

IBM

@server

iSeries

Introducción al sistema de
archivos integrado

Versión 5





@server

iSeries

Introducción al sistema de
archivos integrado

Versión 5

Contenido

Acerca de Introducción al sistema de archivos integrado	vii
¿Quién debe leer el manual Introducción al sistema de archivos integrado?	vii
Información de exención de responsabilidad del código	vii
Capítulo 1. Introducción al sistema de archivos integrado	1
¿Qué es el sistema de archivos integrado?	1
¿Por qué utilizar el sistema de archivos integrado?	1
Capítulo 2. Conceptos del sistema de archivos integrado	3
Archivo continuo	3
Archivos continuos *TYPE1 y *TYPE2	3
Sistemas de archivos en el sistema de archivos integrado	4
Directorio	6
Directorio actual y directorio de inicio.	8
Directorios *TYPE2	9
Utilización de los directorios *TYPE2 en la V5R1 del OS/400.	11
Conversión a un directorio *TYPE2	11
Los sistemas de archivos "Raíz", QOpenSys, o UDFS no están disponibles	12
Requisitos de almacenamiento auxiliar	12
Consideraciones de enlaces simbólicos	12
Agrupaciones de almacenamiento auxiliar (ASP) independientes	13
Consideraciones sobre Guardar/Restaurar	13
Preparación para la conversión a *TYPE2	13
Proceso de conversión	14
Ejemplo: conversión de todos los sistemas de archivos (número pequeño de objetos)	16
Ejemplo: conversión de todos los sistemas de archivos (número elevado de objetos)	16
Ejemplo: conversión de sólo determinadas ASP	17
Nombre de vía de acceso	17
Enlace	18
Enlace fijo	19
Enlace simbólico.	20
Comparación: enlace fijo y enlace simbólico	21
Atributos ampliados.	22
Continuidad de nombres	23
Capítulo 3. Acceda al sistema de archivos integrado utilizando la interfaz del sistema tradicional 25	
Realizar operaciones utilizando los menús y pantallas del iSeries.	25
Realizar operaciones mediante mandatos CL	26
Reglas sobre el nombre de ruta para mandatos CL y pantallas	29
Realizar operaciones mediante un PC	31
Transferencia de archivos utilizando FTP.	32
Trabajar con archivos utilizando el NetServer de iSeries	33
Trasladar objetos a otro sistema de archivos	34
Consideraciones sobre el traslado de objetos a otro sistema de archivos	34
Directorios que proporciona el sistema de archivos integrado	35
Capítulo 4. Acceso al sistema de archivos integrado con el iSeries Navigator	37
Reincorporar un archivo	37
Reservar un archivo	38
Establecer permisos para un archivo o carpeta	38
Configurar la conversión de archivos de texto	38
Enviar un archivo o carpeta a otro sistema	38
Cambiar opciones de la definición del paquete.	39

Planificar fecha y hora en que enviar el archivo o carpeta	39
Crear una carpeta	40
Eliminar una carpeta	40
Crear un compartimiento de archivos	40
Cambiar un compartimiento de archivos	40
Crear un sistema de archivos definido por el usuario	41
Montar un sistema de archivos definido por el usuario	41
Desmontar un sistema de archivos definido por el usuario	42
Iniciar el registro por diario	42
Finalizar registro por diario	42
Capítulo 5. Soporte de programación para el sistema de archivos integrado	45
Copia de datos entre archivos continuos y archivos de base de datos	45
Copiar datos mediante mandatos CL	46
Copiar datos mediante API	47
Copia de datos utilizando la función de transferencia de datos	47
Copia de datos entre archivos continuos y archivos de salvar	49
Realizar operaciones utilizando API	49
Funciones ILE C/400	55
Soporte de archivos voluminosos para las API	55
Reglas de nombres de vía de acceso para las API	56
Descriptor de archivos.	57
Seguridad	58
Soporte de zócalos	59
Soporte internacional y asignación de nombres	59
Conversión de datos	59
Capítulo 6. Sistemas de archivos en el sistema de archivos integrado	61
Comparación de los sistemas de archivos	61
sistema de archivos “raíz” (/)	65
Utilización del sistema de archivos “raíz” (/)	66
Sistema de archivos de sistemas abiertos (QOpenSys)	67
Utilización de QOpenSys.	67
Sistema de archivos definido por el usuario (UDFS).	68
Conceptos de los UDFS	69
Utilización de los UDFS mediante la la interfaz del sistema de archivos integrado.	69
Sistema de archivos de biblioteca (QSYS.LIB).	72
Utilización de QSYS.LIB mediante la interfaz del sistema de archivos integrado	73
QSYS.LIBde ASP independiente	75
Utilización de QSYS.LIB de ASP independiente mediante la interfaz del sistema de archivos integrado.	75
Sistema de archivos de servicios de biblioteca de documentos (QDLS)	78
Utilización de QDLS mediante la interfaz del sistema de archivos integrado	78
Sistema de archivos óptico (QOPT).	80
Utilización de QOPT mediante la interfaz del sistema de archivos integrado	80
Sistema de archivos NetWare (QNetWare)	82
Montar sistemas de archivos NetWare.	83
Estructura de directorios de QNetWare	83
Utilización de QNetWare mediante la la interfaz del sistema de archivos integrado	83
Sistema de archivos de Windows NT Server (QNTC)	85
Utilización de QNTC mediante la la interfaz del sistema de archivos integrado	85
Sistema de archivos del servidor de archivos OS/400 (QFileSvr.400)	88
Utilización de QFileSvr.400 mediante la interfaz del sistema de archivos integrado	88
Sistema de archivos de red (NFS)	91
Utilización de los sistemas de archivos NFS mediante la la interfaz del sistema de archivos integrado.	91

Capítulo 7. Soporte del registro por diario para los objetos del sistema de archivos integrado	95
gestión por diario	95
Objetos que debería registrar por diario	95
Objetos del sistema de archivos integrado registrados por diario	96
Operaciones registradas en diario	97
Consideraciones especiales sobre las entradas de diario	98
Apéndice A. Llamada de procedimiento remoto independiente del transporte	99
Selecciones de red	99
Conversión de nombre a dirección	99
eXternal Data Representation (XDR)	100
Autenticación	101
RPC independiente del transporte (TI-RPC)	102
Interfaces API TI-RPC simplificadas	102
Interfaces API TI-RPC de nivel superior	102
Interfaces API TI-RPC de nivel intermedio	102
Interfaces API TI-RPC de nivel experto	102
Otras interfaces API TI-RPC	103
Apéndice B. Programa de ejemplo con funciones C del sistema de archivos integrado	105
Apéndice C. Ejemplo de código RPG para un sistema de archivos integrado	111
Bibliografía	113
Índice	115

Acerca de Introducción al sistema de archivos integrado

En este manual se ofrece una introducción al sistema de archivos integrado, en la que se explica:

- Qué es el sistema de archivos integrado.
- Por qué puede ser conveniente utilizarlo.
- Conceptos y terminología del sistema de archivos integrado.
- Las interfaces que se pueden utilizar para interactuar con el sistema de archivos integrado.
- Las API y técnicas que se pueden utilizar para crear programas que interactúen con el sistema de archivos integrado.
- Características de los sistemas de archivos individuales.

¿Quién debe leer el manual Introducción al sistema de archivos integrado?

Esta publicación está destinada a usuarios, programadores y administradores de servidores iSeries que deseen conocer el sistema de archivos integrado y cómo utilizarlo.

Información de exención de responsabilidad del código

Este documento contiene ejemplos de programación.

IBM le otorga una licencia de copyright no exclusiva para utilizar todos los ejemplos de código de programación a partir de los cuales puede generar funciones similares que se ajusten a sus necesidades.

IBM ofrece todos los códigos de muestra sólo a modo de ejemplo. Estos ejemplos no se han verificado a fondo bajo todas las condiciones. Por lo tanto, IBM no puede garantizar o implicar la fiabilidad, la operabilidad o el funcionamiento de estos programas.

Todos los programas que aquí se incluyen se ofrecen "TAL CUAL" sin garantías de ningún tipo. Se renuncia explícitamente a las garantías implícitas de no infracción, comerciabilidad y adecuación para una finalidad concreta.

Capítulo 1. Introducción al sistema de archivos integrado

Los temas siguientes describen el sistema de archivos integrado del servidor iSeries y muestran la utilidad que puede tener en el servidor.

¿Qué es el sistema de archivos integrado?

El **sistema de archivos integrado** es un componente de OS/400 que admite la gestión de almacenamiento y entrada/salida continua de forma similar a los sistemas operativos de PC y UNIX, proporcionando una estructura integrada para toda la información almacenada en el servidor.

Las características clave del sistema de archivos integrado son las siguientes:

- Soporte para almacenar la información en archivos continuos que pueden contener largas series continuas de datos. Estas series de datos pueden ser, por ejemplo, el texto de un documento o los elementos de imagen de una imagen. El soporte de archivos continuos se ha diseñado para utilizarlo eficazmente en las aplicaciones cliente/servidor.
- Una estructura de directorios jerárquica que permite organizar los objetos como si de las frutas de las ramas de un árbol se tratara. Especificando la vía de acceso a un objeto a través de los directorios se accede al mismo.
- Una interfaz común que permite a los usuarios y a las aplicaciones acceder no sólo a los archivos continuos, sino también a archivos de bases de datos, documentos y otros objetos almacenados en el servidor.
- Una vista común de los archivos continuos almacenados localmente en el servidor, en el Integrated xSeries Server for iSeries, o en un servidor Windows NT remoto. Los archivos continuos pueden estar almacenados también de manera remota en un servidor de red de área local (LAN), un servidor Novell NetWare, otro servidor iSeries remoto o un servidor del sistema de archivos de red (NFS).

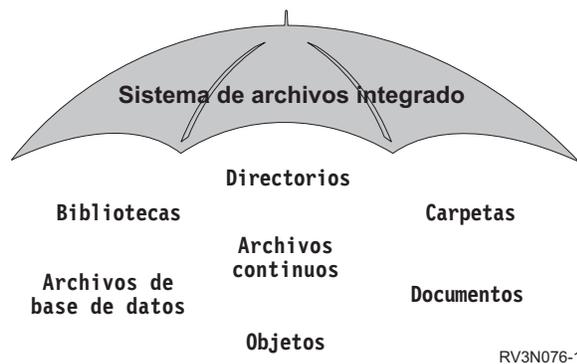


Figura 1. Una estructura que cubre toda la información almacenada en el servidor iSeries

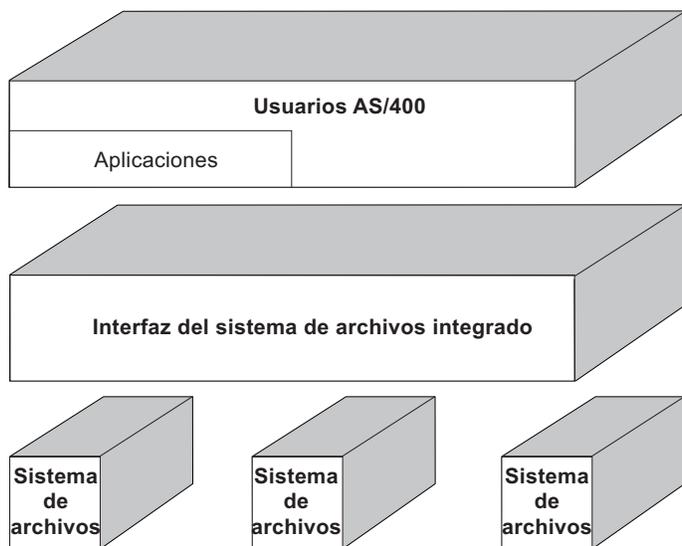
¿Por qué utilizar el sistema de archivos integrado?

El sistema de archivos integrado amplía la ya de por sí extensa capacidad de gestión de datos del OS/400 con posibilidades adicionales para mejorar el soporte a formas emergentes y futuras del proceso de información como, por ejemplo, cliente/servidor, sistemas abiertos y multimedia.

El sistema de archivos integrado puede utilizarse para:

- Ofrecer un acceso más rápido a los datos de OS/400, especialmente para aplicaciones como Client Access que utilizan el servidor de archivos OS/400.
- Permitir un manejo más eficaz de los tipos de datos continuos, como imágenes, audio y vídeo.

- Proporcionar una base de directorios y sistemas de archivos para dar soporte a los estándares de sistemas abiertos basados en UNIX como, por ejemplo, POSIX (Interfaz portátil de sistemas operativos para entornos informáticos) y XPG. Esta estructura de archivos y directorios también proporciona un entorno familiar para los usuarios de sistemas operativos de PC, como el sistema operativo en disco (DOS) y los sistemas operativos de Windows.
- Permitir manejar el soporte de archivos con posibilidades exclusivas (como los archivos de base de datos orientados a registros, archivos continuos basados en UNIX y servicio de archivos) como sistemas de archivos independientes, al mismo tiempo que permite manejarlos mediante una interfaz común.



RV3N062-1

Figura 2. Una interfaz común para distintos sistemas de archivos

- Permitir a los usuarios de PC aprovechar mejor la interfaz gráfica de usuario. Por ejemplo, los usuarios de Windows pueden utilizar las herramientas gráficas de Windows para trabajar con los archivos continuos y otros objetos del servidor iSeries, de la misma forma que lo hacen con los archivos almacenados en sus PC.
- Proporcionar continuidad en los nombres de objeto e información de objeto asociada al pasar de un idioma nacional a otro. Por ejemplo, se garantiza que los caracteres individuales seguirán siendo iguales al cambiar de la página de códigos de un idioma a la de otro.

Capítulo 2. Conceptos del sistema de archivos integrado

Archivo continuo

Un **archivo continuo** es una secuencia de bytes accesible aleatoriamente, sin ninguna otra estructura impuesta por el sistema. El sistema de archivos integrado proporciona soporte para almacenar información y trabajar con ella en la forma de archivos continuos. Los documentos almacenados en las carpetas del servidor son archivos continuos. Otros ejemplos de archivos continuos son los archivos de PC y los archivos de los sistemas UNIX. Un archivo continuo del sistema de archivos integrado es un objeto del sistema cuyo tipo es *STMF.

Para comprender mejor en qué consisten los archivos continuos, resulta efectivo compararlos con los archivos de base de datos del servidor iSeries. Un archivo de base de datos está orientado a registro; tiene subdivisiones predefinidas que constan de uno o más campos con características específicas, como por ejemplo longitud y tipo de datos.

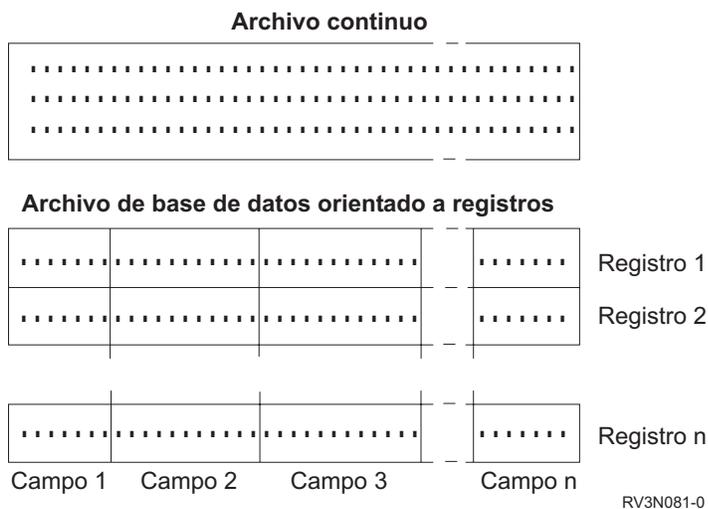


Figura 3. Comparación entre un archivo continuo y un archivo orientado a registro

Los archivos continuos y los archivos orientados a registro están estructurados de forma distinta, y esta diferencia estructural afecta a la utilización de los mismos. La estructura afecta a la forma de escribir una aplicación para interactuar con los archivos y en qué parte de la aplicación es más adecuado cada tipo de archivo. Un archivo orientado a registro, por ejemplo, resulta adecuado para almacenar estadísticas de cliente, como por ejemplo nombre, dirección y saldo de cuentas. Un archivo orientado a registro permite acceder individualmente a estos campos predefinidos y manipularlos, utilizando los amplios recursos de programación del servidor. Sin embargo, un archivo continuo es más adecuado para almacenar información como, por ejemplo, la imagen de un cliente, que se compone de una serie continua de bits que representan variaciones de color. Los archivos continuos son adecuados especialmente para almacenar series de datos como el texto de un documento, imágenes, audio y vídeo.

Para obtener más información sobre los archivos continuos en el sistema de archivos integrado, consulte:

- “Copia de datos entre archivos continuos y archivos de base de datos” en la página 45.
- “Archivos continuos *TYPE1 y *TYPE2”.

Archivos continuos *TYPE1 y *TYPE2

Un archivo tiene una de las dos opciones de formato siguientes: archivo continuo *TYPE1 o archivo continuo *TYPE2.

Un archivo continuo *TYPE1 tiene el mismo formato que los archivos continuos creados en releases anteriores a la versión 4, release 4 de OS/400- Se guarda más rápidamente que un archivo continuo *TYPE2 en releases anteriores a la versión 4 release 4 de OS/400. Tiene un tamaño mínimo de 4096 bytes.

El archivo continuo *TYPE2 dispone de acceso a archivos de alto rendimiento y es una novedad de la versión 4, release 4 del OS/400. Se guarda más lentamente que un archivo continuo *TYPE1 en releases anteriores a la versión 4, release 4 de OS/400. Tiene un tamaño mínimo de objeto de 4096 bytes. Todos los archivos creados con V4R4 y sistemas más nuevos son archivos continuos *TYPE2.

A pesar de que los archivos continuos *TYPE2 trabajan sólo con V4R4 y sistemas más nuevos, es posible guardar un archivo continuo *TYPE2 para poder restaurarlo más tarde en un sistema anterior a V4R4. Sin embargo, este proceso podría ser lento.

Sistemas de archivos en el sistema de archivos integrado

Un **sistema de archivos** proporciona el soporte para acceder a segmentos específicos de almacenamiento que están organizados como unidades lógicas. Estas unidades lógicas del servidor son los archivos, directorios, bibliotecas y objetos.

Cada sistema de archivos tiene un conjunto de estructuras y reglas lógicas para interactuar con la información situada en el almacenamiento. Dichas estructuras y reglas pueden ser distintas en cada sistema de archivos. De hecho, desde el punto de vista de las estructuras y las reglas, el soporte del OS/400 para acceder a los archivos de base de datos y a otros diversos tipos de objetos a través de bibliotecas puede considerarse un sistema de archivos. Igualmente, el soporte del OS/400 para acceder a documentos (que en realidad son archivos continuos) a través de la estructura de carpetas puede considerarse un sistema de archivos independiente.

El sistema de archivos integrado trata el soporte de bibliotecas y el soporte de carpetas como sistemas de archivos independientes. Otros tipos de soporte de gestión de archivos que tienen posibilidades diferentes también se tratan como sistemas de archivos independientes.

Puede consultarse una comparación de las características y limitaciones de cada sistema de archivos en "Comparación de los sistemas de archivos" en la página 61.

Los sistemas de archivos del sistema de archivos integrado son:

"raíz" (/)

Sistema de archivos "**raíz**" (/). Este sistema de archivos obtiene el máximo provecho del soporte de archivos continuos y de la estructura jerárquica de directorios del sistema de archivos integrado. Tiene las características de los sistemas de archivos de DOS y de OS/2.

QOpenSys

Sistema de archivos de sistemas abiertos. Es compatible con los estándares de sistemas abiertos basados en UNIX como, por ejemplo, POSIX y XPG. Al igual que el sistema de archivos raíz, este sistema de archivos aprovecha el soporte de archivos continuos y directorios proporcionado por el sistema de archivos integrado. Además, dispone de soporte para nombres de objeto que distinguen entre mayúsculas y minúsculas.

UDFS Sistema de archivos definido por el usuario. Este sistema de archivos reside en la agrupación de almacenamiento auxiliar (ASP) o en la agrupación de almacenamiento auxiliar (ASP) independiente que se elija. Es él quien crea y gestiona este sistema de archivos.

QSYS.LIB

Sistema de archivos de biblioteca. Da soporte a la estructura de bibliotecas del servidor. Este sistema de archivos proporciona acceso a los archivos de base de datos y al resto de tipos de objeto del iSeries server que gestiona el soporte de bibliotecas en las ASP de usuario básicas y del sistema.

ASP independiente QSYS.LIB

El sistema de archivos QSYS.LIB de ASP independiente. Este sistema de archivos ofrece soporte para su estructura de bibliotecas del servidor en cualquier agrupación de almacenamiento auxiliar (ASP) independiente que cree y defina. Este sistema de archivos proporciona acceso a los archivos de base de datos y al resto de tipos de objeto del iSeries server que gestiona el soporte de bibliotecas.

QDLS Sistema de archivos de servicios de biblioteca de documentos. Este sistema de archivos proporciona acceso a documentos y carpetas.

QOPT

Sistema de archivos óptico. Da acceso a los datos continuos almacenados en soportes ópticos.

QNetWare

Sistema de archivos QNetWare. Este sistema de archivos permite el acceso a objetos y datos locales o remotos almacenados en un servidor en el que se ejecuta Novell NetWare 4.10 o 4.11 o en Servidores PC autónomos en los que se ejecuta Novell NetWare 3.12, 4.10 4.11 o 5.0. El usuario puede montar de forma dinámica los sistemas de archivos NetWare sobre sistemas de archivos locales ya existentes.

QNTC Sistema de archivos de Windows NT Server. Este sistema de archivos proporciona acceso a los datos y objetos que se encuentran almacenados en un servidor que tenga instalado Windows NT 4.0 o una versión superior. Posibilita que las aplicaciones del servidor iSeries utilicen los mismos datos que los clientes de Windows NT. En este caso se incluye el acceso a los datos que se encuentran en un sistema Windows NT Server que se ejecute en un Servidor PC integrado. Si desea obtener detalles, puede consultar la publicación AS/400 Integración de OS/400 con Windows NT Server, SC10-3235 (SC41-5439).

QFileSvr.400

Este sistema de archivos proporciona acceso a otros sistemas de archivos que residen en servidores iSeries remotos.

NFS Sistema de archivos de red. Le permite acceder a los datos y objetos almacenados en un servidor NFS remoto. Un servidor NFS puede exportar un sistema de archivos de red que los clientes NFS montarán acto seguido dinámicamente.

Puede interactuar con cualquiera de los sistemas de archivos a través de una interfaz común. Esta interfaz está optimizada para entrada/salida de datos continuos, en contraste con la entrada/salida de registro proporcionada mediante las interfaces de gestión de datos. Los mandatos, menús y pantallas, e interfaces de programación de aplicaciones (API) suministradas permiten la interacción con los sistemas de archivos a través de esta interfaz común.

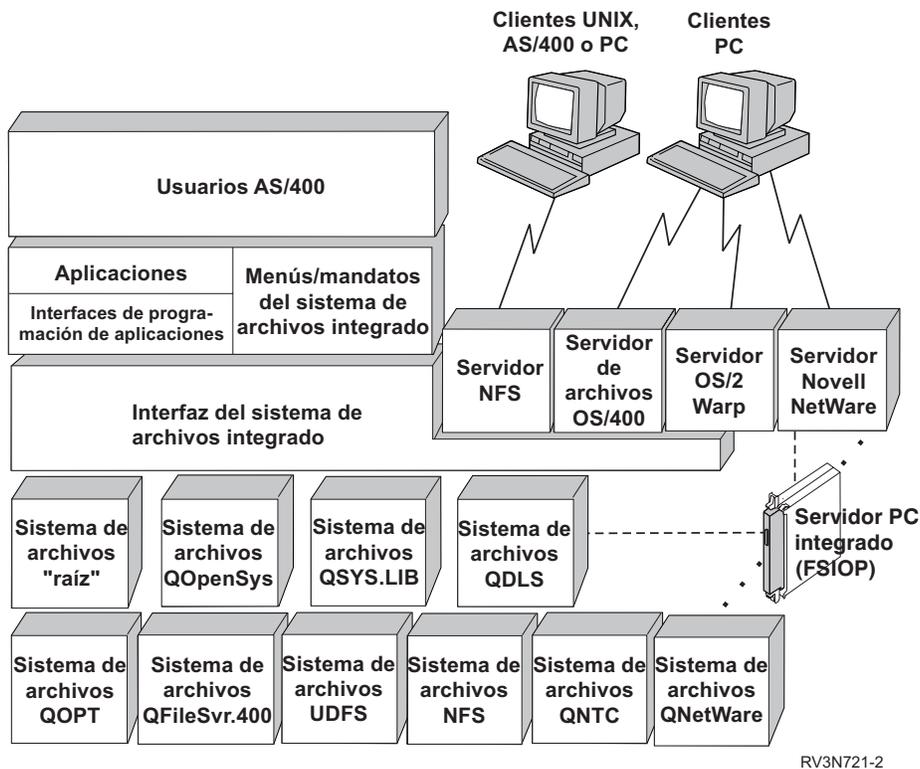


Figura 4. Sistemas de archivos, servidores de archivos y la interfaz del sistema de archivos integrado

Para obtener más información, consulte los temas y publicaciones siguientes:

- Optical Support 
- Soporte del sistema de archivos de red de OS/400 

Directorio

Un **directorio** es un objeto especial utilizado para ubicar objetos según los nombres especificados por el usuario. Cada directorio contiene una lista de objetos que están conectados a él. Dicha lista puede incluir a otros directorios.

