNTC
d-----r-x 2 qdftown 0 406 May 06 10:12 QNetWare
drwxrwsrwx 2 qdftown 0 4096 Dec 31 1969 QOPT
drwxrwsrwx 9 qsys 0 143360 May 21 08:03 QOpenSys
d---rws--- 2 qsecofr 0 12288 May 05 12:23 QSR
drwx---r-x 964 qsys 0 12361728 Jun 16 15:40 QSYS.LIB
drwxrws--- 9 qtcp 0 8192 Oct 04 2002 QTCPTMM
drwxrwsr-x 4 qsys 0 8192 Oct 04 2002 QUGEN
drwxrwsrwx 2 qsys 0 4096 Oct 04 2002 QVGN
$ vi _

```

UNIX style shell interfaces

In addition to the traditional CL commands and menus, i5/OS includes several UNIX style shells and utilities. These shells supply approximately 200 common systems utilities. The X-Window libraries are also provided, as well as pseudoterminal functions to support the *curses* library so that applications can deliver a UNIX style of text interfaces to System i users. Interactive text-mode programs that use the *curses* library can only run within an X terminal as of OS/400 V5R2 and later releases.

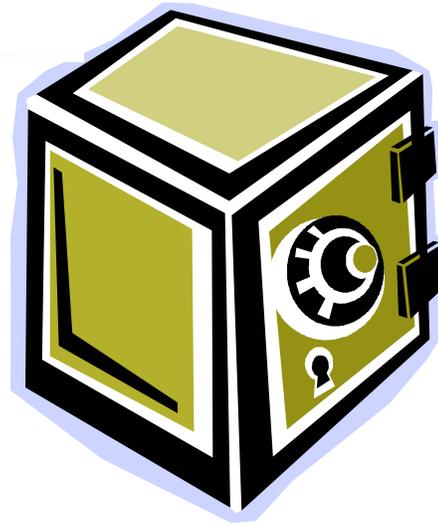
Note that these shell-command interfaces run entirely differently than the traditional CL commands. These shells work as they do on UNIX and Linux, without providing any prompting. From these shells, you can enter CL commands by using the System utility.

Most situations that provide a command interface to i5/OS use CL commands, not shell commands (for example, remote commands that are submitted from FTP, REXEC and iSeries Navigator).

To add ssh connectivity to i5/OS, load the the following product:
IBM Portable Utilities for i5/OS (5733-SC1). Install option 1 – OpenSSH (OpenSSL, zlib)

i5/OS information management

Integrated file system
Stream files
Objects
Built-in DB2 database
Access and management
Security



i5/OS information management

i5/OS provides the ability to store and manage any kind of information — relational database data, free-form text, images, documents, applications, optical libraries, network storage and more. Each kind of data is directed into a file system that is ideal to that data.

Automatic optimized storage

- Automatic management by i5/OS
- Ready-to-use
- Resides on single-level storage (SLS)
- Automatic cache management
- Striping / mirroring / RAID
- Load balancing



Automatic optimized storage

The i5/OS IFS is made up of a few unique file systems. Each of them has a set of logical structures and rules for interacting with the information stored therein. Some file systems connect to remote servers, or to another System i server, or even to other types of remote systems. However, most file systems on i5/OS contain data that is local to that system. The IFS provides an integrated structure over all the information that is stored in the system (for example, stream files, database files, directories, folders and network storage).

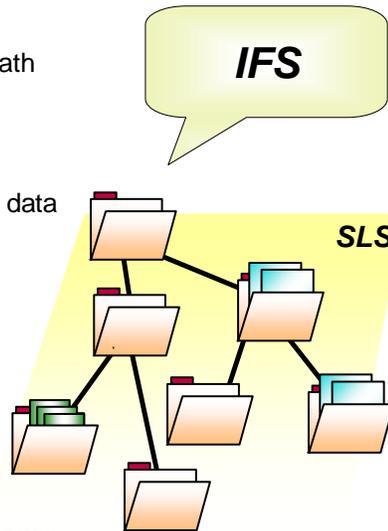
Each file system is a major subtree that is mounted onto the root file system, at the top level of the directory structure. Some basic characteristics of the IFS include the following characteristics:

1. Directories are separated with forward slashes, unlike the backward slashes used on DOS and Windows systems.
2. Applications can use the same APIs to work with data in any file system.
3. A consistent interface for securing local file systems is provided.

Perhaps most importantly, managing file systems and data on individual disks is automatic. The i5/OS operating system handles that task. A System i administrator does not need to be concerned about which file system resides on which drive. Also, occasional checking or defragmenting of file systems is either automatic or unnecessary, depending on your perspective. The system's architecture ensures integrity.

