



INTRODUCCION A AS/400¹

INDICE

1. Introducción a AS/400

- 1.1 [Conceptos Generales](#)
- 1.2 [Agrupaciones de Disco, SLS, Protección](#)
- 1.3 [Pools de Memoria](#)
- 1.4 [LPARs](#)
- 1.5 [Distribución de AS/400 en CAT](#)

2. Objetos

- 2.1. [El Concepto de Objeto en AS/400](#)
- 2.2. [Tipos de Objetos y Comandos](#)
- 2.3. [Bibliotecas](#)
- 2.4. [Lista de Bibliotecas](#)

3. Archivos

- 3.1. [Gestión y Sistemas de Archivos](#)
- 3.2. [Gestión de Archivos de Bases de Datos](#)
- 3.3. [Archivos Físicos](#)
- 3.4. [Archivos Lógicos](#)
- 3.5. [Archivos de Pantalla \(Display Files\)](#)
- 3.6. [Archivos de Impresora \(Printer Files\)](#)
- 3.7. [Archivos de Salvado \(Save Files\)](#)
- 3.8. [El Sistema de Archivos Integrado \(IFS - Integrated File System\)](#)

4. Subsistemas y Trabajos

- 4.1 [Concepto de Subsistema](#)
- 4.2 [Descripción de un Subsistema](#)
- 4.3 [El concepto de Trabajo](#)
- 4.4 [Propiedades de un Trabajo](#)
- 4.5 [El Ciclo de vida de un Trabajo](#)

5. Seguridad

- 5.1 [Permisos sobre Objetos](#)
- 5.2 [Lista de Autorizaciones](#)
- 5.3 [Perfiles de Usuario](#)

6. Impresión

- 6.1 [Conceptos Básicos](#)
- 6.2 [Archivos de Impresora \(Printer Files\)](#)
- 6.3 [Colas de Salida \(Output Queues\)](#)
- 6.4 [Printer Writer](#)
- 6.5 [Colas de Salida Remotas \(Remote Output Queues\)](#)

¹ ACCENTURE – BS. AS. SERVICE CENTER - 2006



7. Mensajes

- 7.1 [Tipos de Mensajes](#)
- 7.2 [Descripciones de Mensajes \(Message Descriptions\)](#)
- 7.3 [Archivos de Mensajes \(Message Files\)](#)
- 7.4 [Colas de Mensajes \(Message Queues\)](#)

APENDICE

- A.1 [Links](#)
- A.2 [Document Change Log](#)



1. INTRODUCCION A AS/400

1.1. Conceptos Generales

El AS/400 (Application System/400, también conocido como *iSeries*) es un sistema propietario de IBM. Esto significa que la máquina es de IBM, el sistema operativo también es provisto por IBM, al igual que la base de datos, y el software está hecho por empresas certificadas por IBM.

El sistema operativo que corre se denomina OS/400 (Operating System/400), y está orientado a objetos, lo cual se verá más adelante.

El AS/400 permite definir particiones lógicas (**LPARs** – Logical Partitions), las cuales son tratadas como máquinas lógicas independientes, las cuales pueden correr diversos sistemas operativos. Esto significa que se ven como máquinas aisladas, pero en realidad comparten los recursos físicos entre sí (memoria, disco, procesador), de esta manera se da la idea de tener varias máquinas corriendo en un mismo AS/400 ("máquinas virtuales").



- Foto de un AS400 -



1.2. Agrupaciones de discos, SLS, Protección

La información en el AS/400 está o en el disco (DASD – *Direct Access Storage Device*) o en memoria. El AS/400 usa una arquitectura para manejar el almacenamiento conocido como **Single Level Storage** (SLS). SLS trata toda la memoria direccionable (tanto disco como memoria) como un solo espacio de direccionamiento. Esto permite más flexibilidad y efectividad en el uso del espacio de direccionamiento del sistema.

El espacio de almacenamiento en el AS/400 puede ser dividido en grupos de discos llamados ASP (*Auxiliary Storage Pools*). Siempre hay un ASP, conocido como el ASP del sistema, pero puede haber hasta 16 : 15 de usuario y uno de sistema.

- El ASP 1 es el de Sistema, el cual contiene el "*Load Source*" en un disco (que está espejado), que es la parte del sistema operativo que permite arrancar la máquina. (también contiene datos)
- El resto de los ASP (ASP 2, ASP 3, etc) son de "usuario". Usados para guardar otros tipos de información.

Cuando el espacio en el ASP de usuario se llena al 100%, esa información comienza a pasar al ASP de sistema. Estos datos se conocen como datos de Overflow.

Si el ASP de sistema llega al 100%, el sistema se detiene, dado que no tiene donde hacer el overflow de los datos.

Un efecto secundario que ocurre cuando el DASD llega cerca del 100% es que deja de funcionar el SNADS (*System Network Architecture Distribution Services*), el cual provee la funcionalidad para mandar y recibir mails y mensajes fuera del sistema. Esto esta dado por lo que se denomina el *threshold* (limite) del ASP. Para el ASP de sistema el threshold es del 90%.

Protección

Hay varias formas de proteger los datos en DASDs en el AS/400, las más usadas son:

1. **Mirroring**: La protección *Mirrored* ("espejada") es una función que aumenta la disponibilidad del AS/400 cuando se produce un fallo en algún componente del hardware relacionado con el disco (como el bus, procesador IOP, controladora). Hay varios niveles de Mirroring posibles, dependiendo en el componente que se duplique. Esto permite que ante la falla de algún componente el sistema siga estando disponible, si dicho componente está duplicado.
 - **DASD Mirroring dentro de un ASP**: Este nivel de Mirroring, conocido como *power domain mirroring* ocurre cuando los IOPs (*Input/Output Processors* - *Procesadores de Entrada/Salida*) del DASD y sus buses están conectados y aislados a diferentes fuentes de energía (racks, o torres). Este nivel de protección permite que si se cae una fuente, el sistema siga disponible. Es una protección a nivel de bus.
 - **DASD Mirroring remoto de todo el sistema o de ASPs**: La idea detrás de este nivel de Mirroring es tener un grupo de discos en sitios remotos a través de buses ópticos, y otros en buses locales. Esto permite que el sistema siga disponible si falla alguno de los dos.



2. **Protección de Dispositivos con Paridad (RAID-5)**: La protección de dispositivos mediante paridad es una función del hardware que protege la pérdida de los datos debido a fallas o daños en los discos. Para tener una configuración de este tipo se requieren como mínimo 4 discos.
- El RAID-5 funciona de la siguiente manera: suponer que se tienen 4 discos conectados, por cada sector en cada disco hay o datos o "paridad". Tomando el primer sector por ejemplo de cada uno de los discos, en 3 de ellos va a haber datos, y en el cuarto habrá datos de "paridad", los cuales se calculan mediante una función lógica. Esto permite que se puedan reconstruir los datos si un disco falla, mediante esta información de paridad.
- El hecho de repartir la información de paridad en todos los discos permite que haya mejor performance cuando falla una unidad, porque no todos los sectores deben reconstruirse ya que para los que tienen información de paridad en el disco caído la información no se perdió.
- Este tipo de configuración también se conoce como DPY (Device Parity).

1.3. Pools de Memoria

La memoria en el AS/400 está dividida en *Pools de Memoria (Memory Pools)*, los cuales son divisiones de memoria principal que son usados para el procesamiento de trabajos.

Se puede controlar la cantidad de trabajo en un subsistema mediante el ajuste del tamaño y número de los pools de memoria. Cuánto más grande sea el tamaño de los pools de memoria, más trabajo se podrá hacer.

Los pools de memoria tienen definidos *Niveles de Actividad (Activity Levels)*. El Nivel de Actividad de un pool es la cantidad de hilos (threads) que pueden usar la CPU simultáneamente en un pool de memoria.

El número de hilos corriendo (*Active Threads*) se refiere a la cantidad de hilos que están disponibles para ser seleccionados por el procesador y que cuentan al momento de calcular el nivel de actividad del pool. En este sentido, los hilos activos no incluyen hilos esperando por Entrada/Salida, por mensajes, para un dispositivo a ser alocado, o por un archivo a ser abierto. Los hilos activos no incluyen hilos que son inelegibles (*Ineligible*), hilos que están listos para correr cuando el nivel de actividad del procesador está al máximo (o sea, el procesador no puede correrlos debido a el límite de su nivel de actividad).

Entre los hilos que están corriendo se pueden producir varios tipos de transiciones:

Active a Wait – Esto se produce cuando al job activo se le terminó el *Time Slice* (concepto que se verá cuando se vean jobs), sería el tiempo de procesamiento del hilo.

Active a Ineligible – Este caso se produce cuando, por problemas de performance, el sistema ajusta los pools de memoria y los niveles de actividad, sacando al hilo de su ejecución.

Wait a Ineligible – Es el peor caso de todos, ocurre cuando el hilo está en espera del procesador, y por problemas de performance, es puesto como *Ineligible*.

- **Pools de Memoria compartidos (Shared Storage Pools)**: Un pool de memoria compartido es un pool en el que múltiples subsistemas pueden ejecutar trabajos. Pueden ser tanto especiales o generales. El *Machine Pool* y el *Base Pool* son considerados pools de memoria especiales, el resto son generales.



Especiales:

**MACHINE*: Este pool de memoria es usado para programas del sistema operativo. Los jobs que corren en este pool generalmente no necesitan atención. Se identifica como el pool número 1.

**BASE*: El pool *BASE contiene toda la memoria no asignada en el sistema. Puede ser compartido por varios subsistemas. Se identifica como el pool número 2.

Generales:

Los pools compartidos generales son pools que pueden ser usados por varios subsistemas al mismo tiempo.