El sistema de archivos integrado proporciona una estructura jerárquica de directorios que permite acceder a todos los objetos del servidor. Se puede imaginar esa estructura de directorios como un árbol invertido, donde la raíz está en la parte superior y las ramas en la inferior. Las ramas representan a los directorios en la jerarquía de directorios. Las ramas de directorios tienen ramas subordinadas que se denominan subdirectorios. Prendidos a las diversas ramas de directorios y subdirectorios hay objetos tales como archivos. La localización de un objeto requiere la especificación de una vía de acceso a través de los directorios hasta el subdirectorio al que está conectado el objeto. De los objetos conectados a un directorio determinado se dice a veces que están “en” ese directorio.

Una rama de directorio concreta junto con todas las ramas subordinadas (subdirectorios) y todos los objetos conectados con esas ramas recibe el nombre de **subárbol**. Cada uno de los sistemas de archivos es un subárbol principal de la estructura de directorios del sistema de archivos integrado. En el QSYS.LIB y los subárboles de los sistemas de archivos QSYS.LIB de las ASP independientes, las bibliotecas se manejan del mismo modo que los subdirectorios. Los objetos de una biblioteca se manejan como los objetos de un subdirectorio. Dado que los archivos de base de datos contienen objetos (miembros de archivos de base de datos), se manejan como subdirectorios y no como objetos. En el sistema de

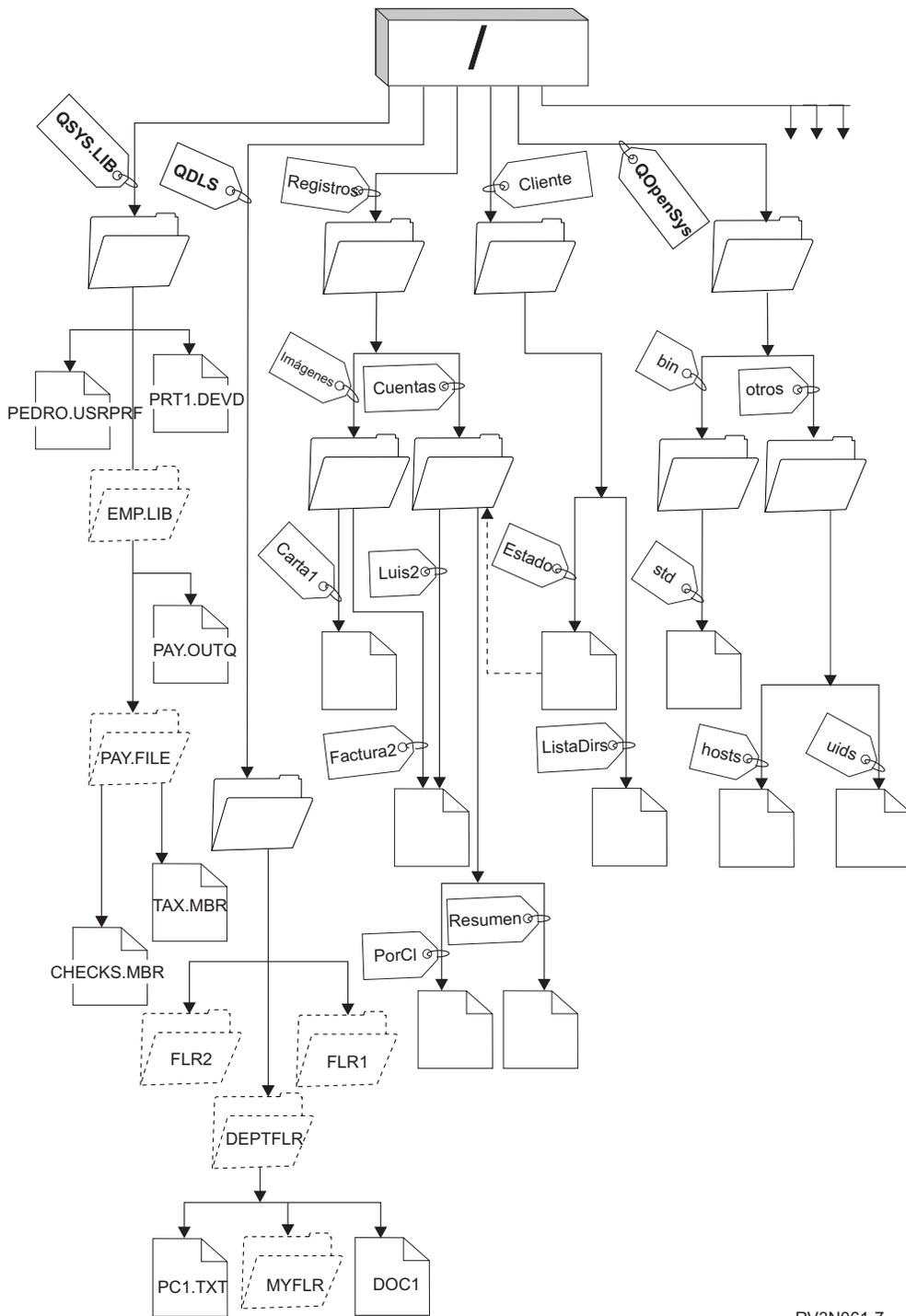
| archivos de servicios de biblioteca de documentos (subárbol QDLS), las carpetas se manejan como subdirectorios y los documentos de las carpetas se manejan como objetos de un subdirectorio.

Debido a las diferencias entre los sistemas de archivos, las operaciones que se pueden realizar en un subárbol de la jerarquía de directorios pueden no funcionar en otro subárbol.

El soporte de directorios del sistema de archivos integrado es similar al soporte de directorios proporcionado por el sistema de archivos de DOS. Además, proporciona características típicas de los sistemas UNIX, como es la posibilidad de almacenar un archivo una sola vez pero acceder a él a través de varias vías de acceso utilizando enlaces.

| Si quiere obtener más información sobre los directorios del sistema de archivos integrado, consulte:

- “Directorio actual y directorio de inicio” en la página 8
- “Directorios que proporciona el sistema de archivos integrado” en la página 35
- | • “Directorios *TYPE2” en la página 9



RV3N061-7

Figura 5. Los sistemas de archivos y los objetos son ramas del árbol de directorios del sistema de archivos integrado

Directorio actual y directorio de inicio

El **directorio actual** es el primer directorio en el que el sistema operativo busca los programas y archivos y almacena los archivos temporales y la salida. Cuando se solicita una operación sobre un objeto, como por ejemplo un archivo, el sistema busca el objeto en el directorio actual a menos que se especifique una

vía de acceso de directorio distinta. El directorio actual es semejante a la idea de biblioteca actual. También se denomina el **directorio de trabajo actual** o simplemente **directorio de trabajo**.

El **directorio inicial** se utiliza como el directorio actual cuando se inicia una sesión en el sistema. El nombre del directorio inicial se especifica en el perfil de usuario. Cuando se inicia un trabajo, el sistema busca en el perfil de usuario el nombre del directorio inicial. Si en el sistema no existe ningún directorio con ese nombre, el directorio inicial se cambia por el directorio "raíz" (/).

Normalmente, el administrador del sistema que crea el perfil de usuario para un usuario también crea el directorio inicial del usuario. Es aconsejable crear directorios iniciales individuales para cada usuario bajo el directorio /home. El directorio /home es un subdirectorío del directorio "raíz" (/). El valor por omisión del sistema espera que el nombre del directorio inicial del usuario sea el mismo que el del perfil de usuario.

Por ejemplo, el mandato CRTUSRPRF USRPRF(Pedro) HOMEDIR(*USRPRF) asignará el directorio inicial de Pedro a /home/PEDRO. Si el directorio /home/PEDRO no existe, el directorio raíz (/) pasa a ser el directorio inicial de Pedro.

Se puede especificar un directorio distinto al directorio inicial como directorio actual en cualquier momento después de iniciar la sesión, mediante el mandato CL Cambiar directorio actual (CHGCURDIR), la API chdir(), o la API fchdir().

El directorio inicial elegido durante la iniciación del proceso seguirá siendo por omisión el directorio inicial de cada hebra. Esta acción se realizará independientemente de si se cambia el perfil de usuario activo para la hebra tras la iniciación. No obstante, puede utilizarse el soporte que proporciona la API Cambiar Trabajo (QWTCHGJB) para cambiar el directorio inicial que utilizará una hebra por el directorio inicial del perfil de usuario actual de dicha hebra (o el directorio "raíz" (/) si dicho directorio inicial no existe). Las hebras secundarias siempre heredarán el directorio inicial de la hebra que las ha creado. Tenga en cuenta que el directorio actual del proceso no cambia cuando se utiliza QWTCHGJB para cambiar el directorio inicial de la hebra. El directorio actual tiene un ámbito de proceso, mientras que el directorio inicial tiene un ámbito de hebra. Al cambiar el directorio de trabajo actual en cualquier hebra se cambia para todo el proceso. Al cambiar el directorio inicial de una hebra no se cambia su directorio de trabajo actual.

Consulte el apartado Interfaces de programación de aplicaciones (API) para obtener más información sobre la API QWTCHGJB.

Directorios *TYPE2

Los sistemas de archivo "raíz" (/), QOpenSys, y los definidos por el usuario (UDFS) en el sistema de archivos integrado admiten el formato de los directorios *TYPE2. El formato de los directorios *TYPE2 es una mejora del formato de directorios *TYPE1 original. Los directorios *TYPE2 tienen una estructura interna diferente y una implementación diferente que los directorios *TYPE1.

Las ventajas de los directorios *TYPE2 son:

- Mejor rendimiento
- Mejor fiabilidad
- Mayor funcionalidad
- Menor espacio de almacenamiento auxiliar (en muchos casos)

Los directorios *TYPE2 mejoran el rendimiento del sistema de archivos de los directorios *TYPE1, especialmente al crear y suprimir directorios.

Los directorios *TYPE2 son más fiables que los directorios *TYPE1. Cuando un sistema finaliza de forma anómala, los directorios *TYPE2 se recuperan completamente a menos que se haya producido una anomalía de almacenamiento auxiliar. Los directorios *TYPE1 pueden requerir el uso del mandato Reclamar almacenamiento (RCLSTG) para la completa recuperación.

Los directorios *TYPE2 proporcionan la siguiente funcionalidad añadida:

1. Los directorios *TYPE2 admiten cambiar el nombre de mayúsculas a minúsculas y viceversa en un sistema de archivos que sólo admita un caso (por ejemplo, pasar de B a b).
2. Un objeto de un directorio *TYPE2 puede tener un máximo de un millón de enlaces, mientras que los directorios *TYPE1 sólo admiten 32.767 enlaces. Es decir que es posible tener hasta 1 millón de enlaces fijos a un archivo continuo, y un directorio *TYPE2 puede incluir hasta 1 millón de subdirectorios.
3. Con el iSeries Navigator, la lista de entradas se clasifican automáticamente en orden binario al abrir un directorio con el formato *TYPE2.

Normalmente, los directorios *TYPE2 que tienen menos de 350 objetos requieren menos almacenamiento auxiliar que los directorios *TYPE1 con el mismo número de objetos. Los directorios de *TYPE2 con más de 350 objetos son un 10 por ciento mayores (de media) que los directorios *TYPE1.

Existen varias formas de obtener directorios *TYPE2 en su sistema:

- Un sistema de archivos definido por el usuario (UDFS) en una agrupación de almacenamiento auxiliar (ASP) independiente se convierte a formato *TYPE2 la primera vez que la ASP independiente se activa en un sistema instalado con la V5R2 del OS/400.
- El resto de sistemas de archivo admitidos en ASP independientes (excepto los UDFS) deben convertirse a *TYPE2 utilizando el mandato Convertir directorio (CVTDIR).
- Los nuevos servidores del iSeries en los que previamente se ha instalado la V5R2 del OS/400 tienen directorios *TYPE2. No es necesario ningún tipo de conversión para los sistemas de archivo "raíz" (/), QOpenSys, y UDFS en las ASP 1-32.
- Una instalación de la V5R2 del OS/400 partiendo de cero en un iSeries server incluye directorios *TYPE2. No es necesario ningún tipo de conversión para los sistemas de archivo "raíz" (/), QOpenSys, y UDFS en las ASP 1-32.

Para determinar el formato de directorios de los sistemas de archivos del servidor, utilice el mandato Convertir directorio (CVTDIR):

```
CVTDIR OPTION(*CHECK).
```

Nota: En la V5R1 del OS/400 se admiten los directorios *TYPE2, pero existen algunas diferencias con el soporte de directorios *TYPE2 normal. Para obtener más información, consulte Utilización de los directorios *TYPE2 en la V5R1 del OS/400.

Para obtener más información sobre los directorios *TYPE2, consulte los siguientes temas:

- Conversión a un directorio *TYPE2
- Los sistemas de archivos "Raíz", QOpenSys, o UDFS no están disponibles
- Requisitos de almacenamiento auxiliar
- Consideraciones de enlaces simbólicos
- Agrupaciones de almacenamiento auxiliar (ASP) independientes
- Consideraciones sobre Guardar/Restaurar
- Preparación para la conversión a *TYPE2
- Proceso de conversión
- Ejemplo: conversión de todos los sistemas de archivos (número pequeño de objetos)
- Ejemplo: conversión de todos los sistemas de archivos (número elevado de objetos)
- Ejemplo: conversión de sólo determinadas ASP

Utilización de los directorios *TYPE2 en la V5R1 del OS/400.

Los sistemas de archivo "raíz" (/), QOpenSys, y los definidos por el usuario (UDFS) en el sistema de archivos integrado admiten el formato de los directorios *TYPE2 en la V5R1 del OS/400. El formato de los directorios *TYPE2 es una mejora del formato de directorios *TYPE1 original. Los directorios *TYPE2 tienen una estructura interna diferente a la de los directorios *TYPE1 y ofrecen mejor rendimiento y mayor fiabilidad.

Si tiene la V5R1, puede convertir los directorios de la V5R1 al formato de directorios *TYPE2. Es recomendable la conversión al formato de directorios *TYPE2 **antes** de instalar un release nuevo del OS/400. Es necesario porque la conversión de directorios puede efectuarse automáticamente durante la instalación. Una conversión automática durante la instalación aumentaría **significativamente** el tiempo necesario para la instalación.

Nota: Si se efectúa una instalación a la V5R1 o V5R2 del OS/400, **no** tiene lugar una conversión automática al formato de directorios *TYPE2. No es necesaria la conversión de los directorios antes de estas instalaciones.

Los parches PTF permiten obtener soporte para los directorios *TYPE2 en la V5R1. La herramienta de conversión es algo diferente de la utilizada en la versión V5R2. Consulte el APAR II13161 informativo para obtener más información sobre los directorios *TYPE2 en la V5R1. Utilice uno de los métodos siguientes para acceder al APAR:

1. Descargue el APAR informativo en su iSeries server y visuelécelo. Utilice los mandatos siguientes:

```
SNDPTFORD PTFID((II13161))  
DSPPTFCVR LICPGM(INFOAS4) SELECT(II13161)
```
2. Acceda a <http://www-912.ibm.com>  sitio Web para consultar el APAR informativo en línea. Seleccione **Authorized Program Analysis Reports (APAR) → V5R1 APAR → APAR number II13161**.

Conversión a un directorio *TYPE2

El mandato CVTDIR se encarga de la conversión de un directorio *TYPE1 a un directorio *TYPE2. Adicionalmente, proporciona información sobre cómo convertir sistemas de archivos al formato de directorios *TYPE2. CVTDIR efectúa las siguientes tareas:

- Lista el formato de directorios actual para los sistemas de archivos existentes que admiten el formato de directorios *TYPE2.
- Estima el tiempo necesario para la conversión.
- Estima los requisitos de almacenamiento auxiliar para la conversión.
- Convierte los sistemas de archivos al formato *TYPE2. Se convierte cualquier directorio existente a *TYPE2, y cualquier directorio nuevo creado después de la conversión tiene el formato *TYPE2.

Existen diversas formas para convertir los directorios a uno de los sistemas de archivos:

- Manualmente, utilizando el mandato CVTDIR:
- Automáticamente, la primera vez que se activa una ASP independiente en un sistema con la V5R2 del OS/400 instalada.
- Durante una IPL, si el sistema determina que la conversión de un sistema de archivos estaba en curso cuando el sistema finalizó de forma anómala.
- Durante el mandato Reclamar almacenamiento (RCLSTG SELECT(*ALL)), si se encuentran directorios *TYPE1 perdidos que forman parte de un sistema de archivos que se convirtió al formato *TYPE2.

Los sistemas de archivos "Raíz", QOpenSys, o UDFS no están disponibles

La conversión de los sistemas de archivo "raíz" (/) o QOpenSys debe efectuarse mientras el sistema se encuentra en estado restringido. Para la conversión de UDFS, no es necesario que el sistema se encuentre en estado restringido; sin embargo, los UDFS de esa ASP no estarán disponibles durante la conversión. El período de tiempo requerido para efectuar la conversión depende del tamaño del sistema de archivos. Por lo tanto, es necesaria una planificación para programar cuál es el mejor momento para efectuar la conversión. La opción *ESTIMATE del mandato CVTDIR estima el período de tiempo necesario para convertir el sistema de archivos especificado. El período de tiempo estimado es el valor estimado más elevado. Estima el período de tiempo en una ejecución de conversión en un trabajo con una sola hebra. La conversión real utiliza múltiples hebras y debería tardar menos que el tiempo estimado. Normalmente, los sistemas de archivos con más de 40.000 enlaces pueden convertirse en 30-50 por ciento del tiempo estimado. Sin embargo, el tiempo real depende del hardware y la configuración del servidor.

Si el sistema finaliza de forma anómala durante la ejecución del mandato CVTDIR, durante la posterior IPL la función de conversión se ejecuta en el trabajo SCPF. El trabajo SCPF no permite que haya múltiples hebras activas. Por lo tanto, cuando es necesario completar la conversión de un sistema de archivos durante la IPL, ésta utiliza una sola hebra. La función de conversión se ejecuta cuando se visualiza SRC C900 2A85 durante la IPL, y se muestra el mensaje de estado CPIA089 que indica el progreso de la conversión.

La función de conversión se ejecuta durante RCLSTG si existen directorios *TYPE1 perdidos de un sistema de archivos que ya se ha convertido a *TYPE2. La función de conversión se ejecuta en el trabajo que emite el mandato RCLSTG. Si se encuentran directorios perdidos que deben convertirse, la conversión se ejecuta en una hebra única debido a restricciones del sistema.

Requisitos de almacenamiento auxiliar

Deben considerarse requisitos de almacenamiento auxiliar antes de convertir los directorios de un sistema de archivos al formato *TYPE2. Existen varios asuntos a considerar referentes a los requisitos de almacenamiento auxiliar:

- El tamaño final de los directorios tras la conversión al formato *TYPE2
- El almacenamiento adicional requerido cuando se ejecuta la función de conversión

En muchos casos, el tamaño final de un directorio *TYPE2 es menor que el de un directorio *TYPE1. Normalmente, los directorios *TYPE2 que tienen menos de 350 objetos requieren menos almacenamiento auxiliar que los directorios *TYPE1 con el mismo número de objetos. Los directorios de *TYPE2 con más de 350 objetos son un 10 por ciento mayores (de media) que los directorios *TYPE1.

Mientras se ejecuta la función de conversión, se requiere almacenamiento adicional. La función de conversión requiere la existencia simultánea de un número de directorios que tienen la versión *TYPE1 y la versión *TYPE2. El número depende de la configuración del iSeries server y la estructura de directorios del sistema de archivos que se está convirtiendo.

La opción *ESTIMATE del mandato CVTDIR proporcionará información estimada sobre el almacenamiento auxiliar que se necesitará durante la conversión.

Consideraciones de enlaces simbólicos

Los enlaces simbólicos son objetos del sistema de archivos integrado que contienen una vía de acceso a otro objeto. Existen algunas instancias durante la conversión que permiten modificar el nombre de un objeto. Si durante la conversión se cambia el nombre de uno de los elementos de la vía de acceso de un enlace simbólico, los contenidos del enlace simbólico dejarán de referirse al objeto. Consulte Cambiar el nombre de los objetos para obtener más información.

Agrupaciones de almacenamiento auxiliar (ASP) independientes

La primera vez que se activa una ASP independiente en un sistema con la versión V5R2 del OS/400 instalada, los directorios se convierten a *TYPE2. Para finalidades de planificación, en la V5R1 del OS/400 se proporciona una función de estimación que ofrece información sobre el período de tiempo que tardará la conversión. Antes de activar la ASP independiente en el servidor V5R2, ejecute la siguiente API en su sistema V5R1 al activar la ASP independiente (con el nombre ASP_NAME):

```
CALL QP0FCVT2 (*ESTIMATE ASP_NAME *TYPE2)
```

Nota: Se recomienda ejecutar RCLSTG en la ASP independiente del sistema V5R1 antes de llamar a esta función.

Consideraciones sobre Guardar/Restaurar

Los directorios que existen como *TYPE1 pueden guardarse y restaurarse en un sistema de archivos que se ha convertido a *TYPE2. Del mismo modo, es posible guardar los directorios que existen como *TYPE2 y restaurarlos en un sistema de archivos con el formato *TYPE1, siempre que no se hayan excedido ninguno de los límites de *TYPE1 cuando el directorio existía como directorio *TYPE2.

Preparación para la conversión a *TYPE2

Existen varios mandatos CL y parámetros recomendados antes de efectuar la conversión al formato de directorios *TYPE2:

- Reclamar almacenamiento (RCLSTG)

El uso del mandato RCLSTG SELECT(*ALL) antes de convertir cualquier sistema de archivos, limpia los directorios y garantiza que los directorios son adecuados. Aunque de este modo no se eliminan todos los posibles problemas que podrían surgir durante una conversión de directorios, se garantiza que podrán leerse los directorios en el sistema de archivos.

Este mandato debe ejecutarse sólo una vez antes de utilizar cualquier opción del mandato CVTDIR.

- Guardar sistema (SAVSYS)

Es aconsejable guardar todo el sistema del iSeries server antes de ejecutar RCLSTG y antes de utilizar la opción *CONVERT del mandato CVTDIR. Para efectuar una copia de seguridad del sistema, utilice el menú Guardar del iSeries. Para acceder al menú Guardar, entre GO SAVE en cualquier línea de comandos, y seleccione la opción 21. Para obtener más información, consulte Copias de seguridad y

recuperación .

Este mandato debe ejecutarse sólo una vez antes de utilizar cualquier opción del mandato CVTDIR.

- La opción *ESTIMATE del mandato CVTDIR.

Utilice la opción *ESTIMATE del mandato CVTDIR para averiguar el tiempo necesario para la conversión de los directorios.

Además de proporcionar estimaciones sobre el tiempo y el almacenamiento auxiliar necesarios, la opción *ESTIMATE tiene otras ventajas. Crea algunos objetos secundarios asociados con los directorios *TYPE1 y ello permite que la conversión sea más rápida (pues no debe crearlos). Estos objetos secundarios siguen existiendo hasta que se utiliza la opción *CONVERT para convertir el sistema de archivos al formato *TYPE2. La opción *ESTIMATE también lee todos los directorios del sistema de archivos, que verifica implícitamente los directorios. La opción *ESTIMATE no garantiza encontrar todos los errores posibles que podrían aparecer durante la conversión real, pero es de utilidad.

Tras ejecutar la opción *ESTIMATE, compruebe si aparece algún error en las anotaciones cronológicas y lleve a cabo cualquier acción de recuperación recomendada antes de convertir el sistema de archivos.

No es imprescindible ejecutar la estimación de nuevo tras efectuar las acciones de recuperación recomendadas, pero podría ser conveniente para comprobar que no surgen otros problemas.