i5/OS storage: Integrated file system

- Hierarchical directory structure
 - Objects are referenced by directory path
- Includes all storage
 - Byte-stream files
 - Documents, images, unstructured data
 - Optical (CD, DVD) storage
 - Network (NFS, SMB) storage
- High-level objects
 - Database
 - Programs
 - Many predefined object types
- All file and object systems in one name space

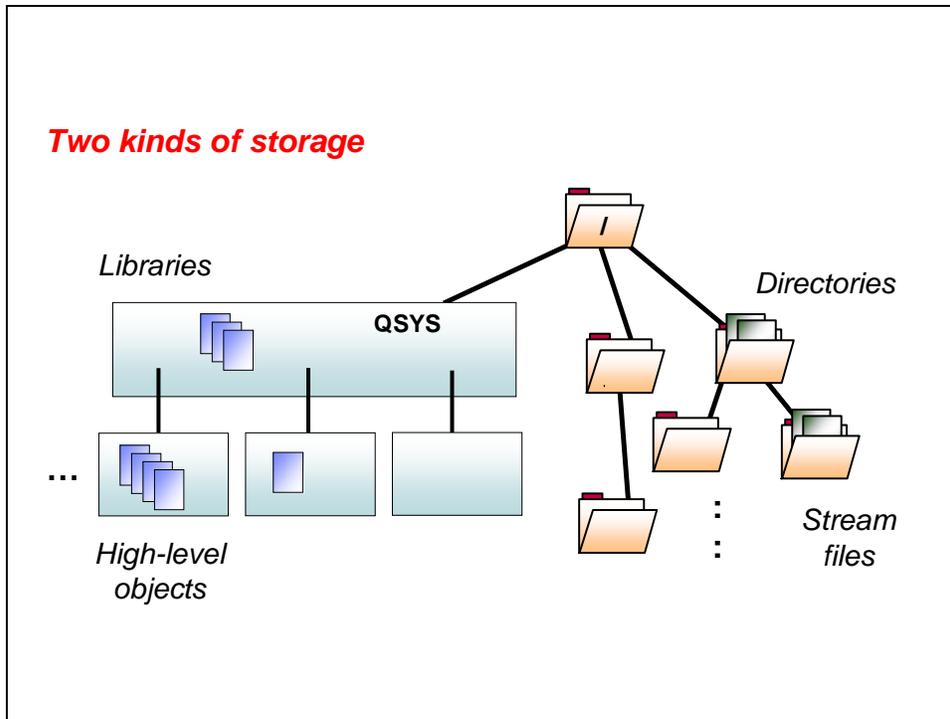


i5/OS storage: Integrated file system

All of these file systems reside under one name space on i5/OS, referred to as the *integrated file system (IFS)*.

The two most common systems are the root file system and QSYS.LIB:

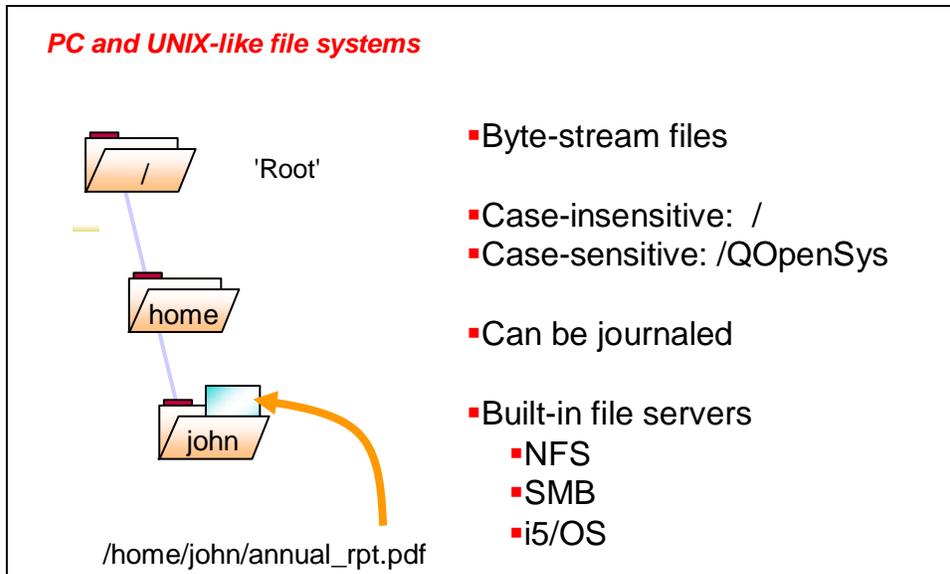
- The **root file system** is a basic file system for storing nonrelational data, where the information is simply a stream of bytes. In this way, it operates much like a Windows platform-based PC file system, although without the drive-letter designations. File and folder names can be quite long; they do not require a specific format, and are not case-sensitive. To the operating system, everything in the root file system is a file, a directory or a link. Some examples of what you can store in the root file system include: static HTML pages, MP3 files and Java class files.