**INTERACT* es el pool usado para jobs interactivos. (los conceptos de jobs se verán más adelante).

**SPOOL* is el pool usado para impresoras y spools.

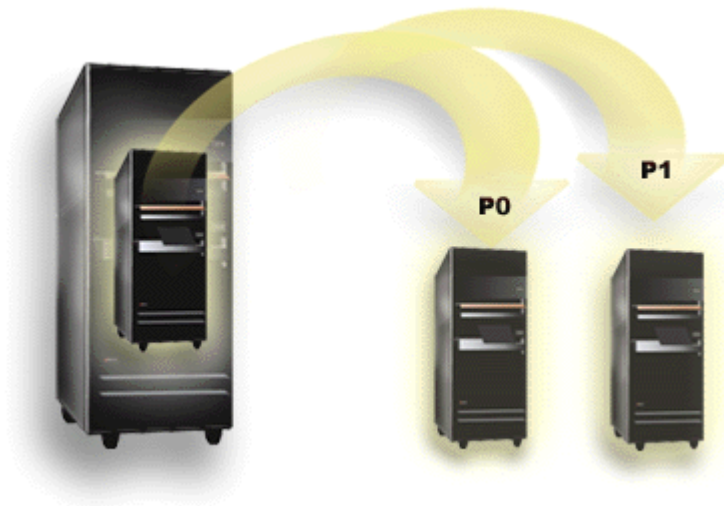
**SHRPOOL1* hasta **SHRPOOL60* son pools que se pueden usar para uso propio.

- **Pools de Memoria Privados** (*Private Storage Pools*): Un pool privado de memoria es un pool en el que sólo un subsistema puede correr trabajos. Se pueden asignar hasta 62 de ellos.

Si la información se encuentra en la memoria principal, puede ser referenciada independientemente del pool en el cual está. Sin embargo, si la información requerida no existe en ningún pool, se trae desde disco hacia el mismo pool donde está el job que la referencia. Esto se conoce como *Fallo de Página (Page Fault)*, porque implica traer de disco las páginas a memoria. Cuando está información se trae al pool de memoria, otros datos se desplazan al disco, esto se conoce como *Paginación (Paging)*. El nivel de paginación debería estar en niveles razonables para no afectar la performance.

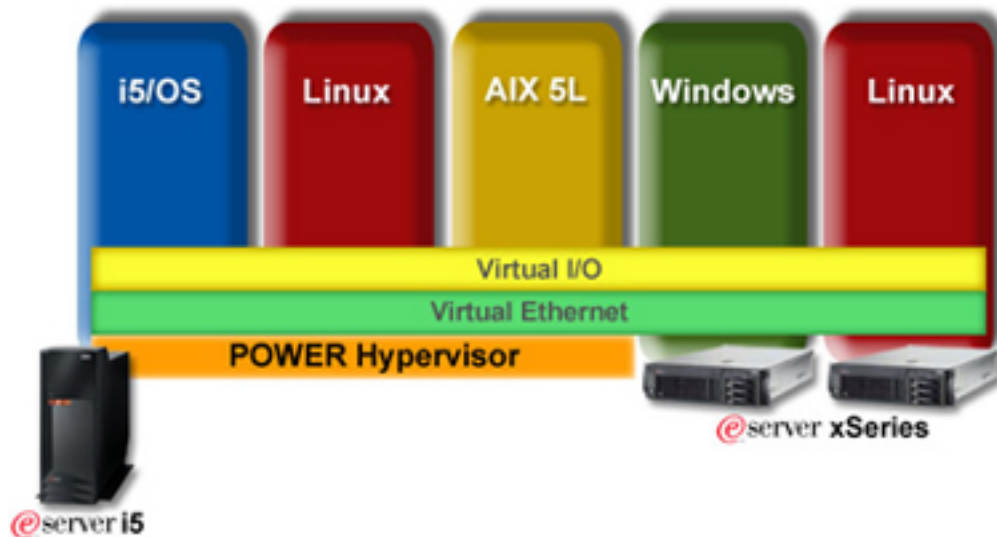
1.4. LPARs (Logical Partitions)

El AS/400 permite definir particiones lógicas (**LPARs** – Logical Partitions), las cuales son tratadas como máquinas lógicas independientes, las cuales pueden correr diversos sistemas operativos. Las LPARs permiten tener varias instancias de OS/400, vistas como máquinas independientes. Permite virtualizar los recursos de hardware para que sean compartidos por estas máquinas independientes.



- Concepto de LPARs -

Las LPARs permiten que un servidor se divida en varios servidores virtuales independientes, corriendo así diferentes sistemas operativos.



Con la introducción del *i5 (iSeries 5ta Generación)* soporta hasta 254 particiones lógicas. Para cada partición se asigna cierta cantidad de procesadores (el AS/400 tiene múltiples procesadores), o incluso se puede asignar hasta un porcentaje de cierto procesador (75% por ejemplo). Con la tecnología i5, hay 3 formas de usar procesadores en cada partición:

- Modo Dedicado
- Modo Compartido/Capped
- Modo Compartido/Uncapped

Capped y *Uncapped* son modos de especificar el uso compartido del procesador para cada LPAR.



1. **Procesadores Dedicados**: Los procesadores asignados a particiones dedicadas no pueden ser usados por otra partición mientras esté ejecutándose. Con i5, si una partición es apagada, su procesador se vuelve disponible para su uso por otra partición usando el Modo Compartido Uncapped. Cuando vuelve a encenderse, reobtiene los procesadores.
2. **Procesadores Compartidos/Capped**: Los procesadores usando este modo son asignados desde el pool compartido de procesadores. Una partición usando procesadores en este modo tiene garantizado el uso de la cantidad asignada de procesador. Sin embargo, si la partición Capped está inactiva, una partición Uncapped puede usar esos procesadores hasta que la partición Capped vuelva a estar activa.
Los procesadores Power5 (que vienen con i5) pueden ser sub-divididos en un décimo de procesador (0.1) hasta 10 procesadores virtuales.
3. **Procesadores Compartidos/Uncapped**: Los procesadores usando este modo también son asignados del pool compartido de procesadores. Una partición en modo Uncapped tiene garantizado el uso total de sus procesadores asignados si es necesario. (similar al modo Capped)
El modo Uncapped es único en el sentido de que si la carga de trabajo requiere más procesador, la partición puede adquirir procesador inactivo de particiones Capped y Uncapped, y procesadores de particiones dedicadas apagadas. Esto ofrece gran flexibilidad para el uso del procesador. Esto es lo que origina que la CPU se vea al 300%, ya que significa que está usando otros procesadores además de los que tiene asignados.

1.5. Distribución de AS/400 en Caterpillar

En CAT los AS/400 de cada dealer se encuentran en Solution Centers en varios puntos del mundo. Tienen una nomenclatura específica para identificar en qué locación se encuentra. La distribución es la siguiente:

Peoria (EEUU)

Nomenclatura de las LPARs: (4 AS/400)

- CATS4x – FDC (ej. CATS42, CATS4N)
- CATS5x – PRD (ej. CATS55, CATS5T)
- CATS6x – FDC
- CATS7x – PRD

Melbourne (Australia)

Nomenclatura de las LPARs: (1 AS/400)

- ASC1x (ej. ASC1A)

Gosselies (Bélgica)

Nomenclatura de las LPARs: (1 AS/400)

- ESC2x (ej. ESC2G)

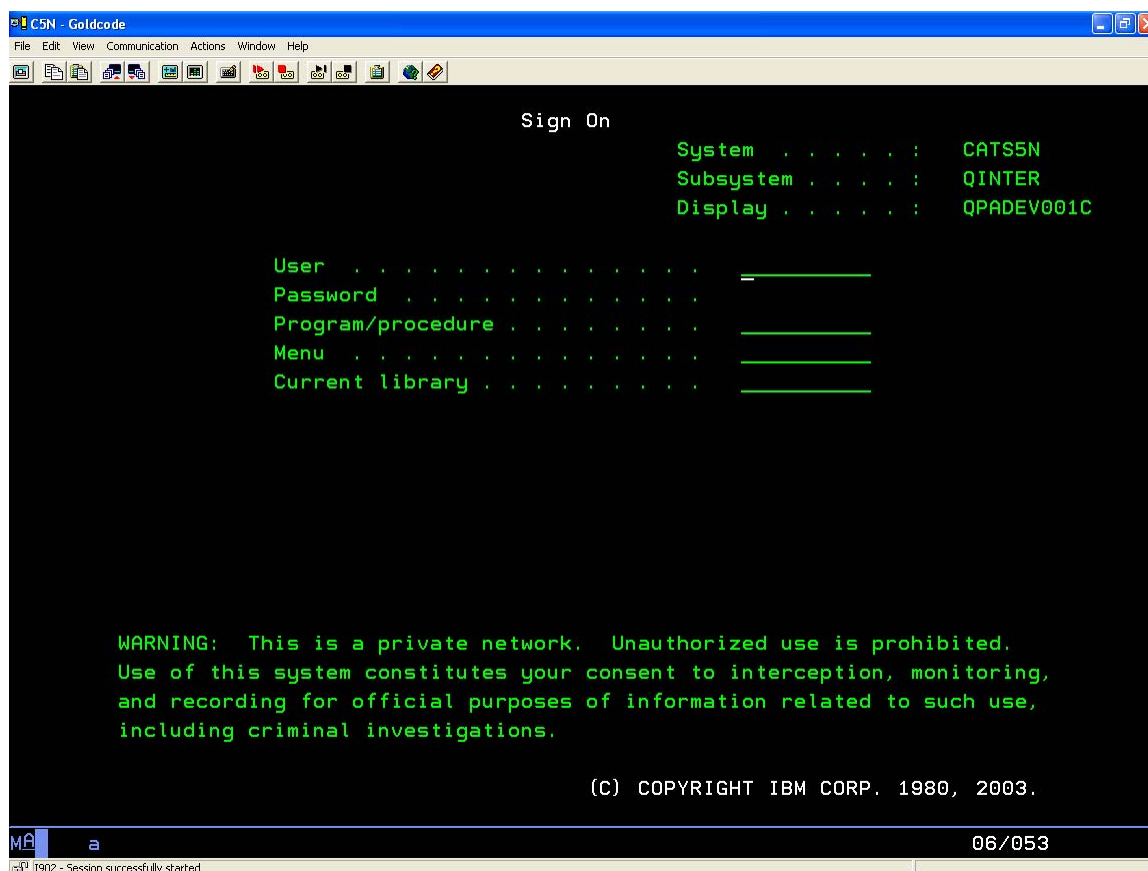
Cada LPAR con la misma nomenclatura se encuentra en el mismo AS/400, compartiendo los recursos físicos.

2. OBJETOS

2.1 El concepto de objeto en AS/400

Antes de explicar lo que es un objeto, hay que loguearse en el AS/400. La conexión al AS/400 se hace mediante el Client Access de IBM (pueden usarse otros clientes).