- Consideraciones de almacenamiento auxiliar

Compruebe el almacenamiento auxiliar disponible para la ASP que contiene el sistema de archivos que se está convirtiendo. La opción CVTDIR *ESTIMATE muestra el mensaje CPIA090, que indica el

almacenamiento auxiliar disponible para la ASP. Adicionalmente, muestra una previsión del almacenamiento auxiliar que se necesitará durante la conversión. También aparece el mensaje CPIA091, que indica si se estima que el tamaño total de los directorios con el formato *TYPE2 en el sistema de archivos sea mayor o menor que los directorios *TYPE1 existentes tras la conversión. El almacenamiento disponible en la ASP debe ser la suma del almacenamiento auxiliar no utilizado (visualizada en el mensaje CPIA090), y la diferencia entre el tamaño de los directorios con el formato *TYPE1 y el tamaño de los directorios con el formato *TYPE2 (visualizada en el mensaje CPIA091).

También es posible consultar el almacenamiento auxiliar disponible utilizando el mandato Iniciar herramientas de servicio del sistema (STRSST) y seleccionando la opción Trabajar con unidades de disco.

Nota: Si hay sólo una ASP definida en el sistema, basta con el mandato Trabajar con el estado del sistema (WRKSYSSTS) para mostrar información sobre el almacenamiento auxiliar disponible.

Es aconsejable efectuar una limpieza general del sistema antes de utilizar cualquiera de las opciones del mandato CVTDIR. Si existen directorios o archivos que ya no son necesarios, elimínelos antes de utilizar las opciones del mandato CVTDIR. De este modo se libera espacio de almacenamiento auxiliar, y permite la conversión en un espacio reducido de tiempo debido a que hay menos objetos que deben procesarse.

- Considere modificar la acción global de la cola de mensajes del trabajo a *PRTWRAP para el trabajo que emite el mandato CVTDIR. Esta acción tiene las siguientes ventajas:
 1. Evita la finalización anómala del trabajo cuando se llenan las anotaciones de trabajo.
 2. Imprime en un archivo de spool los mensajes que se solapan cuando se lleva el archivo de anotaciones de trabajo y las nuevas entradas machacan a las anteriores; de este modo no se pierde ningún mensaje importante.
- En los sistemas con ASP independientes: antes de activar la ASP independiente en un sistema con la V5R2del OS/400, utilice la función *ESTIMATE de la V5R1 para todas las ASP independientes. Esta función proporciona una estimación temporal de cuánto tardará la primera activación de la ASP independiente tras la instalación. Para obtener más información, consulte Agrupaciones de almacenamiento auxiliar (ASP) independientes.

Proceso de conversión

El mandato CVTDIR convierte los directorios *TYPE1 en directorios *TYPE2. Existen varios puntos que deben considerarse durante el proceso de conversión:

- Conversión de "raíz" (/) o QOpenSys
- Conversión de sistemas de archivos definidos por el usuario
- Creación de perfiles de usuario
- Renombrar objetos
- Consideraciones sobre el perfil de usuario

Conversión de "raíz" o QOpenSys

Cuando se convierten los sistemas de archivos "raíz" o QOpenSys, el sistema **debe** estar en estado restringido. No es posible utilizar ningún sistema de archivos durante la conversión. El mandato CVTDIR no monta ningún sistema de archivos UDFS y NFS, y estos sistemas de archivos no vuelven a montarse al finalizar la conversión. Para volver a montar un sistema de archivos UDFS o NFS puede utilizarse el mandato Montar (MOUNT).

Conversión de sistemas de archivos definidos por el usuario

Cuando se convierten los UDFS en las ASP 1-32, estos no están disponibles para ningún usuario del sistema. Para cada una de estas ASP, existe un directorio QASPxx en el directorio /dev. Durante la ejecución del mandato CVTDIR, éste elimina el directorio QASPxx del espacio de nombre para evitar que algún usuario pueda acceder a los UDFS en la ASP. Cuando el mandato CVTDIR finaliza el proceso de todos los objetos, inclusive el directorio QASPxx, los objetos vuelven a colocarse en el espacio de nombre y

vuelven a ponerse a disponibilidad de los usuarios del sistema. El mandato CVTDIR no monta los UDFS para la ASP, y estos sistemas de archivos no vuelven a montarse al finalizar la conversión. Para volver a montar un sistema de archivos UDFS puede utilizarse el mandato Montar (MOUNT).

Nota: El directorio QASP01 existe en todos los sistemas.

Creación de perfiles de usuario

La función de conversión crea perfiles de usuario que se utilizan durante la ejecución de la función de conversión. Los perfiles de usuario tienen el nombre QP0FCVxxxx, siendo xxxx un número como 0001. La función de conversión los utiliza para tomar la propiedad de directorios del sistema de archivos que se está convirtiendo cuando el propietario original no es capaz de tomar la propiedad de sus directorios.

Si es posible, estos perfiles de usuario se suprimen al finalizar la conversión. Cuando se otorga la propiedad de un directorio a uno de estos perfiles de usuario se envía el mensaje CPIA08B.

Renombrar objetos

Los directorios *TYPE2 requieren que los nombres de enlace sean nombres UTF-16 válidos. Ello no sucede en los directorios *TYPE1, que tienen nombres UCS2 de nivel 1. Por este motivo pueden encontrarse nombres no válidos o duplicados durante una conversión de directorio. Cuando se encuentra un nombre inválido o duplicado, el nombre se convierte en un nombre UTF-16 válido y unívoco, y se envía el mensaje CPIA08A a las anotaciones de trabajo con el nombre original y el nombre nuevo. Los caracteres combinados o las parejas de caracteres sustitutos inválidas de un nombre pueden provocar el cambio del nombre de un objeto.

Para obtener más información sobre UTF-16, consulte la página Web de Unicode (<http://www.unicode.org>



Caracteres combinados: Algunos caracteres pueden estar formados por varios caracteres Unicode. Por ejemplo, existen caracteres con acentos o diéresis. Estos caracteres deben cambiarse, o normalizarse, a un formato común antes de almacenarse en el directorio de forma que todos los objetos tengan un nombre unívoco. La normalización de un carácter combinado es un proceso mediante el cual se asigna al carácter un formato conocido y predecible. El formato elegido para los directorios *TYPE2 es el formato canónico compuesto. Si existen dos objetos en un directorio *TYPE1 que contienen los mismos caracteres combinados, estos se normalizan al mismo nombre. Ello provoca una colisión, incluso cuando un objeto contiene caracteres combinados compuestos y el otro objeto contiene caracteres combinados descompuestos. Por lo tanto, a uno de ellos se le cambiará el nombre antes de enlazarlo al directorio *TYPE2.

Sustitución de caracteres: Algunos caracteres no tienen una representación válida en Unicode. Estos caracteres tienen algunos valores especiales; están compuestos por dos caracteres Unicode en dos ámbitos específicos de forma que el primer carácter Unicode se encuentra en un intervalo (por ejemplo 0xD800-0xD8FF) y el segundo carácter Unicode se encuentra en el segundo intervalo (por ejemplo 0xDC00-0xDCFF). Esto se conoce como una pareja sustitutoria. Si falta uno de los caracteres Unicode o si no son correctos (sólo un carácter parcial), se trata de un nombre inválido. Se han permitido nombres de este tipo en los directorios *TYPE1, pero no en los directorios *TYPE2. Para que la función de conversión continúe, si se encuentra un nombre que contenga uno de estos nombres inválidos, se cambiará el nombre antes de enlazar el objeto en el directorio *TYPE2.

Consideraciones sobre el perfil de usuario

Durante la ejecución de la conversión, se intenta asegurar que el mismo perfil de usuario que posee cualquier directorio *TYPE1 seguirá siendo el propietario de los correspondientes directorios *TYPE2. Puesto que momentáneamente existen los directorios *TYPE1 y *TYPE2 simultáneamente, ello afecta a la cantidad de almacenamiento que posee el perfil de usuario y al número de entradas en el perfil de usuario.

Modificar el almacenamiento máximo para un perfil de usuario: Durante el proceso de conversión de directorios, existe un número de directorios que momentáneamente existen en ambos formatos al mismo

tiempo y que pertenecen al mismo perfil de usuario. Cuando no puede crearse el directorio *TYPE2 debido a que se ha alcanzado el límite de almacenamiento máximo para el perfil de usuario, se aumenta el límite para dicho perfil de usuario. Se envía el mensaje CPIA08C a las anotaciones de trabajo y sigue la conversión.

Modificar el propietario de un directorio: Si el perfil de usuario que posee el directorio *TYPE1 no puede ser el propietario del directorio *TYPE2 que se crea, el propietario del directorio *TYPE2 se establece en uno de los perfiles de usuario alternativos que se describen en Crear perfiles de usuario. Se envía el mensaje CPIA08B a las anotaciones de trabajo y sigue la conversión.

Ejemplo: conversión de todos los sistemas de archivos (número pequeño de objetos)

El sistema A tiene configuradas cinco agrupaciones de almacenamiento auxiliar (ASP): 1 (la ASP del sistema), 3, 5, 11 y 25. No se ha convertido ningún sistema de archivos de los directorios *TYPE1 a los directorios *TYPE2 en el sistema. Desearía convertir todos los sistemas de archivos. Los sistemas de archivos no contienen un número elevado de objetos, por lo tanto planea efectuar todos los pasos en un solo día.

Para convertir los directorios de todos los sistemas de archivos con un número reducido de objetos:

1. Coloque el sistema en estado restringido:
2. Entre RCLSTG SELECT(*ALL) en la línea de comandos.
3. Guarde el sistema con el menú Guardar. Introduzca G0 SAVE en la línea de comandos, y seleccione la opción 21.
4. Introduzca CVTDIR OPTION(*ESTIMATE) FILESYS(*ALL) FORMAT(*TYPE2) en la línea de comandos.
5. Compruebe la presencia de mensajes de error desde la función *ESTIMATE.
6. Compruebe que todas las ASP tienen suficiente espacio de almacenamiento auxiliar disponible.
7. Entre CVTDIR OPTION(*CONVERT) FILESYS(*ALL) FORMAT(*TYPE2) en la línea de comandos.

Nota: Cuando convierta todos los sistemas de archivos (*ALL), aparecerá el mensaje CPAA084 y le solicitará que verifique si deben convertirse los sistemas de archivos de la lista.

8. Compruebe la presencia de mensajes de error desde la función *CONVERT.
9. Saque al sistema del estado restringido.

Ejemplo: conversión de todos los sistemas de archivos (número elevado de objetos)

El sistema B tiene configuradas cinco agrupaciones de almacenamiento auxiliar (ASP): 1 (la ASP del sistema), 3, 5, 11 y 25. No se ha convertido ningún sistema de archivos de los directorios *TYPE1 a los directorios *TYPE2 en el sistema. Desearía convertir todos los sistemas de archivos. Los sistemas de archivos contienen un número elevado de objetos, por lo tanto planea efectuar la conversión en dos fines de semana diferentes.

Primer fin de semana:

1. Coloque al sistema en estado restringido:
2. Entre RCLSTG SELECT(*ALL) en la línea de comandos.
3. Guarde el sistema con el menú Guardar. Introduzca G0 SAVE en la línea de comandos, y seleccione la opción 21.
4. Saque al sistema del estado restringido.

Durante la semana:

5. Introduzca CVTDIR OPTION(*ESTIMATE) FILESYS(*ALL) FORMAT(*TYPE2) en la línea de comandos.
6. Compruebe la presencia de mensajes de error desde la función *ESTIMATE.

7. Compruebe que todas las ASP tienen suficiente espacio de almacenamiento auxiliar disponible.

Segundo fin de semana:

8. Coloque al sistema en estado restringido.

9. Entre `CVTDIR OPTION(*CONVERT) FILESYS(*ALL) FORMAT(*TYPE2)` en la línea de comandos.

Nota: Cuando convierta todos los sistemas de archivos (*ALL), aparecerá el mensaje CPAA084 y le solicitará que verifique si deben convertirse los sistemas de archivos de la lista.

10. Compruebe la presencia de mensajes de error desde la función *CONVERT.

11. Saque al sistema del estado restringido.

Ejemplo: conversión de sólo determinadas ASP

El sistema C tiene configuradas seis agrupaciones de almacenamiento auxiliar (ASP): 1 (la ASP del sistema), 2, 4, 8, 10 y 30. No se ha convertido ningún sistema de archivos del sistema. Desea convertir sólo el UDFS de las ASP 4, 10 y 30.

Para convertir los directorios de los UDFS de determinadas ASP:

1. Compruebe el formato de directorio de los sistemas de archivos. Para ello entre `CVTDIR OPTION(*CHECK)` en la línea de comandos.

2. Coloque al sistema en estado restringido.

3. Entre `RCLSTG SELECT(*ALL)` en la línea de comandos.

4. Guarde el sistema con el menú Guardar. Introduzca `G0 SAVE` en la línea de comandos, y seleccione la opción 21.

5. Saque al sistema del estado restringido.

6. Entre `CVTDIR OPTION(*ESTIMATE) FILESYS(*UDFS) ASP(4) FORMAT(*TYPE2)` en la línea de comandos.

7. Compruebe la presencia de mensajes de error desde la función *ESTIMATE.

8. Entre `CVTDIR OPTION(*ESTIMATE) FILESYS(*UDFS) ASP(10) FORMAT(*TYPE2)` en la línea de comandos.

9. Compruebe la presencia de mensajes de error desde la función *ESTIMATE.

10. Entre `CVTDIR OPTION(*ESTIMATE) FILESYS(*UDFS) ASP(30) FORMAT(*TYPE2)` en la línea de comandos.

11. Compruebe la presencia de mensajes de error desde la función *ESTIMATE.

12. Compruebe que todas las ASP tienen suficiente espacio de almacenamiento auxiliar disponible.

13. Entre `CVTDIR OPTION(*CONVERT) FILESYS(*UDFS) ASP(4) FORMAT(*TYPE2)` en la línea de comandos.

14. Compruebe la presencia de mensajes de error desde la función *CONVERT.

15. Entre `CVTDIR OPTION(*CONVERT) FILESYS(*UDFS) ASP(10) FORMAT(*TYPE2)` en la línea de comandos.

16. Compruebe la presencia de mensajes de error desde la función *CONVERT.

17. Entre `CVTDIR OPTION(*CONVERT) FILESYS(*UDFS) ASP(30) FORMAT(*TYPE2)` en la línea de comandos.

18. Compruebe la presencia de mensajes de error desde la función *CONVERT.

Nombre de vía de acceso

Un **nombre de vía de acceso** le indica al servidor cómo localizar un objeto. El nombre de vía de acceso se expresa como una secuencia de nombres de directorio seguidos del nombre del objeto. Los directorios individuales y el nombre de objeto se separan con el carácter barra inclinada (/); por ejemplo:

```
directorio1/directorio2/archivo
```

En los mandatos del sistema de archivos integrado también puede utilizarse la barra inclinada invertida (\) en lugar de la barra inclinada.

Existen dos modos de indicar un nombre de vía de acceso:

- Un **nombre de vía de acceso absoluto** empieza en el nivel superior, o directorio "raíz" (que se identifica por el carácter /). Por ejemplo, piense en la vía de acceso siguiente desde el directorio / hasta el archivo denominado Pérez.

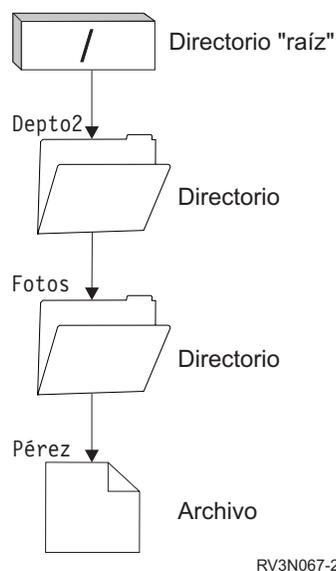


Figura 6. Los componentes de un nombre de vía de acceso

El nombre de vía de acceso absoluto al archivo Pérez es el siguiente:

/Depto2/Fotos/Pérez

El nombre de vía de acceso absoluto también se denomina **nombre de vía de acceso completo**.

- Si el nombre de la vía de acceso no empieza con el carácter /, el sistema presupone que la vía de acceso empieza en el directorio actual. Este tipo de nombre de vía de acceso se denomina **nombre de vía de acceso relativo**. Por ejemplo, si el directorio actual es Depto2 y tiene un subdirectorio denominado Fotos que contiene el archivo Pérez, el nombre de vía de acceso relativo al archivo es:

Fotos/Pérez

Observe que el nombre de vía de acceso no incluye el nombre del directorio actual. El primer elemento del nombre es el directorio u objeto del *siguiente nivel inferior* del directorio actual.

Enlace

Un **enlace** es una conexión especificada entre un directorio y un objeto. Un usuario o un programa pueden indicar al servidor dónde encontrar un objeto especificando el nombre de un enlace con el objeto. Un enlace se puede utilizar como nombre de vía de acceso o como componente de un nombre de vía de acceso.

Para los usuarios de sistemas de archivos basados en directorios, es conveniente pensar en un objeto, como por ejemplo un archivo, como algo que tiene un nombre que lo identifica ante el servidor. De hecho, se identifica mediante la vía de acceso de directorios hasta el objeto. En ocasiones, se puede acceder a un objeto simplemente facilitando el "nombre" del objeto. Esto se puede efectuar únicamente porque el sistema se ha diseñado para suponer cuál es la parte correspondiente al directorio de la vía de acceso en determinadas condiciones. La idea de un enlace aprovecha el hecho de que es la vía de acceso de directorios la que identifica al objeto. El nombre se asigna al enlace y no al objeto.

Una vez se acostumbre a la idea de que el nombre corresponde al enlace y no al objeto, empezará a vislumbrar posibilidades que habían permanecido ocultas. Puede haber múltiples enlaces al mismo objeto. Por ejemplo, dos usuarios pueden compartir un archivo teniendo un enlace desde el directorio inicial de cada usuario hasta el archivo (consulte el apartado “Directorio actual y directorio de inicio” en la página 8). Determinados tipos de enlace pueden cruzar sistemas de archivos y pueden existir sin que exista un objeto.

Hay dos tipos de enlaces: **fijos** y **simbólicos**.

Si desea obtener más información sobre los enlaces, consulte los temas siguientes:

- Enlace fijo
- Enlace simbólico
- Comparación: enlace fijo y enlace simbólico

Enlace fijo

Un **enlace fijo**, en ocasiones denominado simplemente enlace, no puede existir a menos que esté enlazado con un objeto real. Cuando un objeto se crea en un directorio (por ejemplo, al copiar un archivo en un directorio), el primer enlace fijo se establece entre el directorio y el objeto. Los usuarios y los programas de aplicación pueden añadir otros enlaces fijos. Cada enlace fijo se indica con una entrada de directorio independiente en el directorio. Los enlaces procedentes del mismo directorio no pueden tener el mismo nombre, pero los enlaces procedentes de directorios distintos sí.

Si el sistema de archivos lo admite, puede haber varios enlaces fijos a un objeto, procedentes del mismo directorio o de directorios distintos. La única excepción es que el objeto sea otro directorio. Sólo puede haber un enlace fijo de un directorio con otro directorio.

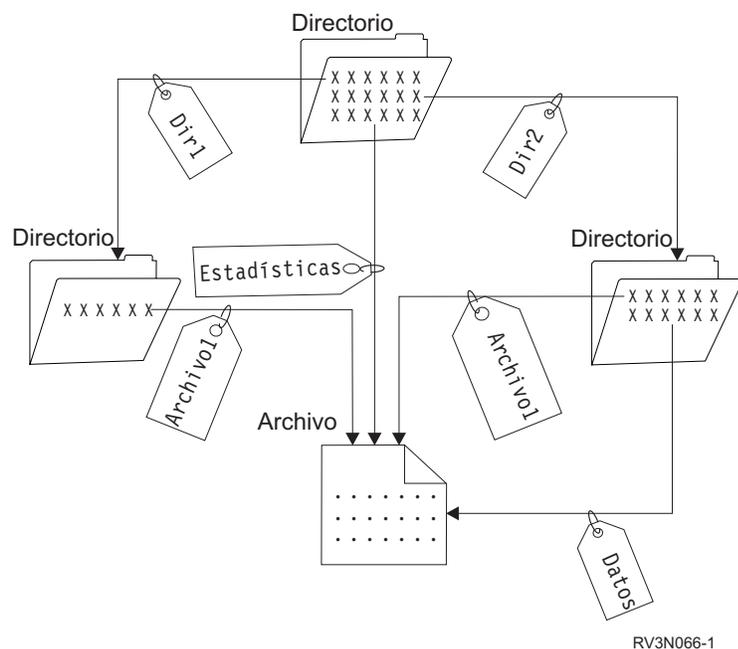


Figura 7. Una entrada de directorio define cada enlace fijo.

Los enlaces fijos se pueden eliminar sin que ello afecte a la existencia de un objeto, siempre que quede, como mínimo, un enlace fijo disponible con el objeto. Cuando se elimina el último enlace fijo, el objeto se elimina del servidor, excepto si una aplicación tiene abierto el objeto. Cada aplicación que tiene abierto el

objeto puede continuar utilizándolo hasta que la aplicación cierre el objeto. Cuando la última aplicación que utiliza el objeto lo cierra, el objeto se elimina del servidor. Un objeto no se puede abrir después de haberse eliminado el último enlace fijo.

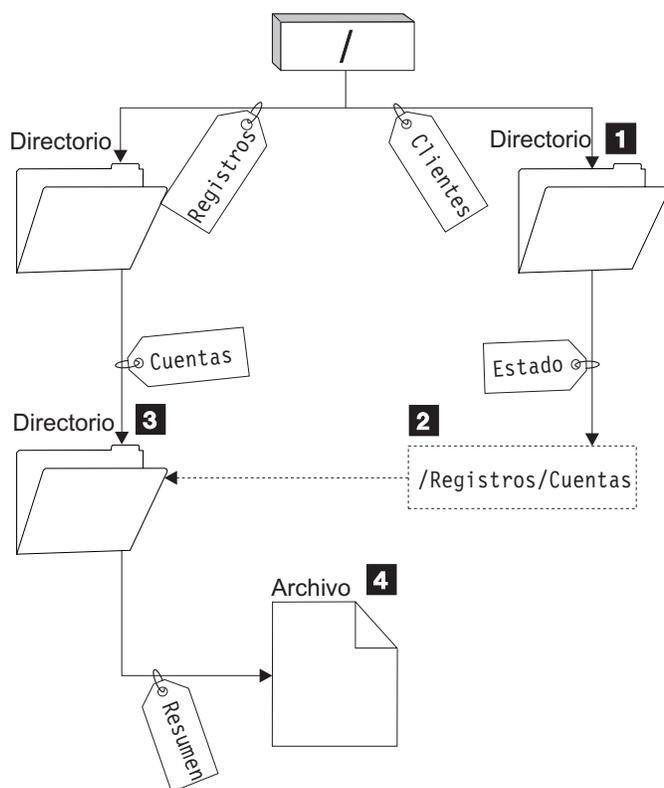
El concepto de enlace fijo también se puede aplicar al sistema de archivos de biblioteca (QSYS.LIB), a los sistemas de archivos de ASP independientes (QSYS.LIB) y al sistema de archivos de servicios de biblioteca de documentos (QDLS), pero con una restricción. Una biblioteca, en efecto, tiene un enlace fijo con cada objeto de la biblioteca. Del mismo modo, una carpeta tiene un enlace fijo con cada documento de la carpeta. No se permiten los enlaces fijos múltiples con el *mismo objeto* en los sistemas de archivos QSYS.LIB, los QSYS.LIB de ASP independientes, o los QDLS.

Un enlace fijo no puede cruzar sistemas de archivos. Por ejemplo, un directorio del sistema de archivos QOpenSys no puede incluir un enlace fijo a un objeto de los sistemas de archivos QSYS.LIB o QSYS.LIB de ASP independiente, o a un documento del sistema de archivos QDLS.

Enlace simbólico

Un enlace simbólico, también denominado enlace dinámico, es un nombre de vía de acceso contenido en un archivo. Cuando el sistema encuentra un enlace simbólico, sigue el nombre de vía de acceso que proporciona el enlace simbólico y después continúa por cualquier vía de acceso disponible que siga al enlace simbólico. Si el nombre de vía de acceso empieza con un signo /, el sistema regresa al directorio / ("raíz") y empieza a seguir la vía de acceso desde ese punto. Si el nombre de vía de acceso no empieza con un carácter /, el sistema regresa al directorio inmediatamente anterior y sigue el nombre de vía de acceso del enlace simbólico empezando en ese directorio.