- **QSYS.LIB** is a space for storing many types of high-level objects that i5/OS understands. These objects are specific, system-defined object types, such as: database tables, user profiles, programs, queues, queries and devices. The names of objects stored in QSYS.LIB are restricted to 10 characters and are not case-sensitive.



Key file systems

Think of i5/OS as providing two main types of data storage, high-level objects and stream files. Both of these file systems are used by most applications, as needed. If an application needs a flat-file system of directories and files, then the stream files noted on the right are used. When an application needs access to high-level objects such as data queues, job queues or subsystem descriptions, then the QSYS system on the left is used.

- **High-level objects:** Objects such as relational database (DB2) tables, user profiles, programs and output queues, are managed as objects in i5/OS libraries. The *library* is the container for all kinds of objects. Many operations on i5/OS deal exclusively with objects in libraries. This means the only reference to the location of an object is the library in which it resides. All libraries other than the QSYS library are organized at the same level; only the QSYS library can contain another library.
- **Stream files:** Stream files and the tree of directories that contain them are similar to UNIX and Windows file systems. The i5/OS operating system sees stream files as nothing more than chunks of contiguous data. The directory structure can be arbitrarily deep, limited only by the total amount of storage on the system. This kind of structure also provides access to the CD or DVD drives, which are located under a special directory called /Qopt. (This is discussed in greater detail shortly.)

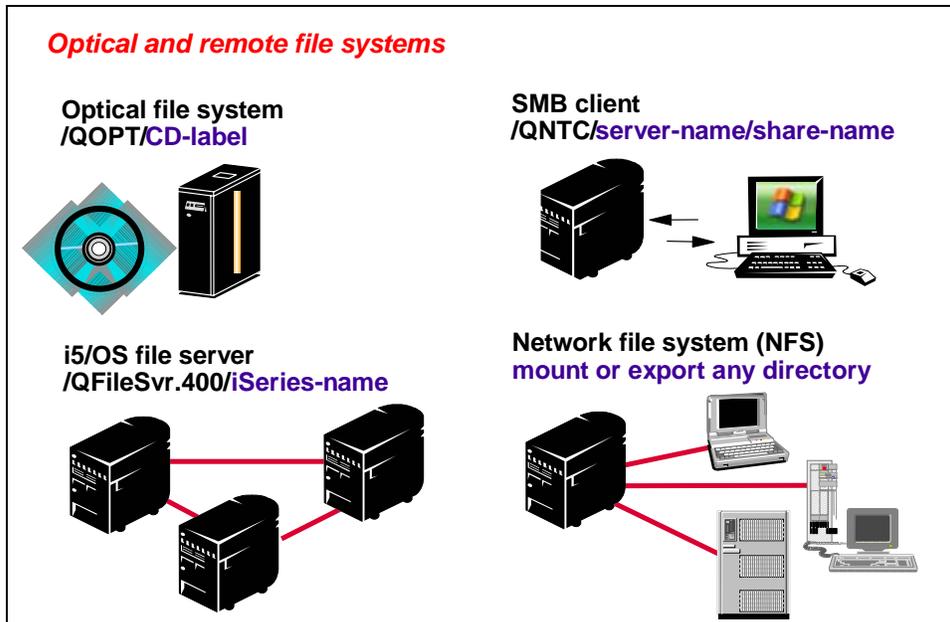


PC and UNIX style file systems

The root and QOpenSys file systems provide a directory structure, interfaces and features that are similar to those found on PCs and UNIX workstations. Directories can be nested to many levels, and file and directory names can be long, with or without file-name extensions. There is also the ability to store a file only once, yet access it through multiple paths by using links. It is also possible to access control by using mode bits. Additional fine-grained authorization is easily added by using the GUI.