La primer pantalla que va a aparecer al abrir la sesión va a pedir usuario y clave:



Después de poner usuario y clave va a aparecer el menú principal para empezar a trabajar con el AS/400.



2.2 Tipos de Objetos y Comandos

El AS/400 tiene una serie de menús para trabajar con él pero básicamente el trabajo se hace entrando **Comandos**, con los cuales se manipula el sistema.

En el AS/400 todo es un **Objeto**, el cual tiene un identificador único. Uno de los atributos de los objetos es el *Tipo*. Hay varios tipos de objetos en el sistema.

- Programas (*PGM)
- Archivos (*FILE)
- Colas de Mensajes (*MSGQ)
- Módulos (*MODULE)
- Áreas de Datos (*DTAARA)
- Colas de Datos (*DTAQ)
- Colas de Salida (*OUTQ)
- Líneas de comunicación (*LIND)
- Controladores (*CTLD)
- Dispositivos (*DEVD)
- Bibliotecas (*LIB)
- Otros



Como se ve, los archivos, las bibliotecas, y los programas son tratados como objetos. Las bibliotecas contienen otros objetos, y las bibliotecas están todas contenidas en una biblioteca llamada QSYS.

Para trabajar con objetos, hay que usar comandos, los cuales son implementados a través del lenguaje CL (*Control Language*). Los comandos tienen una forma predefinida, en la cual se coloca el *verbo*, el cual indica la acción a realizar, y el *sustantivo*, que indica sobre qué tipo de objeto trabajar.

Los verbos más básicos son los siguientes:

- Visualizar (*Display*) DSP
- Cambiar (*Change*) CHG
- Crear (*Create*) CRT
- Borrar (*Delete*) DLT
- Trabajar (*Work*) WRK
- Agregar (*Add*) ADD
- Arrancar (*Start*) STR
- Finalizar (*End*) END
- Imprimir (*Print*) PRT
- Remover (*Remove*) RMV
- Salvar (*Save*) SAV
- Restaurar (*Restore*) RST

Los sustantivos (Objetos) más comunes son los siguientes:

- Programa PGM
- Bibliotecas LIB
- Cola de Salida OUTQ
- Cola de Mensajes MSGQ
- Archivo Físico PF
- Archivo F
- Líneas LIND
- Controladores CTLD
- Dispositivos DEVD
- Comandos CMD
- Perfil de Usuario USRPRF
- Archivo de Salvado SAVF

Entonces, juntando ambos se obtiene el comando para trabajar con cada tipo de objetos. Por ejemplo: WRKJOB (*Work with Job*), DLTF (*Delete File*), DSPUSRPRF (*Display User Profile*), etc.

Si al ejecutar un comando se presiona F4 en vez de Enter ("promptear" el comando) se obtiene una pantalla donde se pueden completar los campos de todos los parámetros. Con F1 se puede obtener ayuda por cada parámetro. Con F10 se amplían los parámetros que se muestran en pantalla.

Para trabajar con objetos en general, se usan los siguientes comandos:

WRKOBJ (*Work with Objects*)
DSPOBJD (*Display Object Description*)
DLTOBJ (*Delete Object*)
RNMOBJ (*Rename Object*)
Etc.



El comando WRKJOB permite trabajar con cualquier objeto del sistema, independientemente de su tipo. La pantalla que aparece al hacer WRKOBJ + F4 es la siguiente:

```
C5N - Goldcode
File Edit View Communication Actions Window Help

Work with Objects (WRKOBJ)

Type choices, press Enter.

Object . . . . . = Name, generic*, *ALL
Library . . . . . *LIBL Name, *LIBL, *CURLIB...
Object type . . . . . *ALL *ALL, *ALRTBL, *AUTL...

F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

Bottom

MA a 05/037
1902 - Session successfully started
```

Así permite poner los objetos con los cuales trabajar, puede ser **ALL* (todos los del sistema), *generic** que permite poner un nombre genérico o directamente el nombre del mismo. Si en Type se deja **ALL* va a traer todos los objetos con ese nombre, sin importar su tipo. Por ejemplo, al hacer **WRKOBJ OBJ(QUSRSYS/QAEZDISK)** se muestra el objeto QAEZDISK en la biblioteca QUSRSYS y los comandos disponibles para trabajar con este objeto:

```
Type options, press Enter.
 2=Edit authority      3=Copy    4=Delete    5=Display authority    7=Rename
 8=Display description 13=Change description

Opt Object   Type   Library   Attribute   Text
_  QAEZDISK  *FILE  QUSRSYS  PF          Output file for RTVDSKINF
```

Esto incluye poder renombrarlo, borrarlo, copiarlo, etc. Es el comando más genérico para trabajar con objetos.



2.3 Bibliotecas

Un tipo importante de objeto es la **Biblioteca** (*Library*). Las bibliotecas son esencialmente contenedores de otros objetos, y se pueden usar para referenciar otros objetos en el sistema. La única biblioteca que puede contener otras bibliotecas es llamada QSYS. Contiene todas las otras bibliotecas del sistema.

Por ejemplo, si se pone **WRKLIB QSYS** (*Work with Libraries*) y se selecciona **12=Work with Objects** se pueden ver todos los objetos dentro de la biblioteca QSYS.

```
C5N - Goldcode
File Edit View Communication Actions Window Help

Work with Libraries

Type options, press Enter.
 1=Create  2=Change  3=Copy   4=Delete  5=Display  6=Print
 8=Display library description  9=Save    10=Restore
11=Save changed objects      12=Work with objects  14=Clear

Opt  Library  Attribute  ASP Device  Text
---  -
12  QSYS      PROD      System Library

Parameters for options 1, 2, 3, 5, 8, 9, 10, 11 and 12 or command
==>
F3=Exit    F4=Prompt  F5=Refresh  F9=Retrieve  F11=Display names only
F12=Cancel F16=Repeat position to  F17=Position to

MA a 21/007
I902 - Session successfully started
```

Efectivamente, se podrá ver que contiene objetos del tipo *LIB, que son todas las bibliotecas del sistema:



Opt	Object	Type	Library	Attribute	Text
—	ABMISCSRC	*LIB	QSYS	PROD	
—	ABPGM500	*LIB	QSYS	PROD	
—	ABSARCHIVE	*LIB	QSYS	PROD	Aldon/CMS DJB/DBS (ARC)
—	ABSPSRC	*LIB	QSYS	PROD	
—	ABSRC500	*LIB	QSYS	PROD	
—	ABS600PDN1	*LIB	QSYS	PROD	
—	ACCSECURE	*LIB	QSYS	PROD	Accenture Security Library
—	ACEUTIL	*LIB	QSYS	PROD	text1
—	ACMSCLIB61	*LIB	QSYS	PROD	
—	ACMSCORPDN	*LIB	QSYS	PROD	Aldon/CMS ALDON/SITE/CORPOR
—	ACMSCTL	*LIB	QSYS	PROD	Aldon/CMS Control Library

More...

2.4 Lista de Bibliotecas

La **Lista de Bibliotecas** (*Library List*) en el AS/400 es similar al "path" en las PC's. Define el orden de bibliotecas en el cual el sistema operativo busca los objetos. (Es un objeto del tipo *LIBL). Cuando se ejecuta un comando o se llama un programa, el AS/400 tiene que saber donde buscar ese comando o programa. Puede especificarse la biblioteca explícitamente, poniendo el nombre de la biblioteca, o implícitamente. En este caso, el AS/400 va a buscar en orden biblioteca por biblioteca el objeto en la lista de bibliotecas. Usará el que encuentre primero. Para ver la lista de bibliotecas actual (de nuestro perfil de usuario) se tiene que ejecutar el comando DSPLIBL (*Display Library List*):

```

CSN - Goldcode
File Edit View Communication Actions Window Help
Type options, press Enter.
5=Display objects in library

System: CATS5N

Opt Library Type ASP Device Text
= RBTSYSLIB SYS Robot SYSTEM LIBRARY
- QSYS SYS System Library
- QSYS2 SYS System Library for CPI's
- QHLPSYS SYS
- QUSRSYS SYS System Library for Users
- TEMPLIBPS USR BA Production Services migration's libr
- SEQUEL USR
- LPDBS USR DBS Product Library
- QGPL USR General Purpose Library
- QTEMP USR
- SPPGM000 USR R0.00 Version Control Production Progra
- SPD000 USR

More...

F3=Exit F12=Cancel F17=Top F18=Bottom
(C) COPYRIGHT IBM CORP. 1980, 2003.
MA a 10/003
1902 - Session successfully started

```



En este ejemplo, se ven dos tipos de entradas en la lista de bibliotecas, pero lo común es tener 3 tipos:

1. SYS: son bibliotecas con objetos del sistema operativo. Acá residen los menús y comandos del AS/400. Es la Lista de Bibliotecas del *Sistema* (*System Library List*).
2. CUR: Es la biblioteca *Actual* (*Current*). Es la primer biblioteca de usuario, y donde se van a crear los objetos si no se especifica la biblioteca a usar.
3. USR: bibliotecas de *Usuario*.

También existe una cuarta entrada: la porción de *Producto*. Cuando un comando CL es ejecutado y especifica una biblioteca de producción, esa biblioteca va a ser añadida a la porción de Producto de la lista de bibliotecas por la duración del comando. Sigue a la porción de sistema y el sistema operativo puede mantener hasta dos bibliotecas en ella.

Hay varios comandos para trabajar con listas de bibliotecas, los más usados son:

DSPLIBL (*Display Library List*) – Muestra toda la lista de bibliotecas.

CHGLIBL (*Change Library List*) – Cambiar lista de bibliotecas.

ADDLIBL (*Add Library List Entry*) – Agrega una entrada a la lista de bibliotecas del usuario.

RMVLIBL (*Remove Library List Entry*) – Quita una entrada a la lista de bibliotecas del usuario.

EDTLIBL (*Edit Library List*) – Permite editar las bibliotecas de usuario (USR).

CHGCURLIB (*Change Current Library*) – Cambiar la biblioteca actual.

CHGSYSLIBL (*Change System Library List*) – Cambiar la lista de bibliotecas del sistema.

Con el comando EDTLIBL, se puede incluso cambiar el orden de las bibliotecas, para hacer que el sistema operativo busque primero en la que se desee.