Considere el ejemplo siguiente sobre cómo se puede utilizar un enlace simbólico:



RV3N068-1

Figura 8. Ejemplo de utilización de un enlace simbólico

Se selecciona una opción de menú para mostrar el estado de las cuentas de clientes. El programa que visualiza el menú utiliza el nombre de vía de acceso siguiente:

/C*l*iente/Estado/Resumen

El sistema sigue el enlace *Cliente*, que conduce a un directorio **1**, y después sigue el enlace *Estado*. El enlace *Estado* es un enlace simbólico, que contiene un nombre de vía de acceso **2**. Puesto que el nombre de vía de acceso empieza por /, el sistema vuelve al directorio (“raíz”) / y sigue los enlaces *Registros* y *Cuentas* en orden secuencial. Esta vía de acceso conduce a otro directorio **3**. Ahora el sistema completa la vía de acceso del nombre de vía de acceso que proporciona el programa. Sigue el enlace *Resumen*, que conduce a un archivo **4** que contiene los datos necesarios.

A diferencia del enlace fijo, el enlace simbólico es un objeto (de tipo *SYMLNK); puede existir sin señalar a un objeto existente. Puede utilizar un enlace simbólico, por ejemplo, para proporcionar una vía de acceso a un archivo que se ha de añadir o sustituir más adelante.

También, a diferencia del enlace fijo, el enlace simbólico puede cruzar sistemas de archivos. Por ejemplo, si está trabajando en un sistema de archivos, puede utilizar un enlace simbólico para acceder a un archivo de otro sistema de archivos. Aunque los sistemas de archivos QSYS.LIB, QSYS.LIB de ASP independientes, y QDLS no permiten crear y almacenar enlaces simbólicos, podría crear un enlace simbólico en el sistema de archivos “raíz” (/) o el sistema de archivos QOpenSys que le permitiera:

- Acceder a un miembro de archivo de base de datos de los sistemas de archivos QSYS.LIB o QSYS.LIB de ASP independiente.
- Acceder a un documento del sistema de archivos QDLS.

Consulte también el apartado “Comparación: enlace fijo y enlace simbólico”.

Comparación: enlace fijo y enlace simbólico

Al utilizar nombres de vías de acceso en los programas, se dispone de la opción de utilizar un enlace fijo o un enlace simbólico (consulte el apartado “Enlace” en la página 18). Cada tipo de enlace tiene sus ventajas y sus desventajas. Las condiciones bajo las cuales un tipo de enlace presenta ventajas respecto el otro tipo.

Tabla 1. Comparación entre enlace fijo y enlace simbólico

Elemento	Enlace fijo	Enlace simbólico
Resolución de nombre	Más rápido. Un enlace fijo contiene una referencia directa al objeto.	Más lento. Un enlace simbólico contiene un nombre de vía de acceso al objeto, que debe resolverse para localizar el objeto.
Existencia de objeto	Necesario. Un objeto debe existir para poder crear un enlace fijo con él.	Opcional. Un enlace simbólico se puede crear cuando el objeto al que hace referencia no existe.
Supresión de objeto	Restringido. Todos los enlaces fijos con un objeto deben estar desenlazados (eliminados) para suprimir el objeto.	Sin restricción. Un objeto se puede suprimir incluso aunque haya enlaces simbólicos que hagan referencia a él.
Objetos dinámicos (los atributos cambian)	Más lento. Muchos de los atributos de un objeto se almacenan en cada enlace fijo. Los cambios efectuados en un objeto dinámico, por lo tanto, son más lentos a medida que aumenta el número de enlaces fijos con el objeto.	Más rápido. Los cambios efectuados en un objeto dinámico no resultan afectados por los enlaces simbólicos.

Tabla 1. Comparación entre enlace fijo y enlace simbólico (continuación)

Elemento	Enlace fijo	Enlace simbólico
Objetos estáticos (los atributos no cambian)	Más rápido. Para un objeto estático, la resolución del nombre es el principal objeto de interés en cuanto a rendimiento. La resolución del nombre es más rápida cuando se utilizan enlaces fijos.	Más lento. La resolución del nombre es más lenta cuando se utilizan enlaces simbólicos.
Ámbito	Restringido. Los enlaces fijos no pueden cruzar sistemas de archivos.	Sin restricción. Los enlaces simbólicos pueden cruzar sistemas de archivos.

Atributos ampliados

Un atributo ampliado (EA) es la información asociada a un objeto que facilita detalles adicionales acerca del mismo. El EA se compone de un nombre, que se utiliza para hacerle referencia, y de un valor. El valor puede ser texto, datos binarios u otro tipo de datos.

Los EA de un objeto existen sólo mientras el objeto existe.

Existen muchos tipos de EA, que se pueden utilizar para contener una gran variedad de información. En particular, puede que necesite tener en cuenta los tres EA siguientes:

.SUBJECT

Descripción breve del contenido o finalidad del objeto.

.TYPE Tipo de datos del objeto. El tipo de datos puede ser texto, binario, fuente de un programa, un programa compilado u otra información.

.CODEPAGE

Página de códigos que se ha de utilizar para el objeto. La página de códigos utilizada para el objeto también se utiliza para el EA asociado al objeto.

Un punto (.) como primer carácter del nombre significa que el EA es un EA del sistema estándar (SEA), que está reservado para uso del sistema.

Diversos objetos en los distintos sistemas de archivos pueden tener o no EA. Los sistemas de archivos QSYS.LIB y QSYS.LIB de ASP independiente ofrecen soporte para tres atributos ampliados predefinidos: .SUBJECT, .TYPE y .CODEPAGE. En el sistema de archivos de servicios de biblioteca de documento (QDLS), las carpetas y los documentos pueden tener cualquier clase de EA. Algunas carpetas y documentos pueden tener EA y puede que otras no. En los sistemas de archivos "raíz" (/), sistemas abiertos (QOpenSys) y el definido por el usuario, los directorios, archivos continuos y enlaces simbólicos pueden tener EA de cualquier clase. Algunos, no obstante, puede que no tengan EA.

El mandato Trabajar con enlaces de objeto (WRKLNK) se puede utilizar para visualizar el atributo ampliado (EA) .SUBJECT de un objeto. No existe otro soporte de sistema de archivos integrado mediante el que las aplicaciones o los usuarios puedan acceder y cambiar EA. Las únicas excepciones a esta regla son los mandatos CL Visualizar un UDFS (DSPUDFS) y Visualizar información del sistema de archivos montado (DSPMFSINF), que presentan a los usuarios los atributos ampliados.

Sin embargo, los EA asociados a algunos objetos de QDLS pueden cambiarse mediante interfaces proporcionadas por el sistema de archivos jerarquizado (HFS). Consulte los apartados "Sistema de archivos de servicios de biblioteca de documentos (QDLS)" en la página 78 y "Sistema de archivos óptico (QOPT)" en la página 80, en los que hallará más información sobre estos sistemas de archivos.

Si un PC cliente está conectado al servidor iSeries a través de OS/2 o Windows, pueden utilizarse las interfaces de programación del sistema operativo respectivo (como por ejemplo DosQueryFileInfo y DosSetFileInfo) para consultar y establecer los EA de cualquier objeto de archivo. Los usuarios de OS/2 también pueden modificar los EA de un objeto del escritorio utilizando el cuaderno de valores; es decir, seleccionando Valores en el menú emergente asociado con el objeto.

Si define atributos ampliados, utilice las directrices de denominación siguientes:

- El nombre de un EA puede tener una longitud máxima de 255 caracteres.
- No utilice un punto (.) como primer carácter del nombre. Un EA cuyo nombre empieza con un punto se interpreta como un EA del sistema estándar.
- Para minimizar la posibilidad de conflictos de nombres, utilice una estructura de denominación coherente para los EA. Se aconseja el formato siguiente:

NombreEmpresaNombreProducto.Nombre_Atributo

Continuidad de nombres

Si utiliza sistemas de archivos "raíz" (/), QOpenSys y definido por el usuario, puede aprovechar el soporte del sistema que garantiza que los caracteres de los nombres de objeto siguen siendo los mismos. Ello ocurre también cuando se utilizan los sistemas de archivos entre iSeries server y dispositivos conectados que tienen esquemas distintos de codificación de caracteres (páginas de códigos). El servidor almacena los caracteres de los nombres en un formato de 16 bits conocido como UCS2 nivel 1 (también llamado **Unicode**) para los directorios *TYPE1 y UTF-16 para los directorios *TYPE2. Consulte el apartado Directorios *TYPE2 para obtener más información sobre los formatos de directorios. UCS2 nivel 1 y UTF-16 son subconjuntos del estándar ISO 10646. Cuando se utiliza el nombre, el sistema convierte el formato almacenado de los caracteres en la representación de los caracteres adecuada de la página de códigos que se está utilizando. Los nombres de los atributos ampliados asociados a cada objeto también se manejan del mismo modo.

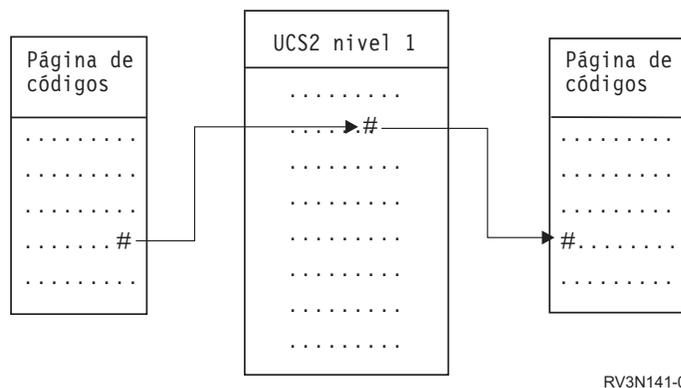


Figura 9. Mantener los caracteres igual en los esquemas de codificación

Este soporte facilita la interacción con un servidor desde dispositivos que utilizan páginas de códigos diferentes. Por ejemplo, los usuarios de PC pueden acceder a un archivo de un servidor iSeries utilizando el mismo nombre de archivo, aunque los PC no tengan la misma página de códigos que el servidor. El servidor maneja automáticamente la conversión de una página de códigos a otra. Obviamente, el dispositivo debe utilizar una página de códigos que contenga los caracteres utilizados en el nombre.

Capítulo 3. Acceda al sistema de archivos integrado utilizando la interfaz del sistema tradicional

Todas las interfaces de usuario (como por ejemplo menús, mandatos y pantallas) que se utilizan para trabajar con las bibliotecas, objetos, archivos de base de datos, carpetas y documentos del sistema siguen funcionando igual que antes de la incorporación del sistema de archivos integrado. Dichas interfaces, sin embargo, no se pueden utilizar para trabajar con los archivos continuos, directorios y otros objetos soportados por el sistema de archivos integrado.

Se proporciona un conjunto separado de interfaces de usuario para el sistema de archivos integrado. Estas interfaces se pueden utilizar con los objetos de cualquier sistema de archivos al que se pueda acceder a través de los directorios del sistema de archivos integrado.

Puede interactuar con los directorios y objetos del sistema de archivos integrado desde el servidor por medio de menús y pantallas o mediante mandatos del lenguaje de control (CL). Además, también puede utilizar interfaces de programas de aplicación (API) para aprovechar las ventajas de los archivos continuos, directorios y otros soportes del sistema de archivos integrado.

También puede interactuar con el sistema de archivos integrado mediante el Navigator de iSeries, una interfaz de usuario gráfica utilizada para gestionar y administrar el servidor desde el escritorio Windows.

Existen varias formas de interactuar con el sistema de archivos integrado:

Utilizando API

Las interfaces de programa de aplicación que realizan operaciones en los directorios y archivos continuos del sistema de archivos integrado tienen el formato de funciones de lenguaje C.

Utilizando mandatos CL

Los mandatos CL pueden funcionar en archivos y otros objetos en cualquier sistema de archivos al que pueda accederse a través de la sistema de archivos integrado.

Utilizando los menús y pantallas del iSeries

Se pueden realizar operaciones con archivos y otros objetos del sistema de archivos integrado utilizando el conjunto de menús y pantallas que proporciona el servidor.

Utilizando iSeries Navigator

iSeries Navigator es la interfaz gráfica de usuario que permite gestionar y administrar servidores desde el escritorio de Windows.

Utilizando un PC

Si el PC está conectado a un iSeries server, podrá interactuar con los directorios y objetos del sistema de archivos integrado como si estuvieran almacenados en el PC.

Realizar operaciones utilizando los menús y pantallas del iSeries

Se pueden realizar operaciones con archivos y otros objetos del sistema de archivos integrado utilizando el conjunto de menús y pantallas que proporciona el servidor. Para ver los menús del sistema de archivos integrado:

1. Inicie una sesión en el servidor.
2. Pulse **Intro** para continuar.
3. Seleccione la opción **Archivos, bibliotecas y carpetas** del menú principal del iSeries.
4. Seleccione la opción **Sistema de archivos integrado** del menú Archivos, bibliotecas y carpetas.

A partir de este punto, ya podrá trabajar con mandatos de directorios, de objetos o de seguridad en el sistema de archivos integrado, según le interese. No obstante, si conoce el mandato CL a utilizar, puede escribirlo en la línea de mandatos en la parte inferior de la pantalla y pulsar la tecla **Intro**, lo que le permitirá saltarse las opciones del menú.

Además, se puede acceder al sistema de archivos integrado desde cualquier menú del servidor si se siguen estos pasos:

1. Escriba `GO DATA` en cualquier línea de mandatos para visualizar el menú Archivos, bibliotecas y carpetas.
2. Seleccione la opción Sistema de archivos integrado.

Para ver un menú de los mandatos del sistema de archivos de red, escriba `GO CMDNFS` en una línea de mandatos. Para ver un menú de mandatos del sistema de archivos definido por el usuario, introduzca `GO CMDUDFS` en cualquier línea de mandatos.

En los menús del sistema de archivos integrado, se pueden solicitar pantallas donde se pueden realizar las operaciones siguientes:

- Crear, convertir y eliminar un directorio
- Visualizar y cambiar el nombre del directorio actual
- Añadir, visualizar, cambiar y eliminar enlaces de objetos
- Copiar, mover y renombrar objetos
- Reservar y reincorporar objetos
- Salvar y restaurar objetos
- Visualizar y cambiar propietarios de objetos y autorizaciones de usuarios
- Copiar datos entre archivos continuos y miembros de archivo de base de datos
- Crear, suprimir y visualizar el estado de los sistemas de archivos definidos por el usuario
- Exportar sistemas de archivos de un servidor
- Montar y desmontar sistemas de archivos en un cliente

Determinados sistemas de archivos no disponen de soporte para todas estas operaciones. Para conocer las restricciones de un sistema de archivos concreto, consulte el apartado “Sistemas de archivos en el sistema de archivos integrado” en la página 4.

Consulte el siguiente apartado para obtener más información sobre los menús y pantallas del sistema de archivos integrado.

- Reglas sobre el nombre de ruta para mandatos CL y pantallas

Realizar operaciones mediante mandatos CL

Todas las operaciones que se pueden realizar a través de los menús y pantallas del sistema de archivos integrado (consulte el apartado “Realizar operaciones utilizando los menús y pantallas del iSeries” en la página 25) también pueden efectuarse entrando mandatos del lenguaje de control (CL). Estos mandatos pueden funcionar en archivos y otros objetos en cualquier sistema de archivos al que pueda accederse a través de la interfaz del sistema de archivos integrado.

La Tabla 1 resume los mandatos del sistema de archivos integrado. Si desea obtener más información sobre los mandatos CL específicamente relacionados con los sistemas de archivos definidos por el usuario, el sistema de archivos de red y los sistemas de archivos montados en general, puede consultar los temas “Sistema de archivos definido por el usuario (UDFS)” en la página 68 y “Sistema de archivos de red (NFS)” en la página 91. Cuando un mandato efectúa la misma operación que un mandato de OS/2 o de DOS, se proporciona un alias (un nombre alternativo de mandato) para comodidad de los usuarios de OS/2 y DOS.

Tabla 2. Mandatos del sistema de archivos integrado

Mandato	Descripción	Alias
ADDLNK	Añadir enlace. Añade un enlace entre un directorio y un objeto.	
ADDMFS	Añadir sistema de archivos montado. Pone los sistemas de archivos de servidor remoto exportados en directorios de cliente locales.	MOUNT
APYJRNCHG ²	Aplicar cambios registrados por diario. Utiliza entradas de diario para aplicar los cambios ocurridos desde que se guardó un objeto registrado por diario o para aplicar los cambios hasta un punto especificado.	
CHGATR	Cambiar atributo. Cambia el atributo de un único objeto, grupo de objetos o árbol de directorios para el que se quiere cambiar el atributo del directorio, su contenido y el contenido de todos sus subdirectorios.	
CHGAUD	Cambiar valor de auditoría. Activa o desactiva la auditoría de un objeto.	
CHGAUT	Cambiar autorización. Otorga autorización específica sobre un objeto a un usuario o a un grupo de usuarios.	
CHGCURDIR	Cambiar directorio actual. Cambia el directorio que se ha de utilizar como el directorio actual.	CD, CHDIR
CHGNFSEXP	Cambiar exportación de sistema de archivos de red. Añade árboles de directorios o los elimina de la tabla de exportación que se exporta a los clientes NFS.	EXPORTFS
CHGOWN	Cambiar propietario. Transfiere la propiedad del objeto de un usuario a otro.	
CHGPGP	Cambiar grupo primario. Cambia el grupo primario de un usuario a otro.	
CHKIN	Reincorporar. Reincorpora un objeto que se había reservado antes.	
CHKOUT	Reservar. Reserva un objeto, con lo que se impide que otros usuarios lo cambien.	
CPY	Copiar. Copia un objeto o un grupo de objetos.	COPY
CPYFRMSTMF	Copiar de archivo continuo. Copia los datos de un archivo continuo en un miembro de archivo de base de datos.	
CPYTOSTMF	Copiar a archivo continuo. Copia los datos de un miembro de archivo de base de datos en un archivo continuo.	
CRTDIR	Crear directorio. Añade un nuevo directorio al sistema.	MD, MKDIR
CRTUDFS	Crear UDFS. Crea un sistema de archivos definido por el usuario.	
CVTDIR	Conversión de directorio. Proporciona información sobre la conversión de directorios del sistema de archivo integrado del formato *TYPE1 al formato *TYPE2, o realizar una conversión.	
CVTRPCSRC	Convertir fuente RPC. Genera código C a partir de un archivo de entrada escrito en lenguaje RPC (Remote Procedure Call).	RPCGEN
DLTUDFS	Suprimir UDFS. Suprime un sistema de archivos definido por el usuario.	
DSPAUT	Visualizar autorización. Muestra una lista de usuarios autorizados de un objeto y sus autorizaciones sobre el mismo.	
DSPCURDIR	Visualizar directorio actual. Muestra el nombre del directorio actual.	

Tabla 2. Mandatos del sistema de archivos integrado (continuación)

Mandato	Descripción	Alias
DSPLNK	Visualizar enlaces de objetos. Muestra una lista de objetos de un directorio y proporciona opciones para visualizar información acerca de los objetos.	
DSPF	Visualizar archivo continuo. Visualiza un archivo continuo o un archivo de base de datos.	
DSPMFSINF	Visualizar información de sistema de archivos montado. Visualiza información acerca de un sistema de archivos montado.	STATFS
DSPUDFS	Visualizar UDFS. Visualiza un sistema de archivos definido por el usuario.	
EDTF	Editar archivo continuo. Edita un archivo continuo o un archivo de base de datos.	
ENDJRN ²	Finalizar registro por diario. Finaliza el registro por diario de los cambios realizados a un objeto o lista de objetos.	
ENDNFSSVR	Finalizar servidor de sistema de archivos de red. Finaliza uno o todos los daemons del sistema de archivos de red del servidor y del cliente.	
ENDRPCBIND	Finalizar daemon enlazador RPC. Finaliza el daemon RPCBind de RPC (Remote Procedure Call).	
MOV	Mover. Mueve un objeto a otro directorio.	MOVE
RLSIFSLCK	Liberar bloqueos de sistema de archivos integrado. Libera todos los bloqueos de rango de byte de NFS mantenidos por un cliente o sobre un objeto.	
RMVDIR	Eliminar directorio. Elimina un directorio del sistema.	RD, RMDIR
RMVLNK	Eliminar enlace. Elimina el enlace con un objeto.	DEL, ERASE
RMVMFS	Eliminar sistema de archivos montado. Elimina los sistemas de archivos de servidor remoto exportados de los directorios de cliente locales.	UNMOUNT
RNM	Redenominar. Cambia el nombre de un objeto de un directorio.	REN
RPCBIND	Iniciar daemon enlazador RPC. Inicia el daemon RPCBind de RPC (Remote Procedure Call).	
RST	Restaurar. Copia un objeto o un grupo de objetos desde un dispositivo de copia de seguridad al sistema.	
RTVCURDIR	Recuperar directorio actual. Recupera el nombre del directorio actual y lo pone en una variable especificada (utilizada en programas CL).	
SAV	Salvar. Copia un objeto o grupo de objetos del sistema a un dispositivo de copia de seguridad.	
SNDJRNE ²	Enviar entrada de diario. Añade entradas de diario de usuario, opcionalmente asociadas con un objeto registrado por diario, a un receptor de diario.	
STRJRN ²	Iniciar registro por diario. Inicia el registro por diario de los cambios (realizados a un objeto o lista de objetos) para un diario concreto.	
STRNFSSVR	Iniciar servidor de sistema de archivos de red. Inicia uno o todos los daemons de NFS del servidor y del cliente.	
WRKAUT	Trabajar con autorizaciones. Muestra una lista de usuarios y sus autorizaciones y ofrece opciones para añadir un usuario, cambiar una autorización de usuario o eliminar un usuario.	

Tabla 2. Mandatos del sistema de archivos integrado (continuación)

Mandato	Descripción	Alias
WRKLNK	Trabajar con enlaces de objetos. Muestra una lista de los objetos de un directorio y ofrece opciones para realizar acciones sobre los mismos.	
WRKOBJOWN ¹	Trabajar con objetos por propietario. Muestra una lista de los objetos que son propiedad de un perfil de usuario y ofrece opciones para realizar acciones sobre los mismos.	
WRKOBJPGP ¹	Trabajar con objetos por grupo primario. Muestra una lista de los objetos controlados por un grupo primario y ofrece opciones para realizar acciones sobre los mismos.	

Nota:

1. Los mandatos WRKOBJOWN y WRKOBJPGP pueden visualizar todos los tipos de objeto, pero puede ser que no funcionen completamente en todos los sistemas de archivos.
2. Consulte gestión por diarios del Information Center del iSeries para obtener más información.

Consulte estos temas para obtener más información sobre los mandatos CL del sistema de archivos integrado y restricciones sobre la utilización de estos mandatos en determinados sistemas de archivos:

- Sistemas de archivos en el sistema de archivos integrado
- Reglas sobre el nombre de ruta para mandatos CL y pantallas
- El tema CL del iSeries Information Center

Reglas sobre el nombre de ruta para mandatos CL y pantallas

Cuando se utiliza un mandato o una pantalla del sistema de archivos integrado para trabajar con un objeto, el objeto se identifica suministrando el nombre de la vía de acceso. A continuación se proporciona un resumen de las reglas que hay que tener en cuenta al especificar nombres de vías de acceso. El término **objeto** en estas reglas hace referencia a directorios, archivos, enlaces o cualquier otro objeto.

- Los nombres de objetos deben ser exclusivos en cada directorio.
- El nombre de vía de acceso que se pasa a un mandato CL de un sistema de archivos integrado debe estar representado en el CCSID actualmente en vigor para el trabajo. Si el CCSID del trabajo es 65535, el nombre de vía de acceso debe estar representado en el CCSID por omisión del trabajo. Como las series de texto suelen estar codificadas en CCSID 37, es necesario convertir los nombres de vía de acceso que no pueden modificarse al CCSID del trabajo antes de pasar la vía de acceso al mandato.
- Cuando se escriben en la línea de mandatos, los nombres de vías de acceso deben estar encerrados entre apóstrofes ('). Cuando los nombres de vías de acceso se escriben en las pantallas, los apóstrofes son opcionales. Si el nombre de vía de acceso contiene series entre apóstrofes, sin embargo, los signos delimitadores ' ' también deben incluirse.
- Los nombres de vías de acceso se escriben de izquierda a derecha, empezando por el directorio de nivel superior y acabando por el nombre del objeto sobre el que se va a ejecutar el mandato. El nombre de cada componente de la vía de acceso se separa con una barra inclinada (/) o una barra inclinada invertida (\); por ejemplo:
 'Dir1/Dir2/Dir3/ArchUsr'
 o
 'Dir1\Dir2\Dir3\ArchUsr'
- Los caracteres / y \ y los nulos no se pueden utilizar en los componentes individuales del nombre de vía de acceso (debido a que los caracteres / y \ se utilizan como separadores). Los mandatos no cambian las minúsculas por mayúsculas. El nombre puede cambiarse o no por mayúsculas,

dependiendo de si el sistema de archivos que contiene el objeto distingue entre mayúsculas y minúsculas y de si el objeto se está creando o buscando.