Root: This file system is designed to take full advantage of stream-file support and the hierarchical directory structure of the IFS. It has the characteristics of recent Windows file systems, because it is case-insensitive.

QOpenSys: This file system is compatible with open-system standards such as POSIX and XPG. Similar to the root file system, this system takes advantage of the stream file and directory support that are provided by the IFS. In addition, it supports case-sensitive file names.



Optical and remote file systems

The **Optical (QOPT)** file system provides access to stream data that is stored on optical media:

- Example file name: /QOPT/VOL1/JOHN/EXEC (where VOL1 is an optical volume name)
- CD, DVD or attached 3995 optical libraries

The **i5/OS file server (QFileSvr.400)** file system provides access to other file systems that reside on remote System i servers:

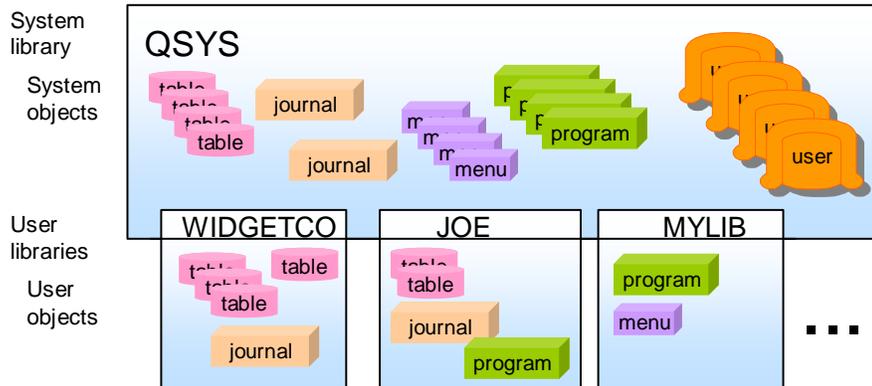
- Example file name: /QFileSvr.400/as4567/home/john/exec (where as4567 is a remote location name)
- Allows transparent access to data on remote System i servers
- The characteristics of this file system depend on the file system being accessed on the target system

The **QNTC** file system provides access to files that are stored on an SMB server:

- Example file name: /QNTC/Servername/Sharename/Dir/File
- To connect, make a directory (using the i5/OS mkdir command) in /QNTC that matches the remote-server name and share name.

The **Network file system (NFS)** provides the user with access to data and objects that are stored on a remote NFS server. It is possible to distribute data across a network by mounting file systems from remote NFS servers or allowing remote NFS clients to mount exported local file systems. Mounted file systems appear to be local.

i5/OS library and object space: /QSYS.LIB



This system of **objects in libraries** was the original storage system for AS/400.

Many operations only apply to **objects in libraries**.

i5/OS library and object space: /QSYS.LIB

Many operations on the system, such as creating a user or querying a database table, use only high-level objects to accomplish a task. There are many kinds of these high-level objects. For example, a user profile is a kind of object known to i5/OS. Another kind of object is a database table.

The space for high-level objects in the system is QSYS. In this space, objects are organized into libraries. In this context, a library is a container for objects. Only the QSYS library can contain other libraries. The organizational hierarchy is only one level deep. Thus, all objects in this space can be identified as being inside just one library. Every object has a distinct object type, as well. Combined with the object name, every object on the system can be uniquely identified with all three of these items: the object name, the object type and the name of the library in which it exists.

Note: Each job maintains a current library, in addition to a current directory.