Edit Library List					
					System: CATS5N
Type new/changed information, press Enter.					
Sequence		Sequence		Sequence	
Number	Library	Number	Library	Number	Library
0		150		300	
10	TEMPLIBPS	160		310	
20	SEQUEL	170		320	
30	LPDBS	180		330	
40	QGPL	190		340	
50	QTEMP	200		350	
60	SPPGM000	210		360	
70	SPD000	220		370	
80	TAAT00L	230		380	
90		240		390	
100		250		400	

El máximo de bibliotecas que puede tener la lista de bibliotecas de usuario es 250. La lista de bibliotecas puede estar especificada en otros lados, como ser la *Job Description* de un trabajo (se verá después), o en un valor del sistema (SYSVAL), el cual es un valor que aplica para todo el sistema.



3. ARCHIVOS

3.1 Gestión y Sistemas de Archivos

El sistema operativo OS/400 tiene varios módulos de gestión de archivos. Se puede dividir en la **Gestión de Archivos de Bases de Datos (*Database File Management*)** y en el **Sistema de Archivos Integrado (IFS – *Integrated File System*)**. También está la función para archivos de Spool.

3.2 Gestión de Archivos de Bases de Datos

Este módulo del sistema operativo es el que controla el manejo de los archivos tradicionales (de tipo *FILE) , permite hacer todas las operaciones con estos archivos (abrir, cerrar, etc.).

En el AS/400, cada archivo tiene una descripción que describe sus características y cómo los datos asociados con el archivo están organizados en registros, y en muchos casos, los campos de los registros. Esta descripción se usa cuando se usa el archivo.

Los archivos tienen atributos, que sirven para diferenciar más qué clase de archivo es. El tipo de objeto siempre es *FILE.

Hay varios tipos de atributos para los archivos, los más comunes son:

- | | |
|--|------|
| • Archivos Físicos (<i>Physical Files</i>) | PF |
| • Archivos Lógicos (<i>Logical Files</i>) | LF |
| • Archivos de Pantalla (<i>Display Files</i>) | DSPF |
| • Archivos de Impresora (<i>Printer Files</i>) | PRTF |
| • Archivos de Salvado (<i>Save Files</i>) | SAVF |
| • Otros | |

3.3 Archivos Físicos

Un **Archivo Físico (Physical File)** es un archivo de bases de datos que guarda datos de aplicaciones. Contiene la manera en la cual los datos serán recibidos o enviados a un programa y como la información esta actualmente guardado en la base de datos. Consiste de registros de longitud fija con campos que pueden tener longitud variable. Los archivos físicos tienen un formato de registro y uno o más **Miembros** (conjuntos de registros). Vistos desde la perspectiva SQL, los archivos físicos son idénticos a las tablas.

Al crear un archivo, hay dos maneras de hacerlo: una es crearlo con los registros de longitud fija. La otra manera es usar lo que se llama DDS (*Data Description Specifications*), que es un archivo fuente que describe cómo son los registros, sus campos, y la vía de acceso a los miembros.



3.4 Archivos Lógicos

Un **Archivo Lógico (Logical File)** es un archivo de base de datos que representa lógicamente uno o más archivos físicos. Contiene una descripción de *cómo* los datos van a ser presentados o recibidos desde un programa. No contiene datos, pero define los formatos de registro para los archivos físicos que representan. Básicamente, un archivo lógico no es más que una manera de acceder los datos del archivo físico con diferentes *campos clave*.

Desde la perspectiva SQL, los archivos físicos son idénticos a *vistas* e *índices*.

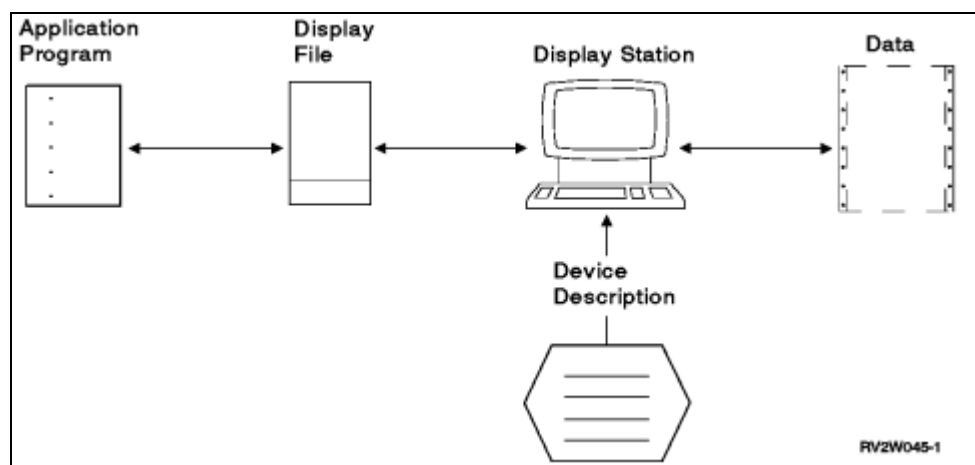
Para crear un archivo lógico, el/los archivo(s) físico(s) que representan deben existir.

3.5 Archivos de Pantalla

Un **Archivo de Pantalla (Display File)** define el formato de la información a ser presentada en la pantalla, y cómo esa información será procesada por el sistema desde y hasta la pantalla.

Las DDS describen los datos referidos por un archivo de pantalla.

Antes de que un programa trabaje con una estación de trabajo (*display station*) un archivo de pantalla debe ser abierto para permitir que los datos fluyan entre la estación de trabajo y los programas.



Una **Descripción de Dispositivo (Device Description)**, el cual es un objeto de sistema que describe la estación de trabajo al sistema, también debe existir para la estación de trabajo. Contiene información como la dirección del dispositivo, tipo de dispositivo, modelo, etc.

Cada archivo de pantalla tiene una *File Description* (creada al crear el archivo de pantalla), la cual describe las características del display file y determina el funcionamiento de éste.

Una *File Description* describe los datos en 3 niveles: a nivel de Campo (*Field-Level Description*), a nivel de Registro (*Record-Level Description*) y a nivel de Archivo (*File-Level Description*).

La descripción de cada uno permite darle al sistema las características de cada uno. Por ejemplo, la Field-Level Description informa entre otras cosas, donde se muestra el campo en pantalla; la Record-Level Description muestra el formato de los registros (*Record Format*); y las File-Level Description da información del archivo en su totalidad, como qué formatos de registros acepta, etc.

La descripción del archivo de pantalla puede estar *descripta externamente (Externally-Described)*, o sea, en un archivo de pantalla, o *descripta por programa (Program-Described)*, o sea, embebida en un programa.



3.6 Archivos de Impresora

Un **Archivo de Impresora (Printer File)** describe cómo el sistema opera con los datos a medida que pasan desde el programa de aplicación a la impresora.

Los Printer Files manejan todas las peticiones de impresión. Se pueden usar printer files provistos por el sistema, o se pueden crear nuevos mediante el comando CRTPRTF (*Create Printer File*).

Los printer files tienen varios parámetros que le dicen al sistema como la salida (output) debe ser formateado, qué fuentes usar para lo que se imprime, si imprimir en ambos lados de la hoja, etc.

Los parámetros que controlan como la salida se maneja son:

Spool the data (SPOOL)

Device (DEV)

Spooled output queue (OUTQ)

Si no se especifica otro printer file, se usa el predeterminado del sistema, QSYSPRT.

El parámetro *Spool the data* significa que la salida de un programa (**Spooled File**) va a ser enviado a una **Cola de Salida (Output Queue)**, que es un objeto que sirve de contenedor para todo tipo de listados (intermediario entre la impresora y el printer file). Si el parámetro está en *YES (default) la salida va a ir a la OUTQ especificada en el parámetro OUTQ. En el printer file QSYSPRT el default es *JOB, lo cual indica que la salida va a ir a la OUTQ del job actual. (se verá más adelante)

Cuando el parámetro está en *NO, la salida se envía directamente a la impresora. El sistema se fija en el parámetro DEV para saber a qué impresora mandar el output.

3.7 Save Files

Los **Save Files** son archivos usados para almacenar otros objetos dentro (es análogo al archivo ZIP en Windows).

Para crear un save file se usa el comando **CRTSAVF** (*Create Save File*):

```
Create Save File (CRTSAVF)

Type choices, press Enter.

Save file . . . . . test      Name
Library . . . . . templibs   Name, *CURLIB
Text 'description' . . . . . *BLANK
```

El comando va a pedir el nombre a ponerle al save file y la biblioteca donde alojarlo. Esto va a crear un save file vacío, o sea, sin nada adentro. Para salvar objetos en el save file, se pueden usar los comandos SAVOBJ, SAVLIB, o simplemente SAV. El comando **SAVOBJ** (Save Object), va a salvar los objetos que le especifiquemos en el parámetro *Objects* (*ALL), en el parámetro *Library* se debe poner la biblioteca que contiene dichos objetos (LPPGM431), en el parámetro *Device* se debe poner *SAVF, y en *Save File* el nombre del save file donde guardar los objetos.



```
C5N - Goldcode
File Edit View Communication Actions Window Help

Save Object (SAVOBJ)

Type choices, press Enter.

Objects . . . . . > *ALL      Name, generic*, *ALL
      + for more values
Library . . . . . > LPPGM431  Name, generic*
      + for more values
Device . . . . . > *SAVF     Name, *SAVF, *MEDDFN
      + for more values
Object types . . . . . *ALL    *ALL, *ALRTBL, *BNDDIR...
      + for more values
Save file . . . . . test      Name
Library . . . . . templibps_  Name, *LIBL, *CURLIB

Additional Parameters

Target release . . . . . *CURRENT *CURRENT, *PRV, V5R1M0...
Update history . . . . . *YES      *YES, *NO
Clear . . . . . *NONE            *NONE, *ALL, *AFTER, *REPLACE
                                   More...

F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

MA a 14/048
J902 - Session successfully started
```