- La longitud del nombre de objeto está limitada por el sistema de archivos en que está el objeto y por la longitud máxima que puede tener una serie de mandato. Los mandatos aceptan nombres de objeto con una longitud máxima de 255 caracteres y nombres de vías de acceso con una longitud máxima de 5000 caracteres.

Véase Sistemas de archivos en el sistema de archivos integrado para obtener información sobre los límites del nombre de vía de acceso en cada sistema de archivos.

- Un carácter / o \ al principio de un nombre de vía de acceso significa que la vía de acceso empieza en el directorio de mayor nivel, el directorio “raíz” (/); por ejemplo:

```
'/Dir1/Dir2/Dir3/ArchUsr'
```

- Si el nombre de vía de acceso no empieza con un carácter / o \, la vía de acceso se presupone que empieza en el directorio actual del usuario que entra el mandato; por ejemplo:

```
'MiDir/MiArch'
```

donde MiDir es un subdirectorio del directorio actual del usuario.

- Una tilde (~) seguida de una barra inclinada (o barra inclinada invertida) al principio de un nombre de vía de acceso significa que la vía de acceso empieza en el directorio inicial del usuario que entra el mandato; por ejemplo:

```
'~/UsrDir/UsrObj'
```

- Una tilde (~) seguida de un nombre de usuario y de una barra inclinada (o barra inclinada invertida) al principio de un nombre de vía de acceso significa que la vía de acceso empieza en el directorio inicial del usuario identificado por el nombre de usuario; por ejemplo:

```
'~nombre-usuario/UsrDir/UsrObj'
```

- En algunos mandatos, el asterisco (*) o el interrogante (?) se pueden utilizar en el último componente de un nombre de vía de acceso para buscar patrones de nombres. El signo * indica al sistema que busque los nombres con cualquier número de caracteres en la posición del carácter *. El carácter ? indica al sistema que busque los nombres que tengan un solo carácter en la posición del carácter ?. El ejemplo siguiente busca todos los objetos cuyos nombres empiezan por *d* y terminan por *txt*.

```
'/Dir1/Dir2/Dir3/d*txt'
```

El ejemplo siguiente busca los objetos cuyos nombres empiezan con *d* seguida de cualquier carácter y terminan con *txt*:

```
'/Dir1/Dir2/Dir3/d?txt'
```

- Para evitar confusiones con los valores especiales del servidor iSeries, los nombres de vía de acceso no pueden comenzar con un único carácter de asterisco (*). Para llevar a cabo una comparación de patrones al principio de un nombre de vía de acceso, utilice dos asteriscos (**); por ejemplo:

```
'**.archivo'
```

Nota: este caso sólo se aplica a los nombres de vía de acceso relativos donde no hay otros caracteres antes del asterisco (*).

- Cuando se trabaja con objetos del QSYS.LIB, los nombres de componentes deben tener el formato *name.object-type*; por ejemplo:

```
'/QSYS.LIB/PAY.LIB/TAX.FILE'
```

Consulte el apartado “Sistema de archivos de biblioteca (QSYS.LIB)” en la página 72 para conocer más detalles.

- Cuando se trabaja con objetos del QSYS.LIB, los nombres de componentes deben tener el formato *name.object-type*; por ejemplo:

```
'/asp_name/QSYS.LIB/PAYDAVE.LIB/PAY.FILE'
```

Consulte el apartado “QSYS.LIBde ASP independiente” en la página 75 para conocer más detalles.

- El nombre de vía de acceso debe estar delimitado por conjuntos adicionales de apóstrofos (') o comillas (") si se utiliza alguno de los caracteres siguientes en un nombre de componente:
 - Asterisco (*)
 - Interrogante (?)
 - Apóstrofo (')
 - Comillas (")
 - Tilde (~), si se utiliza como el primer carácter del primer nombre de componente del nombre de la vía de acceso (si se utiliza en cualquier otra posición, la tilde se interpreta exactamente igual que otro carácter)

Por ejemplo:

```
"/Dir1/Dir/A*Antonio"
```

o

```
'''/Dir1/Dir/A*Antonio'''
```

Esta práctica no es aconsejable debido a que el significado del carácter en una serie de mandato se puede confundir y hay más probabilidades de entrar incorrectamente la serie de mandato.

- No utilice signos de dos puntos (:) en los nombres de vía de acceso. Tienen un significado especial dentro del sistema.
- El soporte de proceso para los mandatos y las pantallas de usuario asociadas no reconoce elementos de código por debajo del hexadecimal 40 como caracteres que puedan utilizarse en series de mandato o en las pantallas. Si se utilizan estos elementos de código, deben entrarse en representación hexadecimal, como por ejemplo:

```
crtDir dir(X'02')
```

Por consiguiente, la utilización de elementos de código por debajo de 40 hexadecimal en nombres de vía de acceso no es recomendable. Esta restricción se aplica sólo a mandatos y pantallas asociadas, no a las API (consulte el apartado "Realizar operaciones utilizando API" en la página 49).

Si desea conocer las restricciones sobre la utilización de un mandato determinado, puede consultar la ayuda del mandato o la sección Lenguaje de control (CL) del iSeries Information Center.

Realizar operaciones mediante un PC

Si el PC está conectado a un iSeries server, podrá interactuar con los directorios y objetos del sistema de archivos integrado como si estuvieran almacenados en el PC. Podrá copiar objetos de un directorio a otro utilizando la posibilidad de arrastrar y soltar del Explorador de Windows. Si fuera necesario, es posible copiar un objeto del servidor al PC seleccionándolo en la unidad del servidor y arrastrándolo a la unidad del PC.

Todos los objetos que se copian entre un iSeries server y PC utilizando la interfaz de Windows pueden convertirse automáticamente de EBCDIC y ASCII. EBCDIC es el código de intercambio decimal ampliado codificado en binario y ASCII es el código estándar norteamericano para el intercambio de información. iSeries Access puede configurarse de forma que efectúe automáticamente esta conversión, y puede especificarse incluso que dicha conversión se efectúe en archivos que tengan una extensión específica. En la V4R4 del OS/400, también puede configurarse NetServer del iSeries para que realice la conversión en los archivos.

Según el tipo de objeto, pueden utilizarse las interfaces del PC, y posiblemente las aplicaciones del PC, para trabajar con él. Por ejemplo, un archivo continuo que contiene texto se podría editar con un editor de PC.

Si está conectado a un iSeries server a través de un PC, el sistema de archivos integrado pone a disposición del PC los directorios y objetos del servidor. Los PC pueden trabajar con los archivos del sistema de archivos integrado utilizando clientes de compartimiento de archivos que se encuentran incorporados en los sistemas operativos Windows, un cliente FTP, o el iSeries Navigator (parte del iSeries Access). Su PC utiliza clientes de compartimiento de archivos de Windows para el acceso a iSeries NetServer, que se ejecuta en el iSeries server.

Transferencia de archivos utilizando FTP

El cliente de FTP le permite transferir archivos que se encuentran en el iSeries server, inclusive aquellos de los sistemas de archivos "raíz" (/), QSYS.LIB, QSYS.LIB de ASP independiente, QOpenSys, QOPT, y QFileSvr.400. También le permite transferir carpetas y documentos en el sistema de archivos de servicios de biblioteca (QDLS).

Trabajar con archivos utilizando el iSeries Navigator

iSeries Access incluye iSeries Navigator, que se conecta a un iSeries server y pone el sistema de archivos integrado a disposición del PC. iSeries Navigator es la interfaz gráfica de usuario que permite gestionar y administrar su iSeries server desde el escritorio de Windows.

Trabajar con archivos utilizando el NetServer del iSeries

NetServer del iSeries es un componente de OS/400 que permite que el compartimiento de archivos e impresoras que incorporan los clientes Windows funcione con el servidor.

Nota: La nueva versión del iSeries Access se basa completamente en NetServer para acceder al sistema de archivos integrado. El soporte de NetServer sólo está disponible para conexiones TCP/IP con un iSeries server que ejecute OS/400 V4R2 y posteriores.

Transferencia de archivos utilizando FTP

El cliente de FTP (protocolo de transferencia de archivos) le permite transferir archivos que se encuentran en el iSeries server, inclusive aquellos de los sistemas de archivos "raíz", QOpenSys, QSYS.LIB, QSYS.LIB de ASP independiente, QOPT, y sistemas de archivos QFileSvr.400. También le permite transferir carpetas y documentos en el sistema de archivos de servicios de biblioteca (QDLS). El cliente FTP puede ejecutarse interactivamente en modo de lotes sin supervisión en el que los submandatos de cliente se leen de un archivo y las respuestas a estos submandatos se escriben a un archivo. También incluye otras funciones para la manipulación de archivos en el servidor.

Para transferir archivos entre cualquiera de los sistemas de archivos siguientes, puede utilizar el soporte de FTP:

- "sistema de archivos raíz" (/)
- Sistema de archivos de sistemas abiertos (QOpenSys)
- Sistema de archivos de biblioteca (QSYS.LIB)
- sistema de archivos QSYS.LIB de ASP independiente
- Sistema de archivos de servicios de biblioteca de documentos (QDLS)
- Sistema de archivos óptico (QOPT)
- Sistema de archivos de red (NFS)
- Sistema de archivos de NetWare (QNetWare)
- Sistema de archivos de Windows NT Server (QNTC)

No obstante, debe tener presentes las limitaciones siguientes:

- El sistema de archivos integrado limita el soporte de FTP a la transferencia de datos de archivos únicamente. No se puede utilizar FTP para transferir datos de atributos.
- Los sistemas de archivos QSYS.LIB y QSYS.LIB de ASP independiente limitan el soporte de FTP a miembros de archivo físico, miembros de archivo físico fuente y archivos de salvar. No se puede utilizar

FTP para transferir otros tipos de objeto como, por ejemplo, programas (*PGM). No obstante, se puede guardar dichos tipos de objeto en un archivo de salvar, transferirlo y, a continuación, restaurar los objetos.

Para obtener información sobre FTP, consulte los siguientes apartados de la categoría **Trabajo en red** del iSeries Information Center:

- FTP
- Transferencia de archivos con FTP

Trabajar con archivos utilizando el NetServer de iSeries

El soporte de iSeries para el Entorno de Red de Windows (iSeries NetServer) es una función del sistema operativo de IBM/400 versión 5 (OS/400) que permite a los clientes Windows el acceso a vías de acceso del directorio compartido y las colas de salida compartidas del OS/400. El NetServer del iSeries permite a los PC que ejecutan software de Windows el acceso sin problemas a datos e impresoras gestionados por su iSeries. Los clientes de PC de una red utilizan las funciones que permiten compartir archivos e impresoras que se incluyen en sus sistemas operativos. Ello significa que no es necesario instalar ningún software adicional en su PC para utilizar el NetServer del iSeries.

Los clientes LINUX con el software de cliente Samba instalado también pueden acceder sin problemas a datos e impresoras a través del NetServer del iSeries. Los sistemas de archivos Samba (smbfs) pueden montarse desde el NetServer del iSeries de un modo similar al utilizado para montar sistemas de archivos NFS desde el iSeries. Para más información, consulte el apartado iSeries NetServer del iSeries Information Center.

El compartimiento de archivos NetServer del iSeries es una vía de acceso de directorios que el NetServer del iSeries comparte con los clientes en la red del iSeries. Un compartimiento de archivos puede consistir en cualquier directorio del sistema de archivos integrado en el iSeries. Para poder trabajar con el compartimiento de archivos utilizando el NetServer del iSeries debe crear un compartimiento de archivos del NetServer del iSeries, y, si es necesario, modificar un compartimiento de archivos del NetServer del iSeries utilizando el iSeries Navigator.

Para acceder a los compartimientos de archivos del sistema de archivos integrado utilizando el NetServer del iSeries:

1. Pulse el botón derecho en **Inicio**, y seleccione **Explorar** para abrir el Explorador de Windows en su PC Windows.
2. Abra el menú **Herramientas**, y seleccione **Correlacionar unidad de red**.
3. Seleccione la letra de una unidad libre para el compartimiento de archivos (como la unidad I:\).
4. Entre el nombre de un compartimiento de archivos NetServer del iSeries. Por ejemplo, puede introducir la siguiente sintaxis: **\\QSYSTEM1\Sharename**

Nota: QSYSTEM1 es el nombre de sistema del NetServer del iSeries, y Nombre compartido es el nombre del compartimiento de archivos que desea utilizar.

5. Pulse en Aceptar.

Nota: Si se conecta mediante NetServer del iSeries, el nombre del servidor puede ser distinto del nombre utilizado por iSeries Access. Por ejemplo, el nombre del NetServer del iSeries puede ser QAS400X, y la vía de acceso para trabajar con los archivos puede ser \\QAS400X\QDLS\MYFOLDER.FLR\MYFILE.DOC. Sin embargo, el nombre del iSeries Access puede ser AS400X, y la vía de acceso para trabajar con los archivos puede ser \\AS400X\QDLS\MYFOLDER.FLR\MYFILE.DOC.

Puede seleccionar qué directorios compartir con la red utilizando el NetServer del iSeries. Estos directorios aparecerán como el primer nivel bajo el nombre del servidor. Por ejemplo, si el administrador

comparte el directorio /home/fred con el nombre fredsdír, un usuario podrá acceder a dicho directorio desde el PC con el nombre \\QAS400X\FRESDIR, o desde un cliente LINUX con el nombre //qas400x/fredsdír.

El sistema de archivos "raíz" (/) proporciona un rendimiento mucho mejor sirviendo archivos de PC que otros sistemas de archivos del iSeries. Quizás desee mover archivos al sistema de archivos "raíz" (/). Consulte consideraciones para trasladar objetos a otro sistema de archivos para obtener más información.

Para obtener más información sobre NetServer del iSeries y los compartimientos de archivos, consulte los siguientes temas de la categoría **Trabajo en red** del iSeries Information Center:

- Netserver del iSeries
- Compartimientos de archivos NetServer de iSeries
- Acceso al compartimiento de archivos NetServer de iSeries con un cliente de PC Windows

Trasladar objetos a otro sistema de archivos

Antes de utilizar el sistema de archivos integrado para trasladar objetos de un sistema de archivos a otro, consulte "Consideraciones sobre el traslado de objetos a otro sistema de archivos".

Para trasladar objetos a otro sistema de archivos, lleve a cabo los pasos siguientes:

1. Salve una copia de todos los objetos que piensa trasladar.

Tener una copia de seguridad permite restaurar los objetos en el sistema de archivos original, si resulta que las aplicaciones no pueden acceder a los objetos en el sistema de archivos a donde los ha trasladado.

Nota: no se pueden salvar objetos de un sistema de archivos y restaurarlos en otro.

2. Cree en el sistema de archivos los directorios a los que se quiere trasladar los objetos, utilizando el mandato Crear directorio (CRTDIR).

Debe examinar detenidamente los atributos del directorio donde están actualmente los objetos para determinar si querrá duplicar esos atributos en los directorios que cree. Por ejemplo, el usuario que cree el directorio será su propietario, en vez del usuario que era propietario del directorio antiguo. Puede ser conveniente transferir la propiedad del directorio después de haberlo creado, si el sistema de archivos tiene soporte para establecer el propietario de un directorio.

3. Traslade los archivos al sistema de archivos elegido utilizando el mandato Mover (MOV).

El uso de MOV es aconsejable debido a que protege la propiedad de los objetos, si el sistema de archivos permite establecer dicha propiedad. Sin embargo, se puede utilizar el mandato Copiar (CPY) para proteger la propiedad de los objetos utilizando el parámetro OWNER(*KEEP). Recuerde que esto funcionará solamente para los sistemas de archivos que permitan establecer la propiedad de un objeto. Observe que cuando se utiliza MOV o CPY:

- Los atributos pueden no coincidir y pueden descartarse.
- Los atributos ampliados pueden descartarse.
- Las autorizaciones pueden no ser equivalentes y descartarse.

Esto significa que si decide devolver el objeto a su sistema de archivos original, puede que no quiera volver a trasladarlo o copiarlo debido a los atributos y autorizaciones que se han descartado. El método más seguro de devolver un objeto es restaurar una versión salvada del objeto.

Consideraciones sobre el traslado de objetos a otro sistema de archivos

Cada sistema de archivos tiene sus propias características exclusivas. Sin embargo, trasladar objetos a un sistema de archivos distinto puede significar perder las ventajas del sistema de archivos donde están almacenados actualmente los objetos. Puede que le interese trasladar objetos de un sistema de archivos

a otro para aprovechar esas características. Antes de trasladar objetos a otro sistema de archivos, deberá familiarizarse con los sistemas de archivos del sistema de archivos integrado y sus características. Si desea obtener más información, puede consultar el apartado “Sistemas de archivos en el sistema de archivos integrado” en la página 4.

También deberá tener en cuenta los siguientes puntos:

- ¿Está utilizando aplicaciones que aprovechan las ventajas del sistema de archivos en el que están actualmente los objetos?

Algunos sistemas de archivos soportan interfaces que no forman parte del soporte del sistema de archivos integrado. Es posible que las aplicaciones que utilizan esas interfaces dejen de poder acceder a los objetos que se han trasladado a otro sistema de archivos. Por ejemplo, los sistemas de archivos QDLS y QOPT admiten el sistema de archivos jerárquico (HFS). Las API y los mandatos funcionan con objetos de tipo carpeta y documento. No se pueden utilizar esas interfaces con objetos que están en otros sistemas de archivos.

- Para usted, ¿cuáles son las características más importantes de los objetos?

No todas las características están soportadas en todos los sistemas de archivos. Por ejemplo, los sistemas de archivos QSYS.LIB o QSYS.LIB de ASP independiente dan soporte sólo para almacenar y recuperar unos cuantos atributos ampliados, mientras que los sistemas de archivos “raíz” (/) y QOpenSys dan soporte para almacenar y recuperar todos los atributos ampliados. Por consiguiente, QSYS.LIB y QSYS.LIB de ASP independiente no son aconsejables para almacenar objetos que tengan atributos ampliados. QDLS soporta muchos atributos “ofimáticos”, pero otros sistemas no. Por lo tanto, QDLS es un buen lugar para conservar los documentos ofimáticos.

Los archivos de PC que están en QDLS son buenos candidatos para ser trasladados. La mayoría de aplicaciones de PC deben ser capaces de seguir trabajando con los archivos de PC que se trasladan desde QDLS a otros sistemas de archivos. Los sistemas de archivos “raíz” (/), QOpenSys, QNetWare y QNTC son buenas elecciones para almacenar dichos archivos de PC. Puesto que disponen de soporte para muchas de las características del sistema de archivos de OS/2, estos sistemas de archivos pueden proporcionar un acceso más rápido a los archivos.

Directorios que proporciona el sistema de archivos integrado

El sistema de archivos integrado crea los siguientes directorios cuando se reinicia el sistema, si todavía no existen:

/tmp El directorio /tmp proporciona a las aplicaciones un lugar donde almacenar archivos temporales. Este directorio es un subdirectorio del directorio “raíz” (/), por lo tanto su nombre de vía de acceso es /tmp.

Una vez que una aplicación ha colocado un archivo en el directorio /tmp, el archivo permanece allí hasta que la aplicación o el usuario lo eliminan. El sistema no elimina automáticamente archivos de /tmp ni ejecuta ningún otro proceso especial para los archivos que se encuentran en /tmp.

Puede utilizar los mandatos y pantallas de usuario que soportan el sistema de archivos integrado para gestionar el directorio /tmp y sus archivos. Por ejemplo, puede utilizar la pantalla Trabajar con enlaces de objetos o el mandato WRKLNK para copiar, eliminar o renombrar el directorio /tmp o los archivos del directorio. A todos los usuarios se les otorga la autorización *ALL sobre el directorio, lo que significa que pueden efectuar la mayoría de acciones válidas sobre el directorio.

Una aplicación puede utilizar las interfaces de programa de aplicación (API) que soportan el sistema de archivos integrado para gestionar /tmp y a sus archivos (consulte “Realizar operaciones utilizando API” en la página 49). Por ejemplo, el programa de aplicación puede eliminar un archivo del directorio /tmp utilizando la API unlink().

Si se elimina /tmp, se volverá a crear automáticamente durante el próximo reinicio del sistema.

/home Los administradores del sistema utilizan el directorio /home para almacenar un directorio independiente para cada usuario. El administrador del sistema establece con frecuencia que el

directorio inicial que está asociado con el perfil de usuario sea el directorio del usuario en /home, por ejemplo/home/john. Para obtener más información, consulte el apartado “Directorio actual y directorio de inicio” en la página 8.

/etc El directorio /etc almacena archivos de administración, de configuración y otros archivos del sistema.

/usr El directorio /usr incluye subdirectorios que contienen información utilizada por el sistema. Generalmente, los archivos de /usr no cambian a menudo.

/usr/bin

El directorio /usr/bin contiene los programas de utilidad estándares.

/QIBM El directorio /QIBM es el directorio del sistema y se facilita con el sistema.

/QIBM/ProdData

El directorio /QIBM/ProdData es un directorio del sistema que se utiliza para datos de producto de Programa bajo Licencia.

/QIBM/UserData

El directorio /QIBM/UserData es un directorio del sistema que se utiliza para datos de usuario de Programa bajo Licencia, como archivos de configuración.

/QOpenSys/QIBM

El directorio /QOpenSys/QIBM es el directorio del sistema para el sistema de archivos QOpenSys.

/QOpenSys/QIBM/ProdData

El directorio /QOpenSys/QIBM/ProdData es el directorio del sistema para el sistema de archivos QOpenSys y se utiliza para datos de producto de Programas bajo Licencia.

/QOpenSys/QIBM/UserData

El directorio /QOpenSys/QIBM/UserData es el directorio del sistema para el sistema de archivos QOpenSys y se utiliza para datos de usuario de Programa bajo Licencia, como archivos de configuración.

/asp_name/QIBM

El directorio /asp_name/QIBM es el directorio del sistema para cualquier ASP independiente que exista en el sistema, siendo asp_name el nombre de la ASP independiente.

/asp_name/QIBM/UserData

El directorio /asp_name/QIBM/UserData es un directorio del sistema que se utiliza para datos de usuario de Programa bajo Licencia como archivos de configuración para cualquier ASP independiente que exista en el sistema, siendo asp_name el nombre de la ASP independiente.

Capítulo 4. Acceso al sistema de archivos integrado con el iSeries Navigator

El iSeries Navigator es la interfaz gráfica de usuario que permite gestionar y administrar sus sistemas desde el escritorio de Windows. El iSeries Navigator facilita el manejo y la administración del sistema y mejora la productividad. Por ejemplo, se puede copiar un perfil de usuario a otro equipo arrastrándolo de un servidor iSeries a otro servidor iSeries. Los asistentes le guiarán en las tareas de configurar la seguridad y los servicios y aplicaciones de TCP/IP.

Existen muchas tareas que pueden llevarse a cabo utilizando el iSeries Navigator. A continuación se listan algunas de ellas, que le ayudarán a empezar:

Trabajar con archivos y carpetas

- “Crear una carpeta” en la página 40
- “Eliminar una carpeta” en la página 40
- “Reincorporar un archivo”
- “Reservar un archivo” en la página 38
- “Establecer permisos para un archivo o carpeta” en la página 38
- “Configurar la conversión de archivos de texto” en la página 38
- “Enviar un archivo o carpeta a otro sistema” en la página 38
- “Cambiar opciones de la definición del paquete” en la página 39
- “Planificar fecha y hora en que enviar el archivo o carpeta” en la página 39

Trabajar con compartimientos de archivos

- “Crear un compartimiento de archivos” en la página 40
- “Cambiar un compartimiento de archivos” en la página 40

Trabajar con sistemas de archivos definidos por el usuario

- “Crear un sistema de archivos definido por el usuario” en la página 41
- “Montar un sistema de archivos definido por el usuario” en la página 41
- “Desmontar un sistema de archivos definido por el usuario” en la página 42

Objetos de diario

- “Iniciar el registro por diario” en la página 42
- “Finalizar registro por diario” en la página 42

Reincorporar un archivo

Para reincorporar un archivo:

1. En el **Navigator de iSeries**, pulse el botón derecho sobre el archivo que quiere reincorporar.
2. Seleccione **Propiedades**.
3. Seleccione **Propiedades de archivo -> Utilizar página**.
4. Pulse **Reincorporar**.

Reservar un archivo

Para reservar un archivo:

1. En el **Navegador de iSeries**, pulse el botón derecho en el archivo que quiere reservar.
2. Seleccione **Propiedades**.
3. Seleccione **Propiedades de archivo** → **Utilizar página**.
4. Pulse **Reservar**.

Establecer permisos para un archivo o carpeta

Añadir permisos a un objeto permite controlar la capacidad de otras personas de manipular ese objeto. Con los permisos se puede permitir a ciertos usuarios que sólo puedan ver los objetos, mientras que a otros se les puede dejar editarlos.