DB2 UDB for i5/OS

- ✓ Standards
- ✓ Performance
- ✓ Integration
- ✓ Interoperability
- ✓ Security
- ✓ Flexibility
- ✓ Ease of administration

Structured data



BUILT-IN

- Open-standard SQL
- Open SQL interfaces
 - CLI - JDBC - ODBC - DAL - DRDA
- Multilanguage support
- Online backup and restore
- Parallel data access
- Referential integrity
- Stored procedures
- Two-phase commit
- System catalog
- Multi-point triggers
- Scrollable cursors
- Recovery optimizer
- SQL optimizer
- Automatic replication
- Query governor
- C2 security
- Advanced indexing technologies
- Cluster support

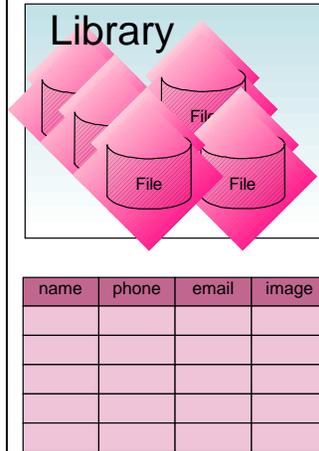
DB2 UDB for i5/OS

Every System i model includes a built-in DB2 engine that provides all of the common SQL interfaces, such as ODBC, JDBC, and CLI. Advanced features (for example, two-phase transaction management, parallelization, distributed data support and multipoint triggers) are all standard and built-in.

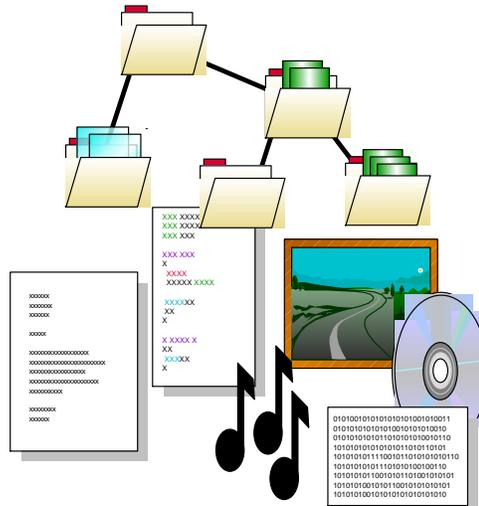
Database management is essentially not an issue. The system autonomously manages space allocation and ensures data integrity. The database engine is built into i5/OS, a self-managing operating system.

DB2 file objects or stream files

Structured relational files (DB2)
QSYS



Unstructured stream files
Root, QOpenSys



DB2 file objects or stream files

An important distinction exists between files that contain structured relational data and stream data. As shown on the left side of this diagram, structured relational data files are tables. They have a well-defined layout and manage data as records or rows. These files are objects inside of libraries in the QSYS object space.

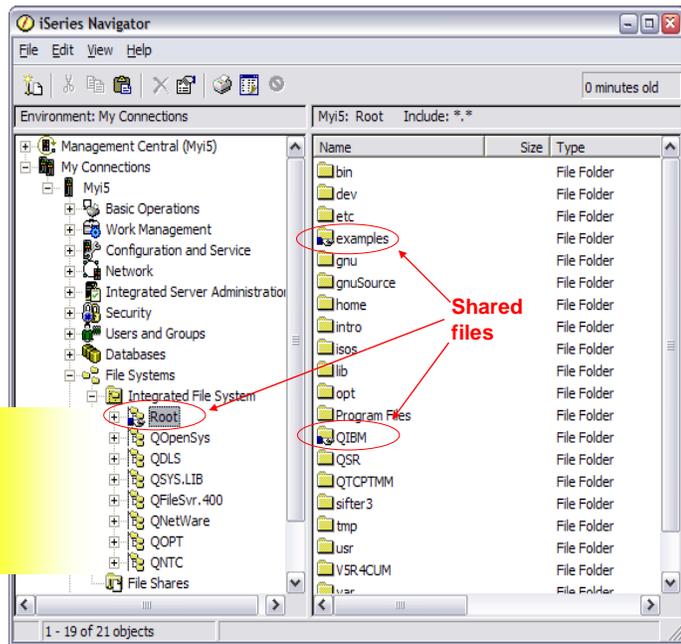
The right side of this same diagram portrays unstructured stream data. These kinds of files are stored in the root and QOpenSys file systems.

IFS view from Windows client

Also accessible from:

- Linux Samba client
- Windows file sharing
- NFS

Individual file systems



IFS (root) view from Windows client

Here, you can see the IFS using the i5/OS graphical client for Microsoft Windows — iSeries Navigator. This client allows you to manage the files, folders and objects anywhere in the IFS.