El comando **SAVLIB** (*Save Library*) va a permitir salvar toda una biblioteca entera en un mismo save file. No se pueden guardar más de una biblioteca en un mismo save file. El comando **SAV** (*Save Object*) es genérico, y permite grabar objetos del Sistema de Archivos Integrado (IFS). (se verán en el punto 3.9) Para visualizar los contenidos de un save file, se usa el comando **DSPSAVF** (*Display Save File*):



```
C5N - Goldcode
File Edit View Communication Actions Window Help

Display Saved Objects - Save File

Library saved . . . : TSLIBLLM      Release level . . . : V5R3M0
ASP . . . . . : 1                Data compressed . . : No
Save file . . . . : ACPATHS       Objects displayed . . : 2
Library . . . . . : TEMPLIBPS     Objects saved . . . : 2
Records . . . . . : 1232          Access paths . . . . : 0
Save command . . . : SAVOBJ
Save active . . . . : *SYSDFN
Save date/time . . : 06/30/05  10:31:11

Type options, press Enter.
5=Display saved data base file members

Opt Object          Type      Attribute  Owner      Size (K)  Data
---
  5 FIXACCPGMS       *FILE    SAVF      ULRICMD    532      YES
  6 FIXACCFILE       *FILE    SAVF      MILLELJ    144      YES

F3=Exit      F12=Cancel

MA a 01/001
I902 - Session successfully started
```

Para restaurar los objetos dentro de los save files a una biblioteca se usan los comandos **RSTOBJ** (*Restore Object*), **RSTLIB** (*Restore Library*), o **RST** (*Restore Object*). Por ejemplo, para restaurar los objetos con el comando RSTOBJ se debe hacer lo siguiente:

```
Restore Object (RSTOBJ)

Type choices, press Enter.

Objects . . . . . > *ALL          Name, generic*, *ALL
+ for more values
Saved library . . . . . > ACPATHS   Name, generic*, *ANY
+ for more values
Device . . . . . > *SAVF          Name, *SAVF, *MEDDFN
+ for more values
Object types . . . . . > *ALL      *ALL, *ALRTBL, *BNDDIR...
+ for more values
Save file . . . . . > ACPATHS      Name
Library . . . . . > TEMPLIBPS     Name, *LIBL, *CURLIB
Restore to library . . . . . > TEMPLIBPS Name, *SAVLIB
```

Todos los comandos se manejan de manera similar. Con F4 se pueden promptear para configurar el resto de las opciones.



Otros comandos usados para trabajar con save files son:

CLRSVAF (*Clear Save File*) – Limpia los contenidos

DLTF (*Delete File*) – Borrar el archivo

Etc.

3.8 El Sistema de Archivos Integrado (IFS)

El **Sistema de Archivos Integrado (IFS – Integrated File System)** es una parte del OS/400 que soporta E/S de *Stream Files* (cadena de ceros y unos contiguos) y almacenamiento de manera similar a las computadoras personales y sistemas operativos UNIX.

El IFS está compuesto de 11 File Systems, cada uno con su propio conjunto de estructuras lógicas y reglas para interactuar con lo que se almacena. Además de esto, el usuario puede crear sus propios file systems.

Las características principales del IFS son:

- Soporte para guardar información en Stream Files que contienen cadenas continuas de datos. Estas cadenas pueden ser, por ejemplo el texto de un documento o los elementos de una foto.
- Una estructura jerárquica de directorios que permite la organización de los objetos en forma de árbol. Con el *Path* se accede a los objetos.
- Una interfaz común que permite acceder no sólo stream files sino archivos de base de datos, documentos, y otros objetos en el AS/400.

Los sistemas de archivos en el OS/400 son:

1. "root" (/)
2. Open systems file system (*QOpenSys*)
3. User-defined file system (*UDFS*)
4. Library file system (*QSYS.LIB*)
5. Independent ASP QSYS.LIB
6. Document library services file system (*QDLS*)
7. Optical file system (*QOPT*)
8. NetWare file system (*QNetWare*)
9. iSeries NetClient file system (*QNTC*)
10. OS/400 file server file system (*QfileSvr.400*)
11. Network file system (*NFS*)

El más usado en el proyecto es el *root (/)* file system. Este sistema de archivos tiene las características de los file systems DOS y OS/2.

Tiene las siguientes características:

- Optimizado para E/S de Stream Files.
- Soporta múltiples *hard links* y *symbolic links*.
- Soporta sockets locales.
- Soporta APIs.
- Soporta objetos *FIFO.
- Soporta que se journalicen los cambios en los objetos.



Usando el *root (/)* file system

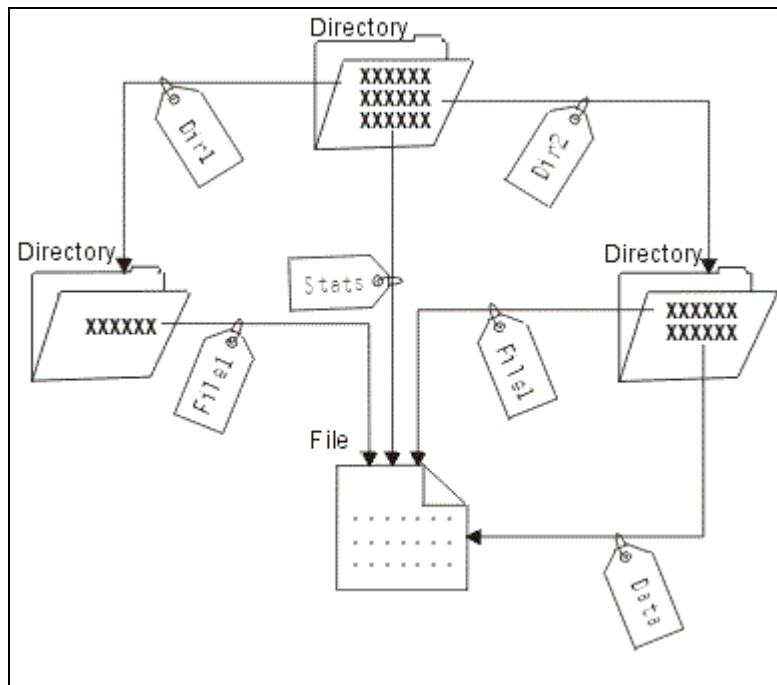
Los nombres de paths para el root file system tienen la siguiente estructura:

Directorio/Directorio . . . /Objeto

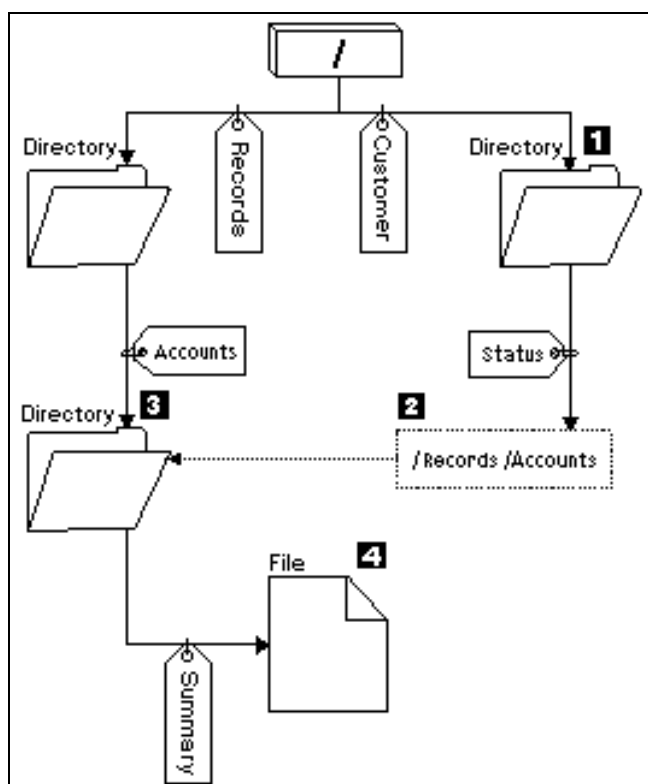
Cada componente del *path name* puede tener hasta 255 caracteres, y el *full path name* (path completo) puede tener hasta 16 MB.

No hay límites para la profundidad de la jerarquía de directorios, más que los límites de espacio en los servidores.

Múltiples *hard links* (referencia directa al objeto) al mismo objeto son permitidos en este file system. Los *Symbolic Links* son soportados totalmente. Un *Symbolic Link* (referencia a través de un path name al objeto, que debe resolverse para accederlo) puede ser usado para vincular objetos de otros file systems (como QSYS.LIB o QDLS) desde el root file system. (ver esquemas abajo)



- Esquema de Hard Links -



- Esquema de Symbolic Links -

4. SUBSISTEMAS Y TRABAJOS

4.1 Concepto de Subsistema

Un **Subsistema** es un entorno operativo lógico mediante el cual el servidor coordina el trabajo y el uso de recursos. El servidor puede tener varios subsistemas, operando independientemente uno de otro. Manejan recursos, y sus características de ejecución están definidas en un objeto llamado **Descripción de Subsistema (Subsystem Description - *SBSD)**.

Todos los trabajos corren dentro de los subsistemas, el concepto de trabajo se verá a continuación.



4.2 Descripción de un Subsistema

Una **Descripción de Subsistema** define cómo, dónde y cuánto trabajo entra en un subsistema y qué recursos utiliza el subsistema para hacer el trabajo.

La descripción de un subsistema está compuesta por entradas (*entries*) de diferentes tipos. Para ver esto mejor, se puede ejecutar el comando **DSPSBSD** (*Display Subsystem Description*). Por ejemplo, si se tipea: **DSPSBSD QBATCH**

Lo que se verá es lo siguiente:

```
CSN - Goldcode
File Edit View Communication Actions Window Help
Display Subsystem Description
Subsystem description: QBATCH      Library: QSYS      System: CATS5N
Status: ACTIVE

Select one of the following:

1. Operational attributes
2. Pool definitions
3. Autostart job entries
4. Work station name entries
5. Work station type entries
6. Job queue entries
7. Routing entries
8. Communications entries
9. Remote location name entries
10. Prestart job entries

More...

Selection or command
===>

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel

MA a 21/007
1992 - Session successfully started
```

En *1. Operation attributes*, se ven los atributos de ejecución del mismo:



```
CSN Goldcode
File Edit View Communication Actions Window Help
Display Operational Attributes
System: CATS5N
Subsystem description: QBATCH Status: ACTIVE
Subsystem description . . . . . : QBATCH
Library . . . . . : QSYS
Maximum jobs in subsystem . . . . . : *NOMAX
Sign-on display file . . . . . : QDSIGNON
Library . . . . . : QSYS
System library list entry . . . . . : *NONE
Press Enter to continue.
F3=Exit F12=Cancel
MA a 01/001
1902 - Session successfully started
```

En 2. *Pool Definitions*, se ve el pool de memoria donde los trabajos van a correr.

```
Display Pool Definitions
System: CATS5N
Subsystem description: QBATCH Status: ACTIVE
Pool ID Storage Size (K) Activity Level
1 *BASE
```

La tercera entrada, 3. *Autostart Job Entries*, indica los jobs que se van a ejecutar automáticamente cuando el subsistema se inicie. Generalmente estos jobs hacen trabajos de inicialización asociados con el subsistema. (sería análogo a los accesos directos que se colocan en el “startup” o “inicio” de Windows)

La cuarta y quinta entrada, 4. *Work station name entries* y 5. *Work station type entries* indican qué nombres o tipos de estaciones de trabajo (nombre de la sesión de pantalla) correrán dentro de ese subsistema. Por ejemplo, si se agregan entradas con el nombre TEST*, todas las sesiones de pantalla que se llamen así (TEST1, TEST2, etc.) pasarán inmediatamente a correr dentro de ese subsistema.