Para establecer los permisos para un archivo o carpeta:

1. Expanda el sistema que desea utilizar en la ventana del **Navegador de iSeries**.
2. Despliegue **Sistemas de archivos**.
3. Despliegue **Sistema de archivos integrado**. Continúe desplegando opciones hasta que vea el objeto para el que se quiere añadir permisos.
4. Pulse con el botón derecho el objeto para el que se quiere añadir permisos y seleccione **Permisos**.
5. Pulse **Añadir** en el recuadro de diálogo **Permisos**.
6. Seleccione uno o más usuarios y grupos o escriba el nombre de un usuario o grupo en el campo nombre de usuario o de grupos del recuadro de diálogo **Añadir**.
7. Pulse **Aceptar**. Esto añadirá los usuarios o grupos al inicio de la lista.
8. Pulse el botón **Detalles** para implementar permisos más detallados.
9. Aplique los permisos para el usuario marcando el recuadro de selección apropiado.
10. Pulse **Aceptar**.

Configurar la conversión de archivos de texto

Se puede configurar la conversión automática de archivos de texto en el iSeries Navigator. La conversión automática de archivos de texto le permite utilizar extensiones de archivos para la conversión de datos de archivos. El sistema de archivos integrado puede convertir un archivo de datos cuando se transfiere entre un iSeries y un PC. Cuando se accede al archivo de datos desde un PC, se trata como si estuviera en ASCII.

Para configurar la conversión de archivos de texto:

1. Expanda el sistema que desea utilizar en el **iSeries Navigator**.
2. Despliegue **Sistemas de archivos**.
3. Pulse con el botón derecho del ratón en **Sistema de archivos integrado** y seleccione **Propiedades**.
4. Escriba la extensión del archivo que se quiere convertir automáticamente en el recuadro de texto **Extensiones de archivo para conversión automática de archivos de texto** y pulse **Añadir**.
5. Repita el paso 4 para todas las extensiones de archivo que quiera convertir automáticamente.
6. Pulse **Aceptar**.

Enviar un archivo o carpeta a otro sistema

Para enviar un archivo o carpeta a otro equipo:

1. Expanda el sistema que desea utilizar en el **iSeries Navigator**.
2. Despliegue **Sistemas de archivos**.

3. Despliegue **Sistema de archivos integrado**. Continúe desplegando opciones hasta que vea el archivo o carpeta que se quiere enviar.
4. Pulse con el botón derecho del ratón en el archivo o carpeta y seleccione **Enviar**. El archivo o carpeta aparecerá en la lista Archivos y carpetas seleccionados del recuadro de diálogo **Enviar archivos desde**.
5. Despliegue la lista de equipos y grupos disponibles.
6. Seleccione un equipo y pulse **Añadir** para añadirlo a la lista **Equipos y grupos de destino**. Repita este paso para todos los equipos a los que se quiere enviar este archivo o carpeta.
7. Pulse **Aceptar** para enviar el archivo o carpeta con las definiciones del paquete por omisión y la información de planificación actuales.

También puede “Cambiar opciones de la definición del paquete” o “Planificar fecha y hora en que enviar el archivo o carpeta”.

Cuando se crea una definición de paquete, ésta se guarda y puede volver a utilizarse en cualquier momento para enviar el conjunto definido de archivos y carpetas a múltiples sistemas de punto final o grupos de sistemas. Si decide crear una instantánea de sus archivos, puede conservar más de una versión de copias del mismo conjunto de archivos. Enviar una instantánea garantiza que no se realizan actualizaciones de los archivos durante la distribución, de forma que el último sistema de destino recibe los mismos objetos que el primer sistema de destino.

Cambiar opciones de la definición del paquete

Una definición de paquete le permite agrupar un conjunto de objetos del OS/400 o archivos del sistema de archivos integrado. La definición de paquete también le permite este mismo grupo de archivos como un conjunto lógico, o como un conjunto físico, tomando una instantánea de los archivos para conservarlos para su posterior distribución.

Para cambiar las opciones de las definiciones del paquete:

1. Lleve a cabo todos los pasos de “Enviar un archivo o carpeta a otro sistema” en la página 38.
2. Pulse la pestaña **Opciones**. Las opciones por omisión son incluir las subcarpetas a la hora de empaquetar y enviar archivos y sustituir un archivo existente por el archivo que se va a enviar.
3. Modifique las opciones según le interese.
4. Pulse **Avanzado** para establecer opciones de salvar y restaurar avanzadas.
5. Pulse **Aceptar** para guardar las opciones avanzadas.
6. Pulse **Aceptar** para enviar el archivo, o pulse **Planificar** para establecer una hora a la que enviarlo.

Temas relacionados:

- “Planificar fecha y hora en que enviar el archivo o carpeta”.

Planificar fecha y hora en que enviar el archivo o carpeta

El uso de la función de planificación le ofrece la flexibilidad de realizar su trabajo cuando le resulte más conveniente. Para planificar fecha y hora en que enviar el archivo o carpeta:

1. Lleve a cabo todos los pasos de “Enviar un archivo o carpeta a otro sistema” en la página 38.
2. Pulse **Planificar**.
3. Seleccione las opciones relativas al momento en que se quiere enviar el archivo o carpeta.

Crear una carpeta

Para crear una carpeta:

1. Expanda el sistema que desea utilizar en el **iSeries Navigator**.
2. Despliegue **Sistemas de archivos**.
3. Despliegue **Sistema de archivos integrado**.
4. Pulse con el botón derecho del ratón el sistema de archivos en el que se quiere añadir la carpeta nueva y seleccione **Carpeta nueva**.
5. Escriba un nombre nuevo para el objeto en el recuadro de diálogo **Carpeta nueva**.
6. Pulse **Aceptar**.

Cuando se crea una carpeta en el servidor iSeries, debe considerar si quiere o no proteger la carpeta (u objeto) nuevo con la administración de diario. Consulte el tema Administración de diario para obtener más información.

Temas relacionados:

- Iniciar registro por diario
- Finalizar registro por diario

Eliminar una carpeta

Para eliminar una carpeta:

1. Expanda el sistema que desea utilizar en el **Navigator del iSeries**.
2. Despliegue **Sistemas de archivos**.
3. Despliegue **Sistema de archivos integrado**. Continúe desplegando opciones hasta que vea el archivo o carpeta que se quiere eliminar.
4. Pulse con el botón derecho del ratón en el archivo o carpeta y seleccione **Suprimir**.

Crear un compartimiento de archivos

Un compartimiento de archivos es una vía de acceso de directorios que el NetServer del iSeries comparte con los clientes de PC en la red del iSeries. Un compartimiento de archivos puede consistir en cualquier directorio del sistema de archivos integrado en el iSeries.

Para crear un compartimiento de archivos:

1. Expanda el sistema en el **Navigator del iSeries**.
2. Despliegue **Sistemas de archivos**.
3. Despliegue **Sistema de archivos integrado**.
4. Despliegue el sistema de archivos que contiene la carpeta para la que se quiere crear un compartimiento.
5. Pulse con el botón derecho del ratón la carpeta para la que se quiere crear un compartimiento y seleccione **Compartimiento**.
6. Seleccione **Compartimiento nuevo**.

Cambiar un compartimiento de archivos

Un compartimiento de archivos es una vía de acceso de directorios que el NetServer del iSeries comparte con los clientes de PC en la red del iSeries. Un compartimiento de archivos puede consistir en cualquier directorio del sistema de archivos integrado en el iSeries.

Para cambiar un compartimiento de archivos:

1. Expanda el sistema en el **Navigator del iSeries**.
2. Despliegue **Sistemas de archivos**.
3. Despliegue **Sistema de archivos integrado**.
4. Despliegue la carpeta que tiene el compartimiento definido que se quiere cambiar.
5. Pulse con el botón derecho del ratón la carpeta que tiene el compartimiento definido y seleccione **Compartimiento**.
6. Seleccione **Compartimiento nuevo**.

Crear un sistema de archivos definido por el usuario

Un sistema de archivos definido por el usuario (UDFS) es un sistema de archivos para el que se crean y definen atributos. Los UDFS residen en agrupaciones de almacenamiento auxiliar (ASP) en el sistema.

Para crear un sistema de archivos definido por el usuario (UDFS) nuevo:

1. Expanda el sistema en el **Navigator del iSeries**.
2. Despliegue **Sistemas de archivos**.
3. Despliegue **Sistema de archivos integrado**.
4. Despliegue **Root (Raíz)**.
5. Despliegue **Dev**.
6. Pulse la agrupación de almacenamiento auxiliar (ASP) que se quiere que contenga el UDFS nuevo.
7. Seleccione **UDFS nuevo** del menú **Archivo**.
8. Especifique el nombre del UDFS, la descripción (opcional), los valores de auditoría, el formato de archivo por omisión y si los nombres de archivo del UDFS nuevo distinguirán entre mayúsculas y minúsculas en el recuadro de diálogo **Sistema de archivos nuevo definido por el usuario**.

Montar un sistema de archivos definido por el usuario

Un sistema de archivos definido por el usuario (UDFS) es un sistema de archivos para el que se crean y definen atributos. Los UDFS residen en agrupaciones de almacenamiento auxiliar (ASP) en el sistema. Para acceder o ver los almacenes de datos en un UDFS, debe montar el UDFS.

Al montar un UDFS, éste cubre cualquier sistema de archivos, directorios u objetos que existan bajo el punto de montaje en la jerarquía de carpetas. Dichos sistemas de archivos, directorios u objetos dejan de estar accesibles hasta que se desmonta el UDFS. Para garantizar el acceso a todos los datos del sistema de archivos integrado, monte el UDFS en una carpeta vacía. Tras montar el UDFS, podrá accederse a los archivos del UDFS desde esa carpeta. Cualquier cambio que se introduzca en la carpeta se efectuará también en el UDFS, en lugar de la carpeta cubierta.

Nota: No es posible montar un UDFS en una ASP independiente.

Para montar un sistema de archivos definido por el usuario (UDFS):

1. Expanda su sistema en el **Navigator de iSeries**.
2. Despliegue **Sistemas de archivos**.
3. Despliegue **Sistema de archivos integrado**.
4. Despliegue **Root (Raíz)**.
5. Despliegue **Dev**.
6. Pulse en la agrupación de almacenamiento auxiliar (ASP) que contiene el UDFS que se quiere montar.
7. En la columna **Nombre del UDFS** del panel derecho de Operations Navigator, pulse con el botón derecho del ratón el UDFS que se quiere montar.
8. Seleccione **Montar**.

Si prefiere arrastrar y soltar, se puede montar un UDFS arrastrándolo a una carpeta del sistema de archivos integrado del mismo servidor. No se puede soltar el UDFS en /dev, /dev/QASPxx, /dev/asp_name, otro equipo o el escritorio.

Desmontar un sistema de archivos definido por el usuario

Al montar un UDFS, éste cubre cualquier sistema de archivos, directorios u objetos que existan bajo el punto del montaje en la jerarquía de carpetas. Dichos sistemas de archivos, directorios u objetos dejan de estar accesibles hasta que se desmonta el UDFS.

Para desmontar un sistema de archivos definido por el usuario (UDFS):

1. Despliegue su equipo en **Operations Navigator**.
2. Despliegue **Sistemas de archivos**.
3. Despliegue **Sistema de archivos integrado**.
4. Despliegue **Root (Raíz)**.
5. Despliegue **Dev**.
6. Pulse en la agrupación de almacenamiento auxiliar (ASP) que contiene el UDFS que se quiere desmontar.
7. Pulse el botón derecho sobre el UDFS que desea desmontar en la columna **Nombre del UDFS** del panel derecho del Navigator del iSeries.
8. Seleccione **Desmontar**.

Iniciar el registro por diario

La función primaria del registro por diario es permitir la recuperación de los cambios realizados en un objeto desde que se guardó por última vez.

Para iniciar el registro por diario en un objeto:

1. Expanda su sistema en el Navigator de **iSeries**.
2. Despliegue **Sistemas de archivos**.
3. Pulse el botón derecho sobre el objeto que desea registrar por diario, y seleccione **Registro por diario....**
4. Tras seleccionar las opciones de registro por diario apropiado, pulse en **Iniciar**.

Para obtener más información sobre el registro por diario de objetos del sistema de archivos integrado, consulte gestión por diarios del iSeries Information Center.

Finalizar registro por diario

La función primaria del registro por diario es permitir la recuperación de los cambios realizados en un objeto desde que se guardó por última vez. Consulte Iniciar el registro por diario para obtener más información sobre cómo iniciar el registro por diario en un objeto. Tras iniciar el registro por diario en un objeto, sea cual sea el motivo, quizás desee finalizar el registro por diario en dicho objeto.

Para finalizar el registro por diario en un objeto:

1. Expanda el sistema en el **iSeries Navigator**.
2. Despliegue **Sistemas de archivos**.
3. Pulse el botón derecho sobre el objeto para el que desea detener el registro por diario, y seleccione **Registro por diario....**
4. Pulse en **Finalizar**.

| Para obtener más información sobre el registro por diario de objetos del sistema de archivos integrado,
| consulte gestión por diarios del iSeries Information Center.

Capítulo 5. Soporte de programación para el sistema de archivos integrado

La incorporación del sistema de archivos integrado al iSeries server no afecta a las aplicaciones del servidor iSeries ya existentes. Los lenguajes de programación, los programas de utilidad y el soporte del sistema como, por ejemplo, las especificaciones de descripción de datos, funcionan del mismo modo que antes de incluir el sistema de archivos integrado.

De todas maneras, para aprovechar las ventajas de los archivos continuos, directorios y demás soporte del sistema de archivos integrado, se deben utilizar un conjunto de las interfaces de programas de aplicación (API) de lenguaje C que se proporcionan para acceder a las funciones del sistema de archivos integrado.

Además, la incorporación del sistema de archivos integrado permite copiar datos entre archivos de bases de datos físicos y archivos continuos. Esta copia puede efectuarse mediante mandatos CL, la función de transferencia de datos de iSeries Access, o las API.

Los siguientes temas explican el modo de utilizar funciones de copia con archivos continuos del sistema de archivos integrado, e introducen las API como un modo de acceder a las funciones del sistema de archivos integrado:

- Copia de datos entre archivos continuos y archivos de base de datos
- Copia de datos entre archivos continuos y archivos de salvar
- Realizar operaciones utilizando API
- Soporte de zócalos
- Soporte internacional y asignación de nombres
- Conversión de datos

Copia de datos entre archivos continuos y archivos de base de datos

Si está familiarizado con el funcionamiento de los archivos de base de datos que utilizan recursos orientados a registros como, por ejemplo, las especificaciones de descripción de datos (DDS), puede que encuentre algunas diferencias fundamentales en la forma de trabajar con archivos continuos. Las diferencias son producto de la distinta estructura (o tal vez ausencia de estructura) de los archivos continuos en comparación con los archivos de base de datos. Para acceder a los datos de un archivo continuo, el usuario indica un desplazamiento de byte y una longitud. Para acceder a los datos de un archivo de base de datos, el usuario normalmente define los campos que se han de utilizar, y la cantidad de registros que se han de procesar.

Ya que el usuario define el formato y las características de un archivo orientado a registros por anticipado, el sistema operativo sabe cómo es el archivo, lo que puede ayudar a evitar que se efectúen operaciones que no son adecuadas para el formato y las características del archivo. Con los archivos continuos, el sistema operativo apenas sabe cómo es el formato del archivo. Es la aplicación la que debe saber cuál es el formato del archivo y cómo trabajar con él correctamente. Los archivos continuos proporcionan un entorno de programación sumamente flexible, pero a costa de recibir poca o ninguna ayuda del sistema operativo. Los archivos continuos son más apropiados en ciertas situaciones de programación; los archivos orientados a registros son más adecuados en otras.

Hay varias formas de copiar datos entre archivos continuos y archivos de base de datos en el sistema de archivos integrado:

- Copiar datos mediante mandatos CL
- Copiar datos mediante API
- Copiar datos utilizando la función de transferencia de datos

Copiar datos mediante mandatos CL

Existen dos conjuntos de mandatos de CL que permiten copiar datos entre archivos continuos y miembros de archivos de base de datos:

- CPYTOSTMF y CPYFRMSTMF
- CPYTOIMPF y CPYFRMIMPF

Mandatos CPYTOSTMF y CPYFRMSTMF

Se pueden utilizar los mandatos Copiar de archivo continuo (CPYFRMSTMF) y Copiar en archivo continuo (CPYTOSTMF) para copiar datos entre archivos continuos y miembros de archivos de base de datos. Se puede crear un archivo continuo a partir de un miembro de archivo de base de datos utilizando el mandato CPYTOSTMF. También se puede crear un miembro de archivo de base de datos utilizando el mandato CPYFRMSTMF. Si el archivo o miembro que es el destino de la copia no existe, se crea.

No obstante, existen algunas limitaciones. El archivo de base de datos debe ser un archivo físico descrito por programa que contenga un solo campo, o bien un archivo físico fuente que contenga un solo campo de texto. Los mandatos proporcionan una gran variedad de opciones para convertir y volver a dar formato a los datos que se están copiando.

Los mandatos CPYTOSTMF y CPYFRMSTMF también pueden utilizarse para copiar datos entre un archivo continuo y un archivo de salvar.

Mandatos CPYTOIMPF y CPYFRMIMPF

También pueden utilizarse los mandatos Copiar a archivo de importación (CPYTOIMPF) y Copiar de archivo de importación (CPYFRMIMPF) para copiar datos entre archivos continuos y miembros de archivos de base de datos. Los mandatos CPYTOSTMF y CPYFRMSTMF no permiten mover datos procedentes de archivos de base de datos complejos descritos externamente (descritos mediante DDS). La expresión *archivo de importación* hace referencia a un archivo de tipo continuo; habitualmente, este término se refiere a un archivo creado a efectos de copia de datos entre bases de datos heterogéneas.

En la copia de datos procedentes de un archivo continuo (o de importación), el mandato CPYFRMIMPF permite especificar un archivo de definición de campos (FDF), que describe los datos del archivo continuo. También puede especificarse que los archivos continuos son delimitados, y los caracteres que se utilizan para marcar los límites de serie, campo y registro. También se proporcionan opciones para la conversión de tipos de datos especiales, tales como campos de hora y fecha.

En estos mandatos se proporciona la conversión de datos si el archivo continuo o miembro de base de datos de destino ya existen. Si el archivo no existe, puede utilizar el siguiente método que consta de dos pasos para efectuar la conversión de los datos:

1. Utilice los mandatos CPYTOIMPF y CPYFRMIMPF para copiar los datos entre el archivo descrito externamente y un archivo físico fuente.
2. Utilice los mandatos CPYTOSTMF y CPYFRMSTMF (que proporcionan una conversión completa de los datos exista o no el archivo de destino) para copiar los datos entre el archivo físico fuente y el archivo continuo.

A continuación se ofrece un ejemplo:

```
CPYTOIMPF FROMFILE(DB2FILE) TOFILE(EXPFILE) DTAFMT(*DLM)
          FLDDLML(';') RCDDLML('X'07') STRDLML('') DATFMT(*USA) TIMFMT(*USA)
```

El parámetro DTAFMT especifica que el archivo continuo (de importación) de entrada es delimitado; la otra opción es DTAFMT(*FIXED), que obliga a especificar un archivo de definición de campos. Los parámetros FLDDLML, RCDDLML y STRDLML identifican los caracteres que actúan como delimitadores o separadores de campos, registros y series, respectivamente.

Los parámetros DATFMT y TIMFMT indican el formato que tendrá toda la información de fecha y hora que se copie al archivo de importación.

El uso de mandatos resulta de utilidad porque pueden ponerse en un programa y se ejecutan totalmente en el servidor. No obstante, las interfaces son complejas.

Para obtener más información, consulte la ayuda del mandato o el apartado Lenguaje de mandatos (CL) del iSeries Information Center.

Copiar datos mediante API

Si desea copiar miembros de un archivo de base de datos a un archivo continuo en una aplicación, puede utilizar las funciones open(), read() y write() del sistema de archivos integrado para abrir un miembro, leer datos de su interior y escribir datos en el mismo. Para obtener más información, consulte el apartado API del sistema de archivos integrado del iSeries Information Center.

Copia de datos utilizando la función de transferencia de datos

Las aplicaciones de transferencia de datos del iSeries Access presentan la ventaja de tener una interfaz gráfica fácil de utilizar y una función de conversión automática de los datos de tipo numérico y carácter. Sin embargo, la transferencia de datos requiere la instalación del producto iSeries Access y el uso de recursos del PC y del iSeries server, así como las comunicaciones entre ambos.

Si tiene instalado iSeries Access en el PC y en el servidor, también puede utilizar las aplicaciones de transferencia de datos para transferir datos entre archivos continuos y archivos de base de datos. También pueden transferirse datos a un archivo de base de datos nuevo basado en un archivo de base de datos ya existente, a un archivo de base de datos descrito externamente, o a una definición de archivo de base de datos y archivo nuevos.

Las tareas siguientes le ayudarán a copiar y transferir datos utilizando las aplicaciones de transferencia de datos:

- Transferir datos de un archivo de base de datos a un archivo continuo
- Transferir datos de un archivo continuo a un archivo de base de datos
- Transferir datos a una definición de archivo de base de datos y archivo recién creados
- Crear un archivo de descripción de formato

Transferir datos de un archivo de base de datos a un archivo continuo

Para transferir un archivo de un archivo de base de datos a un archivo continuo del servidor:

1. Establezca una conexión con el servidor.
2. Correlacione una unidad de red con la vía de acceso apropiada del sistema de archivos del iSeries.
3. Seleccione **Transferencia de datos desde iSeries server** de la ventana iSeries Access para Windows.
4. Seleccione el servidor desde el que quiere realizar la transferencia.
5. Seleccione los nombres de archivo, utilizando el nombre de archivo y la biblioteca de la base de datos del iSeries a partir de los que se realizará la copia, y la unidad de red para la ubicación del archivo continuo resultante. También puede elegir **Detalles del archivo de PC** para seleccionar el formato del archivo continuo del PC. La transferencia de datos admite los tipos de archivo de PC más comunes, como son los de texto ASCII, BIFF3, CSV, DIF, de texto delimitado con tabuladores o WK4.
6. Pulse **Transferir datos desde iSeries** para ejecutar la transferencia de los archivos.

Estos datos también pueden trasladarse ejecutando un trabajo de proceso por lotes con las aplicaciones de transferencia de datos. Proceda del mismo modo que antes, pero seleccione la opción de menú **Archivo** para guardar la petición de transferencia. La aplicación Transferencia de datos a iSeries server

crea un archivo .DTT o .TFR. La aplicación Transferencia de datos desde iSeries server crea un archivo .DTF o .TTO. En el directorio iSeries Access, pueden ejecutarse por lotes dos programas desde una línea de mandatos:

- RTOPCB toma un archivo .DTF o .TTO como parámetro
- RFROMPCB toma un archivo .DTT o .TFR como parámetro

Mediante una aplicación planificadora puede establecer que se ejecute cualquiera de estos mandatos de forma planificada. Por ejemplo, puede utilizar System Agent Tool (que forma parte del paquete Microsoft Plus) para especificar el programa que debe ejecutarse (por ejemplo, RTOPCB MIARCH.TTO) y la hora en la que desea que se ejecute el programa.

Transferir datos de un archivo continuo a un archivo de base de datos

Para transferir datos de un archivo continuo a un archivo de base de datos del servidor:

1. Establezca una conexión con el servidor.
2. Correlacione una unidad de red con la vía de acceso apropiada del sistema de archivos del iSeries.
3. Seleccione **Transferencia de datos a iSeries server** desde la ventana iSeries Access para Windows.
4. Seleccione el nombre de archivo de PC que se quiere transferir. Para el nombre de archivo de PC, puede seleccionar **Examinar** la unidad de red que haya asignado y elegir un archivo continuo. También puede utilizar un archivo continuo ubicado en el mismo PC.
5. Seleccione el servidor en el que se quiere ubicar el archivo de base de datos descrito externamente.
6. Pulse **Transferir datos a iSeries server** para ejecutar la transferencia de archivos.

Nota: si traslada datos a una definición de archivo de base de datos que ya existe en el servidor, la aplicación Transferencia de datos a iSeries server le obligará a utilizar un archivo de descripción de formato (FDF) asociado. Un archivo FDF describe el formato de un archivo continuo, y la aplicación Transferencia de datos desde iSeries server lo crea cuando transfiere datos de un archivo de base de datos a un archivo continuo. Para completar la transferencia de datos de un archivo continuo a un archivo de base de datos, pulse **Transferir datos a iSeries**. Si no existe un archivo .FDF disponible, puede crear un archivo .FDF rápidamente.