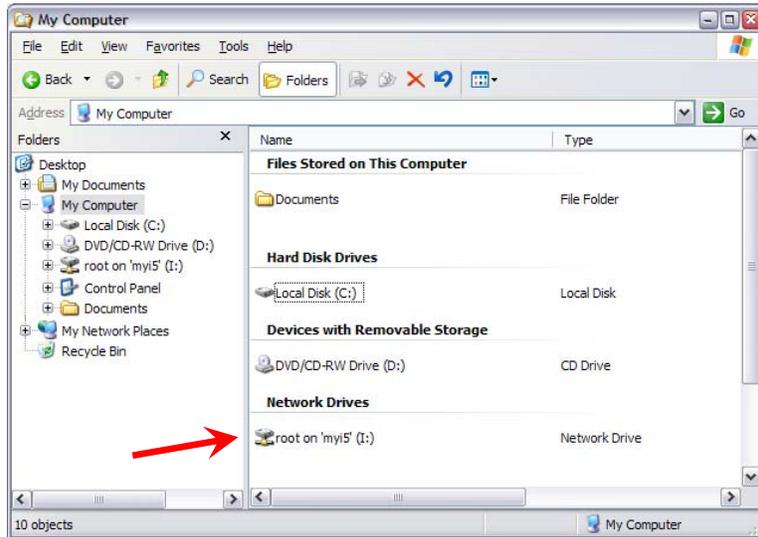
You can also see that there are some shared folders. From this graphical client, you can easily enable or disable sharing. As with most things related to i5/OS, support for file sharing is built-in and ready to use.

Of particular interest in this screen capture, you can see that the top directory of the root file system ("/") has been specified as *shared*. Also, the /QIBM folder, a directory in Root, is shared by default. This contains the client-side code to install iSeries Navigator, part of iSeries Access for Windows.

Next, you will examine the original, object-based structure of i5/OS.

File-system access: Clients

- iSeries NetServer
- Windows network drive
- Linux Samba client
- MacOS X

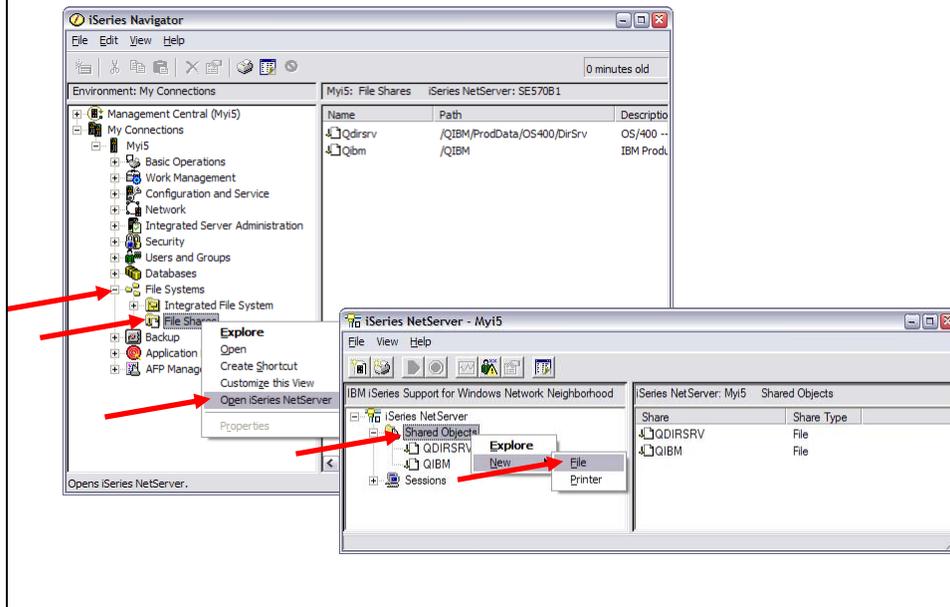


File-system access: Clients

The i5/OS operating system has the built-in ability to serve its file system to clients. This component, called IBM iSeries NetServer™, uses the server-message block (SMB) protocol. Support for mapping a drive on a Windows client is a standard part of recent versions of Windows. For Linux, the Samba package provides connectivity to this i5/OS file and print server.

To share a specific folder (directory), use iSeries Navigator, as already explained. Right-click the folder and select **Sharing**.

Configure i5/OS file server (iSeries NetServer)



Configure i5/OS file server (iSeries NetServer)

To create a new file share using iSeries NetServer, perform the following steps:

1. In the left-hand pane of iSeries Navigator, expand **File Systems --> File Shares**.
2. Right-click **File Shares** and select **Open iSeries NetServer**.
3. Within the iSeries NetServer window, select the icon in the upper left, or, right-click **Shared Objects** and select **New -> File**.