La sexta entrada, 6. *Job queue entries* (entradas de colas de trabajos), indica qué colas de trabajos están asociadas al subsistema.



Seq	Job	Queue	Library	Max	Active	1	2	3	4	5	6	7	8	9
10	QBATCH	QGPL	QGPL	1	*	*	*	*	*	*	*	*	*	*
11	VOL1	DEPLOYMENT	QGPL	1	*	*	*	*	*	*	*	*	*	*
12	VOL2	DEPLOYMENT	QGPL	1	*	*	*	*	*	*	*	*	*	*
13	VOL3	DEPLOYMENT	QGPL	1	*	*	*	*	*	*	*	*	*	*
14	VOL5	DEPLOYMENT	QGPL	1	*	*	*	*	*	*	*	*	*	*
20	QS38EVOKE	QGPL	QGPL	*NOMAX	*	*	*	*	*	*	*	*	*	*
30	XJQROBOT	QGPL	QGPL	*NOMAX	*	*	*	*	*	*	*	*	*	*
35	XJQUPDAT	QGPL	QGPL	1	*	*	*	*	*	*	*	*	*	*
39	ACMSJOBQ	ACMSCTL	QGPL	*NOMAX	*	*	*	*	*	*	*	*	*	*
40	XJQREADO	QGPL	QGPL	1	*	*	*	*	*	*	*	*	*	*
50	QTXSRCH	QGPL	QGPL	*NOMAX	*	*	*	*	*	*	*	*	*	*
70	XJQIMED	QGPL	QGPL	1	*	*	*	*	*	*	*	*	*	*
75	XJQPART	QGPL	QGPL	1	*	*	*	*	*	*	*	*	*	*

Una **Cola de Trabajos (Job Queue - *JOBQ)** es una lista ordenada de trabajos esperando a ser procesados por el subsistema al cual está asociada. Los trabajos no van a ser seleccionados para ser ejecutados desde la cola de trabajos a menos que el subsistema esté activo y la cola de trabajo no esté en status HLD (Hold). Las colas de trabajos se pueden usar para controlar el orden en el cual los trabajos son ejecutados.

IBM provee ciertas colas de trabajos predeterminadas, y también se pueden crear las propias.

La séptima entrada, *7. Routing entries*, se explicará más adelante. (punto 4.9)

La octava y novena entrada, *8. Communications entries* y *9. Remote location name entries* identifican los dispositivos de comunicaciones que otro sistema usa para someter el trabajo tanto por tipo de dispositivo (8) o por nombre (9).

La décima y última entrada, *10. Prestart job entries*, indican los trabajos de prearranque que se ejecutarán. Los *Prestart jobs* mejoran el rendimiento cuando se inicia una conexión con un servidor. Un *trabajo de prearranque* es un trabajo de proceso por lotes que se inicia antes de que un programa de un sistema remoto inicie las comunicaciones con el servidor. Los trabajos de prearranque utilizan las correspondientes entradas existentes en la descripción del subsistema para determinar cuál es el programa, la clase y la agrupación de almacenamiento que deben utilizar cuando se arrancan los trabajos. Dentro de una entrada de trabajo de prearranque se deben especificar los atributos que el subsistema va a utilizar para crear y gestionar una agrupación de trabajos de prearranque. La entrada de trabajo de prearranque se puede definir de tal forma que el trabajo se arranque automáticamente cuando se arranque el subsistema.



4.3 Subsistemas QCTL, QBATCH y QINTER

Los principales subsistemas en el AS/400 son los denominados QBATCH, QCTL y QINTER. Cada uno tiene un propósito particular y está configurado para el mismo.

El subsistema **QCTL** es conocido como el subsistema de control (*controlling subsystem*). Inicia automáticamente cuando inicia el sistema, y es el subsistema mediante el cual el operador del sistema controla el mismo. Se especifica en el valor de sistema QCTLSBSD (un valor de sistema es un valor que aplica para todo el sistema en general). IBM provee dos completas descripciones de subsistema de control: QBASE (predeterminado) y QCTL. Sólo uno puede estar activo al mismo tiempo.

Si el subsistema de control es finalizado, pasa a un estado restringido. Si todos los subsistemas, incluido el de control, son finalizados, el sistema pasa a un *estado restringido*. Cuando el sistema está en este estado, la mayoría de la actividad del sistema está finalizada, y sólo una workstation está activa. El sistema tiene que estar en este estado para comandos tales como SAVSYS (*Save System*) o RCLSTG (*Reclaim Storage*). Para salir de este estado, se debe iniciar el subsistema de control de nuevo.

El subsistema **QBATCH** básicamente sirve para trabajos que corren en batch, es decir, trabajos que no requieren interactividad con el usuario, o sea, que ejecutarán independientemente de que el usuario participe o no.

El subsistema **QINTER** sirve básicamente a sesiones interactivas para los usuarios (sesiones de pantalla), permite que los usuarios ejecuten comandos, y trabajos por ejemplo.

4.4 El concepto de Trabajo

Todo proceso en el AS/400 es un **Trabajo (Job)**. Cada trabajo tiene sus propios atributos, y se los identifica unívocamente dentro del sistema. Todos los jobs, excepto los del sistema, corren dentro de subsistemas. Un job puede entrar a un subsistema de varias maneras, ya sea a través de las job queue entries, work station entries, communications entries, autostart job entries o prestart job entries.

Un job es una colección de uno o más *Hilos (Threads)*. Cada job tiene al menos un hilo, el cual se conoce como el hilo inicial. Se crea cuando empieza el job.

El hilo es una unidad independiente de trabajo. Cada hilo tiene su propio entorno de ejecución, como una *Pila de Llamadas (Call Stack)*, pero el hilo comparte muchos de los recursos asignados al job. El identificador del hilo es único en el job al cual pertenece.

El trabajo en el AS/400 se controla principalmente a nivel de jobs.

Los jobs pueden estar *Activos* o *Inactivos*. Los jobs activos son jobs que empezaron a ejecutarse pero que no terminaron de hacerlo. Algunas características de los jobs activos son:

- Contiene un código de ejecución
- Tiene una pila de llamadas
- Tiene objetos lockeados (*locked*)
- Tiene el status de un job activo, por ejemplo: *Running*, *Waiting for...*



Para ver los jobs activos, se debe ejecutar el comando WRKACTJOB (*Work with Active Jobs*), el cual mostrará la siguiente pantalla:

```
CSN Goldcode
File Edit View Communication Actions Window Help

Work with Active Jobs CATS5N
12/13/05 12:36:27

CPU %: 4.7 Elapsed time: 00:00:06 Active jobs: 316

Type options, press Enter.
2=Change 3=Hold 4=End 5=Work with 6=Release 7=Display message
8=Work with spooled files 13=Disconnect ...

Opt Subsystem/Job User Type CPU % Function Status
--
QPADEV001P DEVITLX INT 2.4 CMD-WRKACTJOB RUN
QPADEV0018 BACZARM INT .5 PGM-LOGUSER DSPW
ZPSE QSYS SBS .0 DEQW
TCPIPL0C QGATE BCH .0 EVTW
SCPF QSYS SYS .0 EVTW
SBM431DDLY PAMARWA BCH .0 DLY-600 DLYW
RMTW000043 QSPLJOB WTR .0 EVTW
RMTW000042 QSPLJOB WTR .0 EVTW
RMTW000041 QSPLJOB WTR .0 EVTW

More...

Parameters or command
===>
F3=Exit F5=Refresh F7=Find F10=Restart statistics
F11=Display elapsed data F12=Cancel F23=More options F24=More keys

MA a 10/002
1902 - Session successfully started
```

Se pueden ver los jobs activos, cuanta CPU consumen, en qué estado están, etc. Se puede ordenar por cualquier campo (ej. CPU %) presionando F16 sobre el campo.

Si al comando WRKACTJOB se le presiona F10 para que muestre todos los parámetros, uno de los tantos que mostrará es el de *Subsystem*, con el cual se verán los trabajos activos sólo de ese subsistema. Por ejemplo: **WRKACTJOB SBS(QBATCH)**

Los jobs inactivos son jobs en una job queue esperando a ser ejecutados, o jobs que ya terminaron de procesar pero que están esperando a imprimir en el spool.

Tipos de trabajos

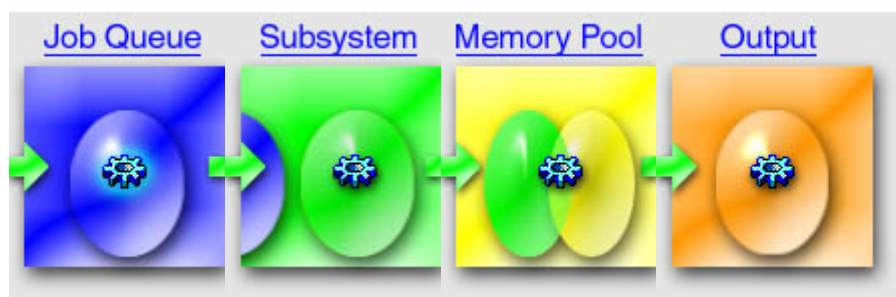
Hay varios tipos de trabajos en el sistema:

- Autostart: jobs que se inician automáticamente cuando el subsistema al cual están asociados arranca.
- Batch: Un job Batch es un grupo predefinido de acciones de procesamiento que se someten al sistema.
- Comunicaciones: Un job de comunicaciones es un job batch que fue iniciado por un programa desde un sistema remoto.
- Interactivo: Un job interactivo requiere input desde un usuario logueado.
- Prestart: Un job prestart es un job batch que empieza antes de que un requerimiento de trabajo se reciba.