Estos datos también pueden trasladarse ejecutando un trabajo de proceso por lotes con las aplicaciones de transferencia de datos. Proceda del mismo modo que antes, pero seleccione la opción de menú **Archivo** para guardar la petición de transferencia. La aplicación Transferencia de datos a iSeries server crea un archivo .DTT o .TFR. La aplicación Transferencia de datos desde iSeries server crea un archivo .DTF o .TTO. En el directorio iSeries Access, pueden ejecutarse por lotes dos programas desde una línea de mandatos:

- RTOPCB toma un archivo .DTF o .TTO como parámetro
- RFROMPCB toma un archivo .DTT o .TFR como parámetro

Mediante una aplicación planificadora puede establecer que se ejecute cualquiera de estos mandatos de forma planificada. Por ejemplo, puede utilizar System Agent Tool (que forma parte del paquete Microsoft Plus) para especificar el programa que debe ejecutarse (por ejemplo, RTOPCB MIARCH.TTO) y la hora en la que desea que se ejecute el programa.

Transferir datos a una definición de archivo de base de datos y archivo recién creados

Para transferir datos a una definición de archivo de base de datos y archivo recién creados:

1. Establezca una conexión con el servidor.
2. Correlacione una unidad de red con la vía de acceso apropiada del sistema de archivos del iSeries.
3. Seleccione **Transferencia de datos a iSeries server** desde la ventana iSeries Access para Windows.
4. Abra el menú **Herramientas** de la aplicación Transferencia de datos a iSeries server.
5. Seleccione **Crear archivo de base de datos de iSeries**. .
Aparecerá un asistente que le permitirá crear un archivo de base de datos de iSeries nuevo a partir de un archivo de PC existente. Se le solicitará que especifique el nombre del archivo de PC en el que se

basará el archivo de iSeries, el nombre del archivo de iSeries a crear, y otros detalles necesarios. Esta herramienta analiza un archivo continuo dado para determinar el número, tipo y tamaño de los campos necesarios en el archivo de base de datos resultante. A continuación, la herramienta puede crear la definición del archivo de base de datos en el servidor.

Crear un archivo de descripción de formato

Si traslada datos a una definición de base de datos que ya existe en el servidor, la aplicación Transferencia de datos a iSeries server le obligará a utilizar un archivo de descripción de formato (FDF) asociado. Un archivo FDF describe el formato de un archivo continuo, y la aplicación Transferencia de datos desde iSeries server lo crea cuando transfiere datos de un archivo de base de datos a un archivo continuo.

Para crear un archivo .FDF:

1. Cree un archivo de base de datos descrito externamente con un formato que coincida con el archivo continuo fuente (número de campos, tipos de datos).
2. Cree un registro de datos temporal en el archivo de base de datos.
3. Utilice la función Transferencia de datos desde iSeries server para crear un archivo continuo y su archivo .FDF asociado a partir de este archivo de base de datos.
4. A continuación, puede utilizar la función Transferencia de datos a iSeries server. Especifique este archivo .FDF con el archivo continuo fuente que desea transferir.

Copia de datos entre archivos continuos y archivos de salvar

Un archivo de salvar se utiliza con mandatos de salvar y de restaurar para retener datos que de otro modo se grabarían en cinta o disquete. El archivo también puede utilizarse como un archivo de base de datos para leer o grabar registros que contengan información de salvar y restaurar. Los archivos de salvar también puede utilizarse para enviar objetos a otros usuario de la red SNADS.

El mandato CPY puede servir para copiar un archivo de salvar hacia y desde un archivo continuo. Sin embargo, al volver a copiar un archivo continuo en un objeto archivo de salvar, los datos deben ser datos válidos para un archivo de salvar (deben haberse originado en un archivo de salvar y haberse copiado en un archivo continuo).

Al utilizar un PC cliente, también se puede acceder al archivo de salvar y copiar los datos al almacenamiento de PC o a la red local. Aun así, recuerde que no es posible acceder a los datos de los archivos de salvar a través del sistema de archivos de red (NFS).

Realizar operaciones utilizando API

Las interfaces de programa de aplicación que realizan operaciones en los directorios y archivos continuos del sistema de archivos integrado tienen el formato de funciones de lenguaje C. Puede elegir entre dos conjuntos de funciones, cada una de las cuales se puede utilizar en programas creados utilizando el entorno de lenguajes integrados (ILE) C/400:

- Funciones C del sistema de archivos integrado que se incluyen en OS/400.
- Funciones C proporcionadas por el programa bajo licencia ILE C/400.

Las funciones del sistema de archivos integrado funcionan sólo mediante el soporte de E/S continua del sistema de archivos integrado. Se admiten las siguientes API:

Tabla 3. API del sistema de archivos integrado

Función	Descripción
access()	Determinar accesibilidad de archivo
accessx()	Determinar accesibilidad de archivo para una clase de usuarios

Tabla 3. API del sistema de archivos integrado (continuación)

Función	Descripción
chdir()	Cambiar directorio actual
chmod()	Cambiar autorizaciones de archivo
chown()	Cambiar propietario y grupo de archivo
close()	Cerrar descriptor de archivo
closedir()	Cerrar directorio
creat()	Crear archivo nuevo o volver a grabar archivo existente
creat64()	Crear archivo nuevo o volver a grabar archivo existente (con soporte de archivos grandes)
DosSetFileLocks()	Bloquear y desbloquear el rango de bytes de un archivo.
DosSetFileLocks64()	Bloquear y desbloquear el rango de bytes de un archivo (con soporte de archivos grandes).
DosSetRelMaxFH()	Cambiar el número máximo de descriptores de archivos
dup()	Duplicar descriptor de archivo abierto
dup2()	Duplicar descriptor de archivo abierto en otro descriptor
faccessx()	Determinar accesibilidad de archivo para una clase de usuarios por descriptor
fchdir()	Cambiar directorio actual por descriptor
fchmod()	Cambiar autorizaciones de archivo por descriptor
fchown()	Cambiar propietario y grupo de archivo por descriptor
fcntl()	Efectuar acción de control de archivo
fpathconf()	Obtener variables de nombre de vía de acceso configurables por descriptor
fstat()	Obtener información de archivo por descriptor
fstat64()	Obtener información de archivo por descriptor (con soporte de archivos grandes)
fstatvfs()	Obtener información por descriptor
fstatvfs64()	Obtener información por descriptor (con soporte de 64 bits)
fsync()	Sincronizar cambios en archivo
ftruncate()	Truncar archivo
ftruncate64()	Truncar archivo (con soporte de archivos grandes)
getcwd()	Obtener nombre de vía de acceso de directorio actual
getegid()	Obtener ID de grupo efectivo
geteuid()	Obtener ID de usuario efectivo
getgid()	Obtener ID de grupo real
getgrgid()	Obtener información de grupo utilizando ID de grupo
getgrnam()	Obtener información de grupo utilizando nombre de grupo
getgroups()	Obtener ID de grupo
getwponam()	Obtener información de usuario del nombre de usuario
getpwuid()	Obtener información de usuario del ID de usuario
getuid()	Obtener ID de usuario real
givedescriptor()	Proporcionar acceso de archivo a otro trabajo

Tabla 3. API del sistema de archivos integrado (continuación)

Función	Descripción
ioctl()	Efectuar acción de control de E/S de archivo
link()	Crear enlace con archivo
lseek()	Establecer desplazamiento de lectura/grabación de archivo
lseek64()	Establecer desplazamiento de lectura/grabación de archivo (con soporte de archivos grandes)
lstat()	Obtener información de archivo o enlace
lstat64()	Obtener información de archivo o enlace (con soporte de archivos grandes)
mmap()	Crear correlación de memoria
mmap64()	Crear correlación de memoria (con soporte de archivos grandes)
mprotect()	Cambiar protección de una correlación de memoria
msync()	Sincronizar correlación de memoria
munmap()	Eliminar correlación de memoria
mkdir()	Crear directorio
mkfifo()	Crear archivo especial FIFO
open()	Abrir archivo
open64()	Abrir archivo (con soporte de archivos grandes)
opendir()	Abrir directorio
pathconf()	Obtener variables de nombre de vía de acceso configurables
pipe()	Crear canal entre procesos con zócalos
pread()	Leer de descriptor con desplazamiento
pread64()	Leer de descriptor con desplazamiento (con soporte de archivos voluminosos)
pwrite()	Grabar en descriptor con desplazamiento
pwrite64()	Grabar en descriptor con desplazamiento (con soporte de archivos voluminosos)
QjoEndJournal()	Finalizar registro por diario
QjoRetrieveJournal Information()	Recuperar información de diario
QJORJIDI()	Recuperar información del identificador de diario
QJOSJRNE()	Enviar entrada de diario
QjoStartJournal()	Iniciar registro por diario
QlgAccess()	Determinar accesibilidad de archivo (utilizando el nombre de vía de acceso habilitado para NLS)
QlgAccessx()	Determinar accesibilidad de archivo para una clase de usuarios (utilizando nombre de vía de acceso habilitado para NLS)
QlgChdir()	Cambiar directorio actual (utilizando el nombre de vía de acceso habilitado para NLS)
QlgChmod()	Cambiar autorizaciones sobre el archivo (utilizando el nombre de vía de acceso habilitado para NLS)
QlgChown()	Cambiar propietario y grupo de archivo (utilizando el nombre de vía de acceso habilitado para NLS)

Tabla 3. API del sistema de archivos integrado (continuación)

Función	Descripción
QlgCreat()	Crear archivo nuevo o volver a grabar un archivo existente (utilizando el nombre de vía de acceso habilitado para NLS)
QlgCreat64()	Crear archivo nuevo o volver a grabar un archivo existente (con soporte de archivos grandes y utilizando el nombre de vía de acceso habilitado para NLS)
QlgCvtPathToQSYSObjName()	Resolver nombre de vía de acceso del sistema de archivos integrado en el nombre de objeto QSYS (utilizando el nombre de vía de acceso habilitado para NLS)
QlgGetAttr()	Obtener atributos del sistema para un objeto (utilizando el nombre de la vía de acceso habilitado para NLS)
QlgGetcwd()	Obtener nombre de la vía de acceso del directorio actual (utilizando el nombre de la vía de acceso habilitado para NLS)
QlgGetPathFromFileID()	Obtener nombre de la vía de acceso de un objeto a partir de su ID de archivo (utilizando el nombre de vía de acceso habilitado para NLS)
QlgGetpwnam()	Obtener información de usuario a partir del nombre de usuario (utilizando el nombre de vía de acceso habilitado para NLS)
QlgGetpwnam_r()	Obtener información de usuario a partir del nombre de usuario (utilizando el nombre de vía de acceso habilitado para NLS)
QlgGetpwuid()	Obtener información de usuario a partir del ID de usuario (utilizando el nombre de vía de acceso habilitado para NLS)
QlgGetpwuid_r()	Obtener información de usuario a partir del ID de usuario (utilizando el nombre de vía de acceso habilitado para NLS)
QlgLchown()	Cambiar propietario y grupo de enlace simbólico (utilizando el nombre de vía de acceso habilitado para NLS)
QlgLink()	Crear enlace con archivo (utilizando el nombre de vía de acceso habilitado para NLS)
QlgLstat()	Obtener información de archivo o enlace (utilizando el nombre de vía de acceso habilitado para NLS)
QlgLstat64()	Obtener información de archivo o enlace (con soporte de archivos grandes y utilizando el nombre de vía de acceso habilitado para NLS)
QlgMkdir()	Crear directorio (utilizando el nombre de vía de acceso habilitado para NLS)
QlgMkfifo()	Crear archivo especial FIFO (utilizando el nombre de vía de acceso habilitado para NLS)
QlgOpen()	Abrir archivo (utilizando el nombre de vía de acceso habilitado para NLS)
QlgOpen64()	Abrir archivo (con soporte de archivos grandes y utilizando el nombre de vía de acceso habilitado para NLS)
QlgOpendir()	Abrir directorio (utilizando el nombre de vía de acceso habilitado para NLS)
QlgPathconf()	Obtener variables de nombre de vía de acceso configurables (utilizando el nombre de vía de acceso habilitado para NLS)
QlgProcessSubtree()	Procesar directorios u objetos de un árbol de directorios (utilizando el nombre de vía de acceso habilitado para NLS)
QlgReaddir()	Leer entrada de directorio (utilizando el nombre de vía de acceso habilitado para NLS)

Tabla 3. API del sistema de archivos integrado (continuación)

Función	Descripción
QlgReaddir_r()	Leer entrada de directorio (permitir ejecución multihebra y utilizar nombre de vía de acceso habilitado para NLS)
QlgReadlink()	Leer valor de enlace simbólico (utilizando el nombre de vía de acceso habilitado para NLS)
QlgRenameKeep()	Redenominar archivo o directorio, mantener <i>nuevo</i> si existe (utilizando el nombre de vía de acceso habilitado para NLS)
QlgRenameUnlink()	Redenominar archivo o directorio, desenlazar <i>nuevo</i> si existe (utilizando el nombre de vía de acceso habilitado para NLS)
QlgRmdir()	Eliminar directorio (utilizando el nombre de vía de acceso habilitado para NLS)
QlgSaveStgFree()	Guardar datos de objetos y liberar su almacenamiento (utilizando el nombre de vía de acceso habilitado para NLS)
QlgSetAttr()	Establecer atributos del sistema para un objeto (utilizando el nombre de vía de acceso habilitado para NLS)
QlgStat()	Obtener información de archivo (utilizando el nombre de vía de acceso habilitado para NLS)
QlgStat64()	Obtener información de archivo (con soporte de archivos grandes y utilizando el nombre de vía de acceso habilitado para NLS)
QlgStatvfs()	Obtener información del sistema de archivos (utilizando el nombre de vía de acceso habilitado para NLS)
QlgStatvfs64()	Obtener información del sistema de archivos (con soporte de archivos grandes y utilizando el nombre de vía de acceso habilitado para NLS)
QlgSymlink()	Crear enlace simbólico (utilizando el nombre de vía de acceso habilitado para NLS)
QlgUnlink()	Desenlazar archivo (utilizando el nombre de vía de acceso habilitado para NLS)
QlgUtime()	Establecer horas de acceso y de modificación de archivo (utilizando el nombre de vía de acceso habilitado para NLS)
QP0FPTOS()	Realizar funciones de diversa índole sobre sistemas de archivos
Qp0ICvtPathToSYSObjName()	Resolver nombre de vía de acceso del sistema de archivos integrado en el nombre de objeto QSYS
Qp0IFLOP()	Realizar operaciones de diversa índole sobre objetos
Qp0IGetAttr()	Obtener los atributos de sistema de un objeto
Qp0IGetPathFromFileID()	Obtener nombre de vía de acceso de objeto a partir de su ID de archivo
Qp0IOpen()	Abrir archivo con nombre de vía de acceso habilitado para NLS
Qp0IProcessSubtree()	Procesar directorios u objetos de un árbol de directorios
Qp0IRenameKeep()	Redenominar archivo o directorio, mantener <i>nuevo</i> si existe
Qp0IRenameUnlink()	Redenominar archivo o directorio, desenlazar <i>nuevo</i> si existe
QP0LROR()	Recuperar referencias de objetos
Qp0ISaveStgFree()	Guardar datos de objetos y liberar su almacenamiento
Qp0ISetAttr()	Establecer atributos de sistema de un objeto

Tabla 3. API del sistema de archivos integrado (continuación)

Función	Descripción
Qp0IUnlink()	Eliminar enlace con archivo con nombre de vía de acceso habilitado para NLS
qsyssetegid()	Establecer ID de grupo efectivo
qsysseteuid()	Establecer ID de usuario efectivo
qsyssetgid()	Establecer ID de grupo
qsyssetregid()	Establecer ID de grupo real y efectivo
qsyssetreuid()	Establecer ID de usuario real y efectivo
qsyssetuid()	Establecer ID de usuario
QZNFRTVE()	Recuperar información de exportación de NFS
read()	Leer de archivo
readdir()	Leer entrada de directorio
readdir_r()	Leer entrada de directorio (permitir ejecución multihebra)
readlink()	Leer valor de enlace simbólico
readv()	Leer de archivo (vector)
rename()	Redenominar archivo o directorio. Se puede definir para tener la semántica de Qp0IRenameKeep() o de Qp0IRenameUnlink().
rewinddir()	Restablecer corriente de directorio
rmdir()	Eliminar directorio
select()	Comprobar estado E/S de múltiples descriptores de archivos
stat()	Obtener información de archivo
stat64()	Obtener información de archivo (con soporte de archivos grandes)
statvfs()	Obtener información de sistema de archivos
statvfs64()	Obtener información de sistema de archivos (con soporte de archivos grandes)
symlink()	Crear enlace simbólico
sysconf()	Obtener variables de configuración del sistema
takedescriptor()	Tomar acceso de archivo desde otro trabajo
umask()	Establecer máscara de autorización para trabajo
unlink()	Eliminar enlace con archivo
utime()	Establecer horas de acceso y de modificación de archivo
write()	Grabar en archivo
writev()	Grabar en archivo (vector)

Nota: Algunas de estas funciones también las utilizan los zócalos de OS/400 . Para conocer las restricciones sobre el uso de estas funciones en sistemas de archivos concretos, consulte el apartado “Sistemas de archivos en el sistema de archivos integrado” en la página 4. En el Apéndice B, “Programa de ejemplo con funciones C del sistema de archivos integrado” en la página 105, puede consultar un programa de ejemplo que utiliza funciones de C del sistema de archivos integrado.

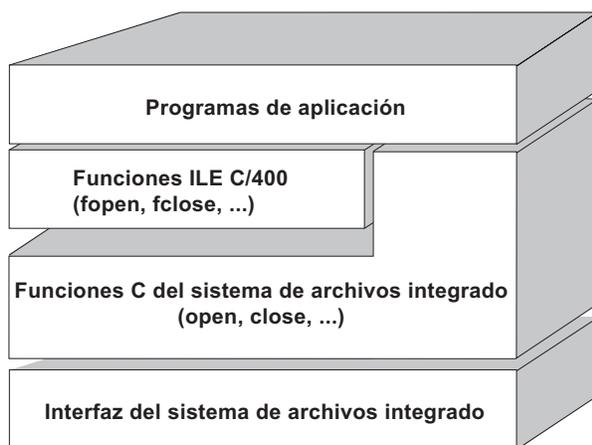
Si desea obtener más información sobre las API del sistema de archivos integrado, consulte los temas siguientes:

- funciones ILE C/400
- Soporte de archivos voluminosos para las API
- Reglas de nombres de vía de acceso para las API
- Descriptor de archivos
- Seguridad
- El apartado referente a las Interfaces de programación de aplicaciones (API) del iSeries Information Center

Funciones ILE C/400

ILE C/400 proporciona las funciones C estándar definidas por el ANSI (American National Standards Institute). Estas funciones pueden operar mediante el soporte de E/S de gestión de datos o el soporte de E/S continua del sistema de archivos integrado, según lo especificado al crear el programa C. El compilador utiliza la E/S de gestión de datos, a menos que se indique lo contrario.

Para indicar al compilador que utilice la E/S continua del sistema de archivos integrado, debe especificar *IFSIO en el parámetro Opción de interfaz de sistema (SYSIFCOPT) de los mandatos Crear módulo ILE C/400 (CRTCMOD) o Crear programa C enlazado (CRTBNDC). Al especificar *IFSIO, se enlazan las funciones de E/S del sistema de archivos integrado en lugar de las funciones de E/S de gestión de datos. De hecho, las funciones C de ILE C/400 utilizan las funciones del sistema de archivos integrado para efectuar la E/S.



RV3N070-3

Figura 10. Las funciones ILE C/400 utilizan las funciones de E/S continua del sistema de archivos integrado

Para obtener más información sobre la utilización de las funciones ILE C/400 con la E/S continua del sistema de archivos integrado consulte la publicación WebSphere Development Studio: ILE C/C++

Programmers Guide . Para obtener más información sobre cada función ILE C/400 C, consulte la publicación WebSphere Development Studio: C/C++ Language Reference .

Soporte de archivos voluminosos para las API

Las API del sistema de archivos integrado se han mejorado para permitir que las aplicaciones almacenen y manipulen archivos muy voluminosos. El sistema de archivos integrado admite tamaños de archivo continuo de hasta 256 gigabytes en el sistema de archivos "raíz" (/), el sistema de archivos de sistemas abiertos (QOpenSys), y el sistema de archivos definido por el usuario.

El sistema de archivos integrado proporciona un conjunto de API de tipo UNIX de 64 bits y permite correlacionar fácilmente las API de 32 bits existentes con las API de 64 bits capaces de acceder a tamaños y desplazamientos utilizando argumentos de tipo entero de ocho bytes. Para obtener más información sobre cada una de las API de 64 bits, consulte el apartado sobre las sistema de archivos integrado API de iSeries Information Center.

Se proporcionan las siguientes funciones para que las aplicaciones puedan utilizar el soporte de archivos voluminosos:

1. Si se define la etiqueta de macro `_LARGE_FILE_API` durante la compilación, las aplicaciones tendrán acceso a las API y a las estructuras de datos habilitadas para 64 bits. Por ejemplo, una aplicación en la que se pretenda utilizar la API `stat64()` y la estructura `stat64` deberá definir `_LARGE_FILE_API` durante la compilación.
2. Si las aplicaciones definen la etiqueta de macro `_LARGE_FILES` durante la compilación, las API y las estructuras de datos existentes se correlacionan con sus versiones de 64 bits. Por ejemplo, si una aplicación define `_LARGE_FILES` durante la compilación, una llamada a la API `stat()` se correlacionará con la API `stat64()` y una estructura `stat()` se correlacionará con una estructura `stat64()`.

Las aplicaciones en las que se pretenda utilizar el soporte de archivos grandes pueden definir `_LARGE_FILE_API` durante la compilación y codificar directamente las API de 64 bits, o bien pueden definir `_LARGE_FILES` durante la compilación. Todas las API y estructuras de datos apropiadas se correlacionarán automáticamente con sus versiones de 64 bits.

Las aplicaciones en las que no se pretenda utilizar el soporte de archivos voluminosos no se ven afectadas y pueden seguir utilizando las API del sistema de archivos integrado sin modificaciones.

Reglas de nombres de vía de acceso para las API

Cuando se utiliza una API del sistema de archivos integrado o de ILE C/400 para realizar operaciones sobre un objeto, éste se identifica especificando su vía de acceso de directorio. A continuación se proporciona un resumen de las reglas que hay que tener en cuenta al especificar nombres de vías de acceso en las API. El término **objeto** en estas reglas hace referencia a cualquier directorio, archivo, enlace u otro objeto.

- Los nombres de vías de acceso se especifican por orden jerárquico, empezando por el nivel superior de la jerarquía de directorios. El nombre de cada componente de la vía de acceso se separa con una barra inclinada (/); por ejemplo:

```
Dir1/Dir2/Dir3/ArchUsr
```

La barra inclinada invertida (\) no se reconoce como separador. Se maneja exactamente igual que otro carácter del nombre.

- Los nombres de objeto deben ser exclusivos en el directorio.
- La longitud máxima de cada componente del nombre de vía de acceso y la longitud máxima de la serie del nombre de vía de acceso pueden variar en cada sistema de archivos. Consulte el apartado “Comparación de los sistemas de archivos” en la página 61 para conocer los límites de cada sistema de archivos.
- Un carácter / al principio de un nombre de vía de acceso significa que la vía de acceso empieza en el directorio de mayor nivel, el directorio “raíz” (/); por ejemplo:

```
/Dir1/Dir2/Dir3/ArchUsr
```

- Si el nombre de vía de acceso no empieza con el carácter /, se presupone que la vía de acceso empieza en el directorio actual; por ejemplo:

```
MiDir/MiArch
```

donde MiDir es un subdirectorio del directorio actual.

- Para evitar confusiones con los valores especiales del iSeries server, los nombres de vía de acceso no pueden comenzar con un único caracter de asterisco (*). Para especificar un nombre de vía de acceso que comience con cualquier número de caracteres, utilice dos asteriscos (**); por ejemplo:

```
'**.file'
```

Tenga en cuenta que este caso sólo se aplica a los nombres de vía de acceso relativa donde no hay otros caracteres antes del asterisco (*).

- Cuando se trabaja con objetos del QSYS.LIB, los nombres de componentes deben tener el formato *name.object-type*; por ejemplo:

```
/QSYS.LIB/PAYROLL.LIB/PAY.FILE
```

Consulte el apartado “Sistema de archivos de biblioteca (QSYS.LIB)” en la página 72 para conocer más detalles.

- Cuando se trabaja con objetos del QSYS.LIB de ASP independiente, los nombres de componentes deben tener el formato *name.object-type*; por ejemplo:

```
'/asp_name/QSYS.LIB/PAYDAVE.LIB/PAY.FILE
```

Consulte el apartado “QSYS.LIB de ASP independiente” en la página 75 para conocer más detalles.