4. You are presented the iSeries NetServer File Share window, as shown on this set of screen captures. Enter the characteristics for the file share that you want to create.

i5/OS built-in security

- Single implementation covers all application models and objects, including database
- Implemented in lowest system levels
 - Enforced by hardware
 - Not an add-on layer
- User and group authorizations assigned to all objects and files
 - i5/OS permissions
 - UNIX mode bits
- Global authentication
 - Kerberos
 - Enterprise identity mapping

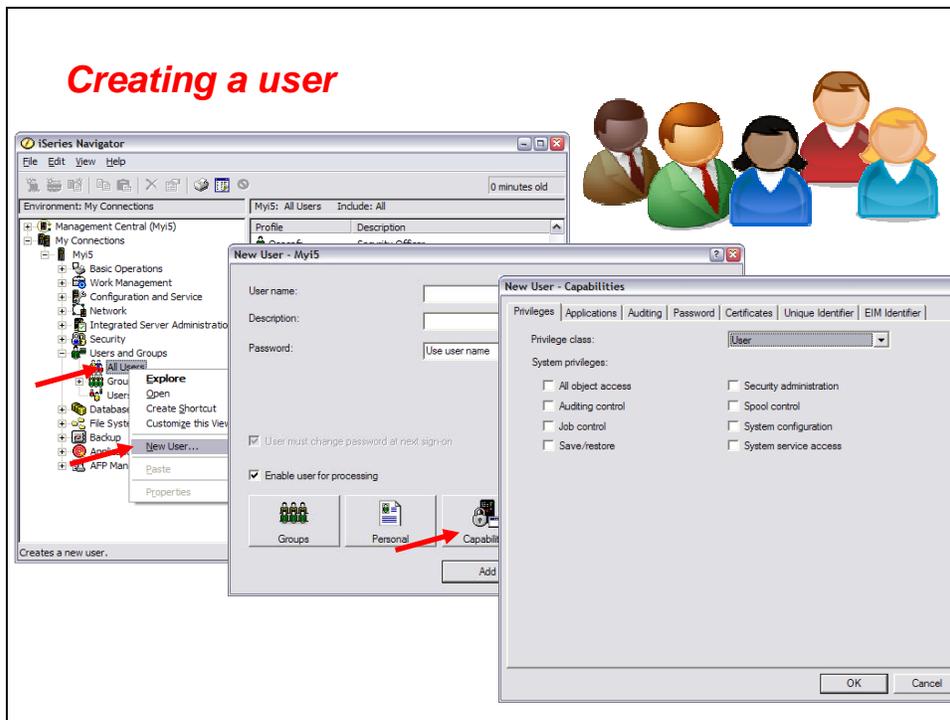


i5/OS built-in security

The single-implementation security offered by the System i platform covers all application models and objects, including the database. Security is implemented in the lowest system levels and is enforced by the hardware. Security is not an add-on tier.

i5/OS offers multiple layers of security. For example, specific authorization to objects is required and some operations require special authorities.

Extensive logging and auditing is selectively available. This is important for the reporting and tracking of unauthorized attempts to access data. Other security implementation strategies enforce strictly controlled management of user IDs and passwords, as well as any changes to the data itself and to permissions related to creation, alterations and deletions to any of the above.

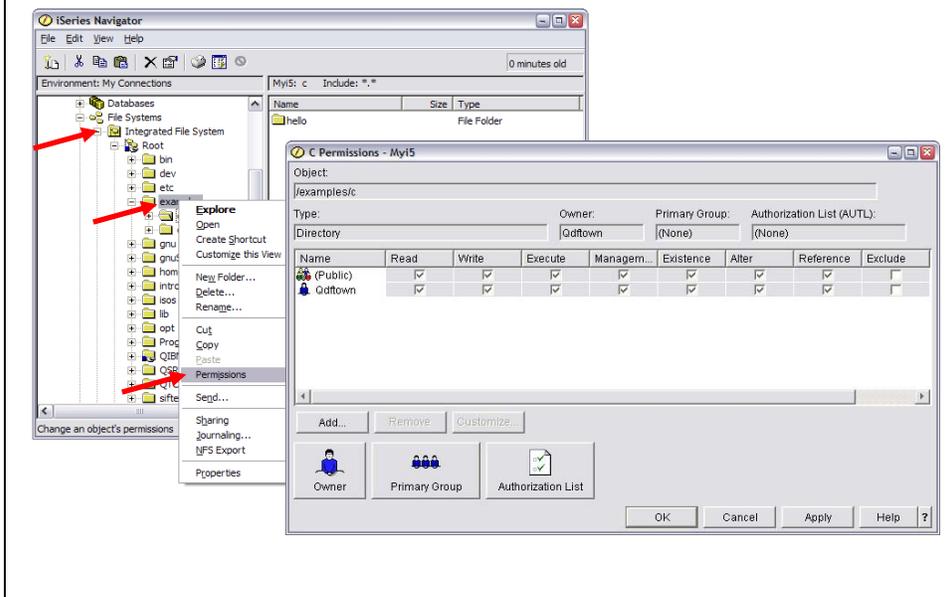


Creating a user

You can create a user with iSeries Navigator — providing that you have the proper authorization. Here is how you do it:

1. Click the plus symbol (+) next to your system name to display the functions tree underneath it.
2. Click the plus symbol (+) next to Users and Groups.
3. Right-click **All Users** and select **New User** from the context menu.
4. If you are creating an administrator-level profile, then on the Capabilities panel, change the Privilege class (user class) to **Security Officer**.

Setting permissions



Setting object permissions

You can change resource security with iSeries Navigator — providing that you have the proper authorization. Here is how you do it:

1. Click the plus symbol (+) next to your system name to display the functions tree underneath it.
2. Click the plus symbol (+) next to File Systems, then select the object on which you want to change authorizations.
3. Right-click the object and from the context menu, select **Permissions**.

Note: The display can vary from the screen captures shown here, depending on the type of file system that you select.

Advantage: IBM System i



- integration
- high availability
- leading 64-bit technology
- object-based applications
- multiple OS + DLPAR
- simplicity
- capacity on demand
- bullet-proof security
- self-managing
- growing market



Summary

The IBM System i platform brings new capabilities to a proven business-computing platform. The strength of the platform is its flexibility in meeting the varying needs of on demand business computing: exactly what you need, when you need it.

Every System i model runs the i5/OS operating system as its primary operating system. In addition, AIX, Windows and Linux applications also run on this family of IBM systems.

i5/OS manages several types of data that are primarily organized into two file systems. The QSYS object space provides storage for traditional OS/400 objects that are contained in libraries. The root file system contains byte-stream files, similar to UNIX file systems, which reside in directories.

The operating system has a built-in DB2 database, complete facilities for managing workloads across one or many systems, and several other enterprise-computing capabilities.

Programs that run on these systems fall into one of three categories. Traditional programs are objects that reside in libraries and can be created from C/C++, COBOL, RPG and a few other languages. Java classes, which are identical to Java classes on any other platform, are stored as files in the root file system. Similarly, i5/OS PASE programs are stored as files in the root file system.

Hopefully, this course has helped you as you begin to explore the vast possibilities provided by this integrated family of IBM systems. Now that you have some background information on the System i hardware and i5/OS software, you might be asking yourself, "Where can I find further details?" The starting point for all information for these systems is the IBM eServer™ iSeries Information Center (available at <http://publib.boulder.ibm.com/series/>), which is the primary means of distributing all System i technical information. It contains links to all the product manuals related to i5/OS and the IBM products that run on the i5/OS operating system.

Note: The eServer iSeries Information Center is also available as a CD and is shipped with every System i model. It is also separately orderable (SK3T-2027).

Resources

- IBM System i Information Center
<http://publib.boulder.ibm.com/series/>
- IBM Publications Center
www.elink.ibm.link.ibm.com/public/applications/publications/cgibin/pbi.cgi?CTY=US
- IBM Redbooks®
www.redbooks.ibm.com/
- System i Performance Management Reference manual
ibm.com/eserver/series/perfmgmt

Trademarks and special notices

© Copyright IBM Corporation 2007. All rights Reserved.

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

AIX, AIX 5L, AS/400, AS/400e, DB2, i5/OS, IBM, the IBM logo, iSeries, NetServer, OS/400, POWER, POWER3, POWER4, POWER4+, POWER5, POWER5+, POWER Hypervisor, PartnerWorld, Redbooks, System/36, System/38, System i, System p, System x, System z, WebSphere, X-Architecture and z/OS i are trademarks of International Business Machines Corporation in the United States, other countries, or both.

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

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

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

Information is provided "AS IS" without warranty of any kind.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.