- Reader and Writer: Un "Reader" job es un job que lee datos desde diskette o archivos de base de datos y los coloca en una job queue, y un "Writer" job es un job que escribe desde un spool a una impresora.
- Subsistema: Un job de subsistema provee control sobre un subsistema activo.
- Sistema: Los jobs de sistema son creados por el sistema operativo.
- Server: Los jobs de servidor son jobs que corren continuamente en background esperando por trabajo. Estos jobs corren generalmente en subsistemas que vienen con el sistema: QSYSWRK, QSERVER o QUSRWRK. Los jobs de servidor están comúnmente asociados con funciones tales como HTTP, Lotus Notes y TCP/IP.

4.5 El ciclo de vida de un Trabajo

El ciclo de vida de un job comienza cuando se somete (*Submit*), y después sigue los siguientes pasos:



1. Someter el Trabajo: (*Submit Job*) Cuando un job es sometido (o "submitido"), se crea y entra al sistema. En este momento el job recibe sus propiedades, que las puede obtener de su *Job Description*.
2. El job ingresa a la Job Queue: Cuando el job entra a la *Job Queue*, espera a ser tomado por el subsistema al cual la job queue está asociada. (*Job queue entries*)

```

Work with Job Queue

Queue:  XJQIMEDD2P      Library:  QGPL      Status:  RLS/SBS

Type options, press Enter.
  2=Change  3=Hold  4=End  5=Work with  6=Release

Opt   Job              User              Number      Priority    Status
=     PCJN0900A        QPMA              239615      3          HLD
-     WPJN5300S        QEPR              136453      5          HLD
-     WPJN5305A        QEPR              136454      5          HLD

```

Si el status de la *Job Queue* es *RLS/SBS* significa que está activa, o sea, los jobs van a ir siendo tomados por el subsistema asociado.



3. El job ingresa al subsistema: Cuando el job entra al subsistema está Activo. Hasta que no obtenga su nivel de actividad y pool de memoria, no puede empezar a correr. El hecho de que el job empiece a correr dependerá de varios factores, como ser la cantidad máxima de trabajos que puede ejecutar el subsistema, o la cantidad de hilos que ese pool de memoria tenga corriendo.
4. El subsistema usa memoria de un pool de memoria para correr el job: La memoria es un recurso que el subsistema usa para ejecutar el job. La cantidad de memoria de un pool de memoria, así como la cantidad de jobs que compiten por la memoria, afectan la ejecución del job. El subsistema usan diferentes pools de memoria para correr los jobs dentro de ellos.
5. El job termina y pasa a la output queue: La salida a printer de un job (también llamados *archivos de spool – Spooled Files*) es enviada a una cola de salida (*Output Queue*) donde espera a ser enviado a una impresora o archivo.

5. SEGURIDAD

En el AS/400 cada objeto tiene ciertas autorizaciones que indican los permisos de acceso, quienes pueden accederlo, quienes cambiarlo, quienes borrarlo, etc.

5.1. Permisos sobre objetos

Hay diversos tipos de permisos sobre un objeto en el AS/400. Puede ser accedido a través de un usuario, a través de un grupo de usuarios, o a través de lo que se conoce como *Lista de Autorizaciones (Authorization List)*. También hay permisos sobre el objeto, y permisos sobre los datos.

Permisos sobre el objeto

Son las autoridades que tiene el usuario específicamente sobre el objeto.

Object Operational Authority – Provee la autoridad para ver los atributos del objeto y usarlo como se especifica en las autorizaciones sobre los datos que tiene el usuario.

Object Management Authority – Provee autoridad a seguridad específica, para mover o renombrar el objeto, y agregar miembros si es un archivo de base de datos.

Object Existence Authority – Provee la autoridad para controlar la existencia y propiedad (*ownership*) del objeto.

Object Alter Authority – Provee la autoridad para cambiar los atributos del objeto.

Object Reference Authority – Provee la autoridad para especificar el objeto como primer nivel en una *referential constraint*.

Permisos sobre los datos

Las autoridades específicas que tiene el usuario sobre los datos contenidos en el objeto.

Read – Provee la autoridad para acceder a los contenidos del objeto.

Add – Provee la autoridad para agregar entradas al objeto.

Update – Provee la autoridad para actualizar el contenido de las entradas existentes del objeto.

Delete – Provee la autoridad para remover entradas del objeto.

Execute – Provee la autoridad para ejecutar un programa o buscar una biblioteca o directorio.



Para ver los permisos que tiene determinado objeto, podemos usar el comando WRKOBJ *Biblioteca/Objeto*, la pantalla que se vería es la siguiente:

```
Display Object Authority

Object . . . . . : CSPENVTO      Owner . . . . . : XUPCDIS
Library . . . . . : LPDBS        Primary group . . . : *NONE
Object type . . . . : *FILE       ASP device . . . . . : *SYSBAS

Object secured by authorization list . . . . . : *NONE

User      Group      Object
*PUBLIC    *PUBLIC    Authority
XUPCDIS    XUPCDIS    *CHANGE
                *ALL

Press Enter to continue.

F3=Exit  F11=Display detail object authorities  F12=Cancel  F17=Top
F18=Bottom
(C) COPYRIGHT IBM CORP. 1980, 2003.

MF a 01/001
```

En esta pantalla se ven los diversos tipos de accesos que tiene el objeto. Lo que muestra esta pantalla es lo siguiente:

Owner – Muestra el usuario que actualmente es dueño del objeto. En este caso, es XUPCDIS.
Object secured by authorization list – Indica si el objeto está asegurado a través de una lista de autorizaciones (se verá después). En este caso, no hay ninguna (*NONE).
User – Nombre de los usuarios que están autorizados a usar el objeto. Puede tener los siguientes valores:

- *PUBLIC – El valor público define las autoridades para todos los usuarios que no están específicamente definidos y que no son miembros de la lista de autorizaciones.
- *GROUP – Define las autoridades para el perfil de usuario perteneciente al grupo que aparece en la columna *Group*.
- *ADOPTED – Define las autoridades del objeto que se toman del job que está corriendo actualmente. Si este campo es mostrado, este job tiene un programa corriendo que adopta el perfil de usuario del dueño del objeto.

Group – El nombre del perfil de grupo desde el cual el objeto recibe autoridad. Un nombre de perfil de grupo es mostrado cuando el valor en la columna *USER es *GROUP.

Object Authority – La autoridad que el usuario tiene al objeto. Hay varias autoridades definidas por el sistema para los objetos:

- *ALL – Permite todas las operaciones sobre el objeto, excepto aquellas limitadas al dueño o controladas por la lista de autorizaciones.
- *CHANGE – Permite todas las operaciones sobre el objeto, excepto aquellas limitadas al dueño o controladas por *object existence authority*, *object alter authority*, *object reference authority*, y *object management authority*.
- *EXCLUDE – Todas las operaciones sobre el objeto están prohibidas.



**USE* – Permite el acceso a los atributos del objeto y a su uso. El usuario no puede cambiar el objeto.

USER DEF – Lo muestra el sistema cuando los permisos sobre el objeto y sobre los datos no se corresponden ninguno de los tipos especificados arriba.

**AUTL* – Se puede usar cuando se especifica *PUBLIC. Indica que va a tomar las autorizaciones públicas que están especificadas en la lista de autorizaciones.

Si se presiona F11 en la pantalla anterior, se pueden ver los permisos sobre el objeto y sobre los datos, según las categorías especificadas arriba:

```
C5N - Goldcode
File Edit View Communication Actions Window Help

Edit Object Authority

Object . . . . . : CSPENVTO      Owner . . . . . : XUPCDIS
Library . . . . . : LPDBS        Primary group . . . : *NONE
Object type . . . . : *FILE       ASP device . . . . : *SYSBAS

Type changes to current authorities, press Enter.

Object secured by authorization list . . . . . : *NONE

User      Group      Object Authority  Opr  Mgt  Exist  Alter  Ref
*PUBLIC   *PUBLIC   *CHANGE  X    -    -    -    -
XUPCDIS   *PUBLIC   *ALL     X    X    X    X    X

Bottom

F3=Exit  F5=Refresh  F6=Add new users  F10=Grant with reference object
F11=Display data authorities  F12=Cancel  F17=Top  F18=Bottom

MA  a  09/068
I902 - Session successfully started
```




```
C5N - Goldcode
File Edit View Communication Actions Window Help
[Icons]

Edit Object Authority

Object . . . . . : CSPENVTO      Owner . . . . . : XUPCDIS
Library . . . . . : LPDBS        Primary group . . . : *NONE
Object type . . . . : *FILE       ASP device . . . . . : *SYSBAS

Type changes to current authorities, press Enter.

Object secured by authorization list . . . . . : *NONE

User      Group      Object Authority  Read  Add  Update  Delete  Execute
*PUBLIC    *CHANGE  X      X      X      X      X
XUPCDIS    *ALL     X      X      X      X      X

Bottom

F3=Exit  F5=Refresh  F6=Add new users  F10=Grant with reference object
F11=Nondisplay detail  F12=Cancel  F17=Top  F18=Bottom

MA a 09/068
[Icons] [902 - Session successfully started]
```

5.2. Lista de autorizaciones

El hecho de proveer para cada usuario los accesos a cada objeto sobre el cual necesitan trabajar puede necesitar mucho tiempo y esfuerzo duplicado, ya que en la mayoría de los casos, muchos usuarios necesitan acceder al mismo grupo de objetos.

Una forma mucho más fácil de dar estos accesos es a través de una **Lista de Autorizaciones (Authorization List)**, la cual consiste en una lista de usuarios o grupo de usuarios, el tipo de autoridad (USE, CHANGE, y EXCLUDE) para cada usuario o grupo de usuario, y una lista de objetos a los cuales ésta lista de autorizaciones les provee el acceso.

En el AS/400 las listas de autorizaciones están implementadas como objetos de tipo *AUTL.

5.3. Perfiles de usuario

El **Perfil de Usuario (User Profile)** es un objeto con un nombre único que contiene una clave, la lista de autoridades especiales asignados al usuario, y los objetos de los cuales este usuario es dueño.

Los siguientes parámetros del perfil de usuario proveen la información que determina dónde el output va a ir:

- Job Description (JOBDE)
- Output queue (OUTQ)
- Printer Device (PRTDEV)



Los atributos más importantes son:

Previous sign-on – Indica la última fecha en la cual el usuario se logueó en el sistema.

Status – Indica si el usuario está habilitado o deshabilitado (*ENABLED o *DISABLED)

User class – El tipo de usuario. Puede ser:

- *PGMR - Programador
- *SECADM - Administrador de seguridad
- *SECOFR - Oficial de seguridad
- *SYSOPR - Operador del sistema
- *USER - Usuario

Special Authority – Indica las autoridades especiales que le fueron dadas al usuario. Puede ser una de las siguientes:

- *NONE - Sin autoridades especiales
- *ALLOBJ - Autorización sobre todos los objetos
- *AUDIT - Autoridad para auditoria
- *IOSYSCFG - Autoridad para configurar el sistema
- *JOBCTL - Autoridad para controlar jobs
- *SAVSYS - Autoridad para salvar el sistema
- *SECADM - Autoridad para administrar la seguridad
- *SERVICE - Autoridad para servicio
- *SPLCTL - Autoridad para controlar Spools

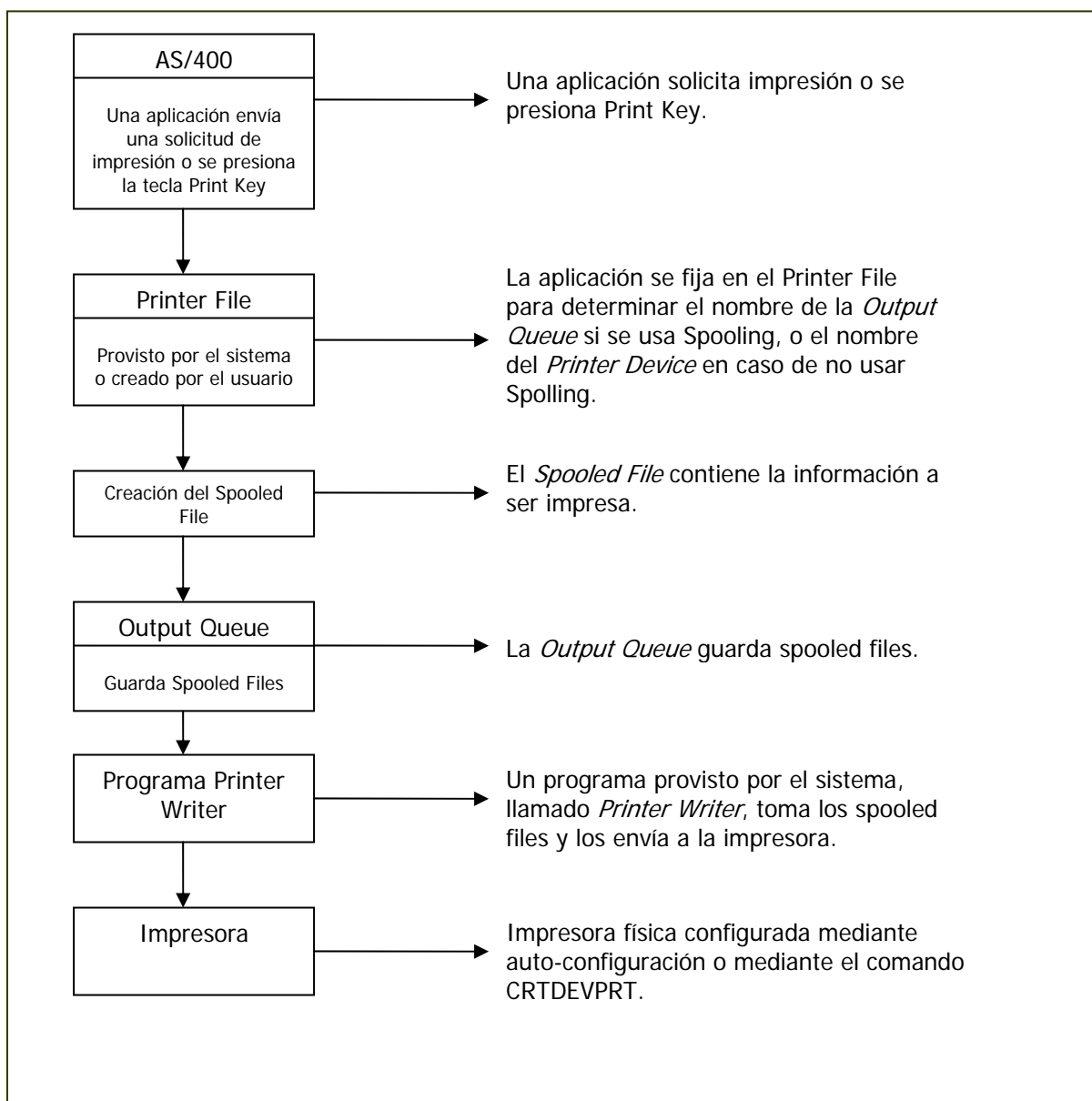
Group Profile – Nombre del primer grupo al cual pertenece el perfil de usuario. Grupos adicionales pueden especificarse en el campo *Supplemental Groups*.

6. IMPRESIÓN

Para entender cómo se maneja la impresión en el AS/400 primero hay que entender ciertos conceptos.

6.1. Conceptos Básicos

Al imprimir a través del AS/400 se lleva a cabo el siguiente proceso:



Cuando una aplicación envía una solicitud de impresión, o se presiona la tecla Print Key, se inicia el proceso detallado arriba.



Al iniciar el proceso de impresión, un *Printer File* es accedido, un *Spooled File* es generado y ese spooled file es enviado a una cola de salida (*Output Queue*). Si esta cola de salida tiene asociada una impresora, el programa *Printer Writer* cuando detecta que un spooled file llega a la misma lo toma y lo envía a la impresora física.

6.2. Archivos de Impresora

Las impresoras son soportadas en el AS/400 a través de **Archivos de Impresora** (*Printer Files*). Estos archivos describen cómo se van a manejar los datos que se pasan de la aplicación a la impresora. Hay Printer Files provistos por el sistema, y se pueden crear printer files de usuario, a través del comando CRTPRTF.

Un Printer File maneja cada solicitud de impresión. Tiene parámetros que indican cómo la salida debe ser formateada, que fuentes usar, etc.

Los parámetros que controlan cómo se maneja la salida son los siguientes:

- Spool the data (SPOOL)
- Device (DEV)
- Spooled output queue (OUTQ)
- Spooled File Owner (SPLFOWN)

6.3. Colas de Salida

Como se vio, el *Printer File* le indica al *Spooled File* donde ir. En caso de que el parámetro SPOOL sea *YES, el listado irá a la **Cola de Salida** (*Output Queue*) especificada.

Esta cola de salida sirve como repositorio, el cual almacena los *Spooled Files* que le son enviados, y los redirige a la impresora si es que hay una asociada.

Para saber si una *Output Queue* tiene una impresora física asociada, se tipea el comando WRKOUTQ <Nombre de la Output Queue>, y nos fijamos en el *Status* de la misma.

6.4. Printer Writer

Cuando se envía una impresión a una *Output Queue*, y la misma tiene asociada una Printer, entonces el job conocido como **Printer Writer** entra en funcionamiento.

La funcionalidad de este job es enviar los Spooled Files que están en la Output Queue especificada a la impresora física asociada a dicha Output Queue.

6.5. Colas de Salida Remotas

Hay cierto tipo especial de colas de salida, llamadas **Colas de Salida Remotas** (*Remote Output Queues*), las cuales se caracterizan por ser colas de salida con IP. O sea, las impresiones van directamente a la impresora física, sin necesidad de pasar por un DEVD.



7. MENSAJES

Los **Mensajes** (*Messages*) son comunicaciones enviadas desde un usuario, programa o procedimiento a otro.

7.1 Tipos de Mensajes

En el OS/400 existen dos tipos de mensajes:

- *Mensajes Inmediatos*, que son creados por el programa o usuario al ser enviados y que no se almacenan permanentemente en el sistema.
- *Mensajes Predefinidos*, los cuales están creados antes de ser usados. Están almacenados en archivos conocidos como *Message Files* y desde ahí se obtiene su descripción.

7.2 Descripciones de Mensajes

Los mensajes predefinidos, tienen una *Descripción de Mensaje* (*Message Description*) la cual le da el contenido al mensaje.

Para ver la descripción de un mensaje, se debe tipear el siguiente comando: *DSPMSGD* (*Display Message Description*)

7.3 Archivos de Mensajes

Las descripciones de cada mensaje se almacenan en un *Archivo de Mensajes* (*Message File*), que es un objeto del tipo *MSGF.

7.4 Colas de Mensajes

Todos los mensajes en el sistema son enviados a una **Cola de Mensajes** (*Message Queue*). Los mensajes que son enviados en respuesta a una solicitud directa, como un comando, son automáticamente mostrados en la pantalla desde la cual se hizo la solicitud. Para todos los demás mensajes, el usuario, programa o procedimiento debe recibir el mensaje desde la cola de mensajes para mostrarlo.

Una cola de mensajes es provista para cada usuario y para cada sesión interactiva (*Display Station*). También hay colas de mensajes para cada trabajo corriendo en el sistema.

También se proveen colas de mensajes para el log histórico (*QHST – History Log*) y para el operador del sistema (*QSYSOPR*).

Además de estas colas de mensajes, se pueden crear colas de mensajes propias.



APENDICE

A.1 Links

IBM Information Center V5R4

<http://publib.boulder.ibm.com/infocenter/iseries/v5r4/index.jsp?lang=en>

IBM iSeries Support - Authorized Problem Analysis Reports (APARs)

http://www-912.ibm.com/n_dir/nas4apar.nsf/nas4aparhome

A.2 Document Change Log

Date	Versión	Author	Description
10-Julio-2006	1.2	Leonardo De Vita	Add de APYJRNCHG
10-Julio-2006	1.1	Leonardo De Vita	Fix en Ejercicios 5.5
15-Junio-2006	1.0	Leonardo De Vita	Versión 1.0