- No utilice signos de dos puntos (:) en los nombres de vía de acceso. Tienen un significado especial en el servidor.
- A diferencia de los nombres de vía de acceso de los mandatos del sistema de archivos integrado (consulte el apartado “Reglas sobre el nombre de ruta para mandatos CL y pantallas” en la página 29), el asterisco (*), el interrogante (?), el apóstrofo ('), las comillas (") y la tilde (~) carecen de significación especial. Se manejan exactamente igual que otro carácter del nombre. Las únicas API que constituyen una excepción a esta regla son: QjoEndJournal() y QjoStartJournal.

Descriptor de archivos

Al utilizar las funciones de E/S continua de ILE C/400 definidas por el ANSI (American National Standards Institute) para ejecutar operaciones en un archivo, el archivo se identifica mediante la utilización de punteros. Al utilizar las funciones C del sistema de archivos integrado, el archivo se identifica especificando un **descriptor de archivo**. Un descriptor de archivo es un entero positivo que debe ser exclusivo en cada trabajo. El trabajo utiliza un descriptor de archivo para identificar un archivo abierto al realizar operaciones sobre el archivo. El descriptor de archivo se representa con la variable *fildes* en las funciones C que operan en el sistema de archivos integrado y con la variable *descriptor* en las funciones C que operan en zócalos.

Cada descriptor de archivo hace referencia a una **descripción de archivo abierto**, que contiene información como, por ejemplo, desplazamiento de archivo, estado del archivo y modalidades de acceso para el archivo. Varios descriptores de archivo pueden hacer referencia a la misma descripción de archivo abierto, pero un descriptor de archivo sólo puede hacer referencia a una descripción de archivo abierto.

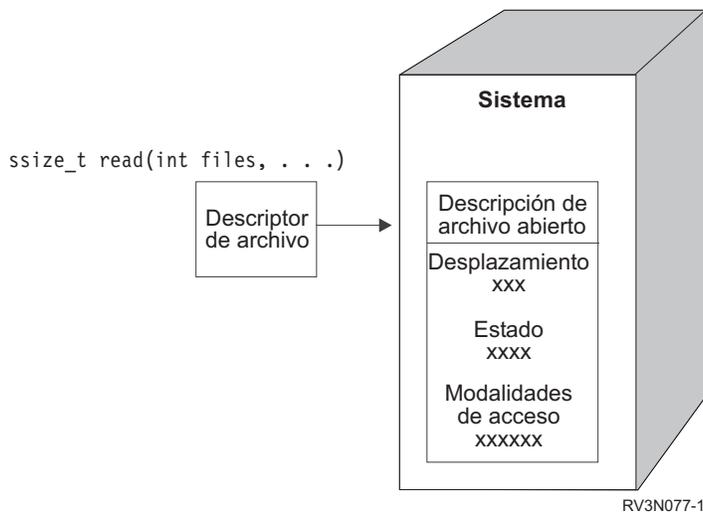


Figura 11. Descriptor de archivo y descripción de archivo abierto

Si se utiliza una función de E/S continua de ILE C/400 con el sistema de archivos integrado, el soporte de tiempo de ejecución de ILE C/400 convierte el puntero de archivo en un descriptor de archivo.

Cuando se utilizan los sistemas de archivos “raíz” (/), QOpenSys o el definido por el usuario, se puede pasar el acceso a una descripción de archivo abierto de un trabajo a otro, con lo que se permite al trabajo acceder al archivo. Esta operación se efectúa utilizando las funciones `givedescriptor()` o `takedescriptor()` para pasar el descriptor de archivos entre trabajos. Para obtener una descripción de estas funciones, consulte los apartados Programación de zócalos o API de zócalos en iSeries Information Center.

Seguridad

Al utilizar las API del sistema de archivos integrado, puede restringir el acceso a los objetos del mismo modo que al utilizar las interfaces de gestión de datos. Tenga en cuenta, de todos modos, que no se dispone de soporte para adoptar autorizaciones. Una API del sistema de archivos integrado utiliza la autorización del perfil de usuario con el que se ejecuta el trabajo.

Cada sistema de archivos puede tener sus propios requisitos de autorizaciones especiales. Los trabajos de servidor NFS constituyen la única excepción de esta regla. Las peticiones de servidor de sistema de archivos de red se ejecutan bajo el perfil del usuario cuyo número de identificación (UID) se recibió en el servidor NFS en el momento de la petición.

Las autorizaciones del servidor son equivalentes a los **permisos** de los sistemas UNIX. Los tipos de permisos son leer y grabar (para un archivo o un directorio) y ejecutar (para un archivo) o buscar (para un directorio). Los permisos se indican por medio de un conjunto de bits de permiso, que componen la “modalidad de acceso” del archivo o directorio. Se pueden cambiar los bits de permiso utilizando las funciones de “cambiar modalidad” `chmod()` o `fchmod()`. También se puede utilizar la función `umask()` para controlar qué bits de permiso de archivo se establecen cada vez que un trabajo crea un archivo.

Hallará detalles acerca de la seguridad de los datos y de las autorizaciones en la publicación Security —

Reference  .

Soporte de zócalos

Si la aplicación está utilizando los sistemas de archivos “raíz” (/), QOpenSys o definidos por el usuario, puede aprovechar el soporte de **zócalo local** de sistema de archivos integrado. Un objeto zócalo local (tipo de objeto *zócalo) permite que dos trabajos que se ejecutan en el mismo sistema establezcan una conexión de comunicaciones entre sí.

Uno de los trabajos establece un punto de conexión utilizando la función `bind()` del lenguaje C para crear un objeto zócalo local. El otro trabajo especifica el nombre del objeto zócalo local en la función `connect()`, `sendto()` o `sendmsg()`. Estas funciones y el concepto de zócalos en general se describen en la sección sobre Programación de zócalos de iSeries Information Center.

Una vez establecida la conexión, los dos trabajos pueden enviar datos y recibir datos mutuamente utilizando las funciones del sistema de archivos integrado, como pueden ser `write()` y `read()`. Ninguno de los datos que se transfieren pasan en realidad por el objeto zócalo. El objeto zócalo es simplemente un punto de encuentro donde los trabajos se localizan mutuamente.

Cuando los dos trabajos acaban la comunicación, cada uno de ellos utiliza la función `close()` para cerrar la conexión del zócalo. El objeto zócalo local permanece en el sistema hasta que se elimina con la función `unlink()` o con el mandato Eliminar enlace (RMVLNK).

Un objeto zócalo local no se puede salvar.

Soporte internacional y asignación de nombres

El soporte para los sistemas de archivos “raíz” (/) y QOpenSys garantiza que los caracteres de los nombres de objetos se mantienen constantes a través de los esquemas de codificación utilizados en distintos idiomas y dispositivos. Cuando al sistema se le pasa un nombre de objeto, cada carácter del nombre se convierte en un formato de 16 bits en el que todos los caracteres tienen una representación codificada estándar (consulte el apartado “Continuidad de nombres” en la página 23). Cuando se utiliza el nombre, se convierte en el formato codificado adecuado para la página de códigos que se utiliza.

Si la página de códigos a la que se está convirtiendo el nombre no contiene un carácter utilizado en un nombre, el nombre se rechaza por no válido.

Debido a que los caracteres se mantienen constantes a través de las páginas de códigos, no debe efectuar ninguna operación presuponiendo que un carácter concreto se cambiará por otro carácter concreto cuando se utilice una página de códigos específica. Por ejemplo, no debe presuponer que el carácter de signo numérico se cambiará por el carácter de libra esterlina aunque puedan tener la misma representación codificada en páginas de códigos diferentes.

Observe que los nombres de los atributos ampliados de un objeto se convierten del mismo modo que el nombre del objeto, por lo tanto, son aplicables las mismas consideraciones.

Para obtener más información sobre las páginas de código, consulte la sección dedicada a la Globalización del iSeries Information Center.

Conversión de datos

Al acceder a los archivos mediante la sistema de archivos integrado, los datos de los archivos se pueden convertir o no, según la modalidad de apertura solicitada al abrir el archivo.

Un archivo abierto puede estar en una de las dos modalidades de apertura siguientes:

Binario

Los datos se leen del archivo y se graban en él sin convertirlos. La aplicación es responsable del manejo de los datos.

Texto Los datos se leen del archivo y se graban en él, suponiendo que estén en un formato de texto. Cuando los datos se leen del archivo, se convierten del identificador de juego de caracteres (CCSID) del archivo al CCSID de la aplicación, trabajo o sistema que recibe los datos. Cuando los datos se graban en el archivo, se convierten del CCSID de la aplicación, trabajo o sistema al CCSID del archivo. En los archivos continuos reales, todos los caracteres de formato de línea (como por ejemplo, el retorno de carro, el tabulador o el carácter de fin de archivo) se convierten de un CCSID a otro.

Cuando se lee de archivos de registros que se están utilizando como archivos continuos, al final de los datos de cada registro se añaden los caracteres de fin de línea (retorno de carro y salto de línea). Cuando se graba en archivos de registros:

- Los caracteres de fin de línea se eliminan.
- Los caracteres tabulador se sustituyen por el número adecuado de espacios en blanco hasta la siguiente posición del tabulador.
- Las líneas se rellenan con espacios en blanco (para un miembro de archivo físico fuente) o con nulos (para un miembro de archivo físico de datos) hasta el final del registro.

En una petición de apertura, se puede especificar una de las opciones siguientes:

Binario, Forzado

Los datos se procesan como binarios independientemente del contenido real del archivo. La aplicación es responsable de saber cómo manejar los datos.

Texto, Forzado

Se presupone que los datos son de tipo texto. Los datos se convierten del CCSID del archivo al CCSID de la aplicación.

Para la función `open()` del sistema de archivos integrado se utiliza el valor por omisión *Binario, Forzado*.

Capítulo 6. Sistemas de archivos en el sistema de archivos integrado

Los sistemas de archivos del sistema de archivos integrado son:

- “raíz”
- QOpenSys
- UDFS
- QSYS.LIB
- QSYS.LIBde ASP independiente
- QDLS
- QOPT
- QNetWare
- QNTC
- QFileSvr.400
- NFS

Para obtener una visión general de cada sistema de archivos, consulte Comparación de los sistemas de archivos.

Comparación de los sistemas de archivos

La Tabla 4 y Tabla 5 en la página 63 resumen las características y las limitaciones de cada sistema de archivos.

Tabla 4. Resumen de los sistemas de archivos (Parte 1 de 2)

Posibilidad	/ (“raíz”)	QOpenSys	QSYS.LIB ¹⁶	QDLS	QNTC
Componente estándar del OS/400	Sí	Sí	Sí	Sí	Sí
Tipo de archivo	Continuo	Continuo	Registro ¹²	Continuo	Continuo
Integrado con OfiVisión (por ejemplo, el archivo se puede enviar por correo)	No	No	No	Sí	No
Acceso a través del servidor de archivos OS/400	Sí	Sí	Sí	Sí	Sí
Acceso directo a través del FSIOP ¹	No	No	No	No	Sí
Velocidad comparativa de abrir/cerrar	Media ²	Media ²	Baja ²	Baja ²	Media ²
Búsqueda de nombre sensible a las mayúsculas y minúsculas	No	Sí	No ⁴	No ⁵	No
Longitud máxima de cada componente en el nombre de vía de acceso	255 cars.	255 cars.	10.6 cars. ⁶	8.3 cars. ⁷	255 cars.
Longitud máxima de nombre de vía de acceso ⁸	16MB	16MB	55 – 66 cars. ⁴	82 cars.	255 cars.
Longitud máxima de los atributos ampliados de un objeto	2 Gb	2 Gb	Variable ⁹	32 Kb	64KB
Niveles máximos de la jerarquía de directorios dentro del sistema de archivos	Sin límite ¹⁰	Sin límite ¹⁰	3	32	127
Número máximo de enlaces por objeto ¹¹	Variable ¹⁵	Variable ¹⁵	1	1	1
Soporte de enlaces simbólicos	Sí	Sí	No	No	No

Tabla 4. Resumen de los sistemas de archivos (Parte 1 de 2) (continuación)

Posibilidad	/ ("raíz")	QOpenSys	QSYS.LIB ¹⁶	QDLS	QNTC
Objeto/archivo puede tener propietario	Sí	Sí	Sí	Sí	No
Soporte de mandatos del sistema de archivos integrado	Sí	Sí	Sí	Sí	Sí
Soporte de las API del sistema de archivos integrado	Sí	Sí	Sí	Sí	Sí
Soporte de las API del sistema de archivos jerarquizado (HFS)	No	No	No	Sí	No
Admite ejecución en hebras ¹³	Sí	Sí	Sí	No	Sí
Admite el registro por diario de objetos	Sí	Sí	Sí ¹⁴	No	No

Tabla 4. Resumen de los sistemas de archivos (Parte 1 de 2) (continuación)

Posibilidad	/ (“raíz”)	QOpenSys	QSYS.LIB ¹⁶	QDLS	QNTC
Notas:					
<ol style="list-style-type: none"> 1. El FSIOP es un hardware utilizado por el LAN Server. 2. Cuando se accede a través del servidor de archivos OS/400. 3. Cuando se accede a través de un PC cliente de LAN Server. El acceso utilizando las API del iSeries es comparativamente lento. 4. El sistema de archivos QSYS.LIB tiene una longitud máxima de nombre de ruta de 55 caracteres. Consulte el apartado “Sistema de archivos de biblioteca (QSYS.LIB)” en la página 72 para obtener información más detallada. El sistema de archivos ASP independiente QSYS.LIB tiene una longitud máxima de ruta de 66 caracteres. Consulte el apartado “QSYS.LIB de ASP independiente” en la página 75 para obtener información más detallada. 5. Consulte el apartado “Sistema de archivos de servicios de biblioteca de documentos (QDLS)” en la página 78 para obtener información más detallada. 6. Un máximo de 10 caracteres en el nombre de objeto y de 6 caracteres en el tipo de objeto. Consulte el apartado “Sistema de archivos de biblioteca (QSYS.LIB)” en la página 72 para conocer más detalles. 7. Un máximo de 8 caracteres en el nombre y de 1 a 3 caracteres en la extensión de tipo de archivo (si procede). Consulte el apartado “Sistema de archivos de servicios de biblioteca de documentos (QDLS)” en la página 78 para obtener información más detallada. 8. Suponiendo un nombre de vía de acceso absoluto que empiece con / seguido del nombre del sistema de archivos (como, por ejemplo, /QDLS...). 9. Los sistemas de archivos QSYS.LIB y ASP independiente QSYS.LIB ofrecen soporte para tres atributos ampliados predefinidos: .SUBJECT, .CODEPAGE y .TYPE. La longitud máxima queda determinada por la longitud combinada de estos tres atributos ampliados. 10. En la práctica, los niveles de directorios están limitados por los límites de espacio del programa y del sistema. 11. Excepto un directorio, que sólo puede tener un enlace con otro directorio. 12. Los espacios de usuario de QSYS.LIB y los sistemas de archivo ASP independiente QSYS.LIB ofrecen soporte para la entrada y salida de archivo continuo. 13. Las API del sistema de archivos integrado admiten la ejecución en hebras cuando la operación se dirige a un objeto que reside en un sistema de archivos que admite ejecución en hebras. Si estas API están operando sobre objetos en sistemas de archivos que no admiten la ejecución en hebras cuando en el trabajo se ejecutan múltiples hebras, la API fallará. 14. QSYS.LIB y los sistemas de archivo ASP independiente QSYS.LIB pueden dar soporte al registro por diario de objetos distintos de los sistemas de archivos raíz, UDFS y QOpenSys. Consulte el tema Administración de diarios del iSeries Information Center para obtener más información sobre el registro en diario de objetos que residen en QSYS.LIB o sistemas de archivos ASP independientes QSYS.LIB. 15. Los directorios *TYPE2 tienen un límite de un millón de enlaces por objeto. Los directorios *TYPE1 tienen un límite de 32, 767 enlaces por objeto. Para obtener más información, consulte los directorios *TYPE2. 16. Los datos de esta columna se refieren tanto al sistema de archivos QSYS.LIB, como al sistema de archivos QSYS.LIB de ASP independiente. 					
Abreviaturas					
cars. = caracteres					
B = bytes KB = kilobytes MB = megabytes GB = gigabytes					

Tabla 5. Resumen de los sistemas de archivos (Parte 2 de 2)

Posibilidad	QOPT	QFileSvr.400	UDFS	NFS	QNetWare
Componente estándar del OS/400	Sí	Sí	Sí	Sí	No
Tipo de archivo	Continuo	Continuo	Continuo	Continuo	Continuo

Tabla 5. Resumen de los sistemas de archivos (Parte 2 de 2) (continuación)

Posibilidad	QOPT	QFileSvr.400	UDFS	NFS	QNetWare
Integrado con OfiVisión (por ejemplo, el archivo se puede enviar por correo)	No	No	No	No	No
Acceso a través del servidor de archivos OS/400	Sí	Sí	Sí	Sí	Sí
Acceso directo a través del Servidor PC Integrado (FSIOP) ¹	No	No	No	No	Sí
Velocidad comparativa de abrir/cerrar	Baja	Baja ²	Media ²	Media ²	Alta ¹¹
Búsqueda de nombre sensible a las mayúsculas y minúsculas	No	No ²	Sí ¹²	Variable ²	No
Longitud máxima de cada componente en el nombre de vía de acceso	Variable ⁴	Variable ²	255 cars.	Variable ²	255 cars. ¹³
Longitud máxima de nombre de vía de acceso	294 cars.	Sin límite ²	16MB	Sin límite ²	255 cars.
Longitud máxima de los atributos ampliados de un objeto	8 Mb	0 ⁶	2GB ¹⁰	0 ⁶	64KB
Niveles máximos de la jerarquía de directorios dentro del sistema de archivos	Sin límite ⁷	Sin límite ²	Sin límite ⁷	Sin límite ²	100
Número máximo de enlaces por objeto ⁷	1	1	Variable ¹⁵	Variable ²	1
Soporte de enlaces simbólicos	No	No	Sí	Sí ²	No
Objeto/archivo puede tener propietario	No	No ⁹	Sí	Sí ²	Sí
Soporte de mandatos del sistema de archivos integrado	Sí	Sí	Sí	Sí	Sí
Soporte de las API del sistema de archivos integrado	Sí	Sí	Sí	Sí	Sí
Soporte de las API del sistema de archivos jerarquizado (HFS)	Sí	No	No	No ²	No
Admite ejecución en hebras ¹⁴	Sí	No	Sí	No	No
Admite el registro por diario de objetos	No	No	Sí	No	No

Tabla 5. Resumen de los sistemas de archivos (Parte 2 de 2) (continuación)

Posibilidad	QOPT	QFileSvr.400	UDFS	NFS	QNetWare
Notas:					
<ol style="list-style-type: none"> 1. El FSIOP es un hardware utilizado por el LAN Server. 2. Depende de a qué sistema de archivos remoto se esté accediendo. 3. Cuando se accede a través del servidor de archivos OS/400 4. Consulte el apartado "Sistema de archivos óptico (QOPT)" en la página 80 para obtener información más detallada. 5. Suponiendo que se trate de un nombre de vía de acceso absoluto que empiece por / seguido del nombre del sistema de archivos 6. El sistema de archivos QFileSvr.400 no devuelve atributos ampliados aunque el sistema de archivos al que se acceda soporte atributos ampliados. 7. En la práctica, los niveles de directorios están limitados por los límites de espacio del programa y del sistema. 8. Excepto un directorio, que sólo puede tener un enlace con otro directorio. 9. El sistema de archivos al que se accede puede que soporte propietarios de objetos. 10. La longitud máxima de los atributos ampliados de UDFS en sí no puede ser superior a 40 bytes. 11. Si se accede a través de un PC cliente Novell NetWare. El acceso utilizando las API del iSeries es comparativamente lento. 12. Al crear un UDFS, puede especificarse que distinga entre mayúsculas y minúsculas. Si se utiliza el parámetro *MIXED cuando se crea un UDFS, permitirá una búsqueda que distingue mayúsculas de minúsculas. 13. Los objetos de los servicios de directorio NetWare tienen un máximo de 255 caracteres. Los archivos y directorios están limitados al formato DOS 8.3. 14. Las API del sistema de archivos integrado admiten la ejecución en hebras cuando se accede a ellas en un proceso con posibilidad multihebra. El sistema de archivos no permite el acceso a los sistemas de archivos que no admiten la ejecución en hebras. 15. Los directorios *TYPE2 tienen un límite de un millón de enlaces por objeto. Los directorios *TYPE1 tienen un límite de 32.767 enlaces por objeto. Para obtener más información, consulte los directorios *TYPE2. 					
Abreviaturas					
cars. = caracteres					
B = bytes KB = kilobytes MB = megabytes GB = gigabytes					

sistema de archivos "raíz" (/)

El sistema de archivos "raíz" (/) obtiene el máximo provecho del soporte de archivos continuos y de la estructura jerárquica de directorios del sistema de archivos integrado. El sistema de archivo "raíz" (/) tiene las características de los sistemas de archivos DOS y OS/2.

Además:

- Está optimizado para la entrada/salida de archivos continuos.
- Soporta múltiples enlaces fijos y enlaces simbólicos.
- Soporta zócalos locales.
- Soporta objetos *OOPOOL.
- Soporta interfaces del programa de aplicación (API) que permiten la ejecución multihebra.
- Soporta objetos *FIFO.
- Ofrece soporte para los objetos *CHRSF /dev/null y /dev/zero, así como otros objetos *CHRSF.
- Ofrece soporte para el registro en diario de los cambios de los objetos.

El sistema de archivos "raíz" (/) tiene soporte para los archivos especiales de caracteres (*CHRSF) conocidos como /dev/null y /dev/zero. Los archivos especiales de caracteres están asociados con un

dispositivo o recurso de un sistema. Tienen nombres de vía de acceso que aparecen en directorios y tienen la misma protección de acceso que un archivo corriente. Los archivos especiales de caracteres /dev/null o /dev/zero siempre están vacíos, y los datos que se graben en /dev/null o /dev/zero se eliminarán. Los archivos /dev/null y /dev/zero tienen el tipo de objeto *CHRSF y pueden utilizarse como archivos normales, excepto que no es posible leer datos del archivo /dev/null, y que el archivo /dev/zero siempre devuelve los datos eliminados y sustituidos por ceros.

Para obtener más información sobre el sistema de archivos "raíz" (/), consulte Utilización del sistema de archivos "raíz" (/).

Utilización del sistema de archivos "raíz" (/)

Al sistema de archivos "raíz" (/) sólo se puede acceder a través de la interfaz del sistema de archivos integrado utilizando los mandatos del servidor de archivos OS/400 o del sistema de archivos integrado, las pantallas de usuario y las API del lenguaje C.

Mayúsculas y minúsculas en el sistema de archivos "raíz" (/)

El sistema de archivos conserva las mayúsculas y minúsculas con que se han escrito los nombres de objetos, pero no distingue entre mayúsculas y minúsculas cuando el servidor efectúa búsquedas de nombres.

Nombres de vía de acceso en el sistema de archivos "raíz" (/)

- Los nombres de vía de acceso tienen el formato siguiente:

Directorio/Directorio . . . /Objeto

- Los componentes de los nombres de vía de acceso que siguen al nombre compartido siguen las reglas de nombres de vía de acceso de Windows NT. La combinación de nombres de directorio y nombre de objeto puede ocupar una longitud máxima de 255 caracteres.
- No hay límite en la profundidad de la jerarquía de directorios más que los límites de espacio del programa y del servidor.
- Los caracteres de los nombres se convierten al formato UCS2 Nivel 1 (para directorios *TYPE1) y UTF-16 (para directorios *TYPE2) cuando se guardan los nombres (consulte "Continuidad de nombres" en la página 23). Consulte el apartado Directorios *TYPE2 para obtener más información sobre los formatos de directorios.

Enlaces en el sistema de archivos "raíz" (/)

Se permiten varios enlaces fijos con el mismo objeto del sistema de archivos "raíz" (/). Los enlaces simbólicos se soportan totalmente. Puede utilizarse un enlace simbólico para enlazar el sistema de archivos "raíz" (/) a un objeto de otro sistema de archivos, como QSYS.LIB, QSYS.LIB de ASP independiente, o QDLS.

Para ver la descripción de los enlaces, consulte el apartado "Enlace" en la página 18.

Utilización de mandatos del sistema de archivos integrado en el sistema de archivos "raíz" (/)

Todos los mandatos listados en el apartado "Realizar operaciones mediante mandatos CL" en la página 26 y las pantallas descritas en el apartado "Realizar operaciones utilizando los menús y pantallas del iSeries" en la página 25 pueden realizar operaciones en el sistema de archivos "raíz" (/). Sin embargo, puede que la ejecución de estos mandatos en un proceso con posibilidad multihebra no sea segura.

Utilización de las API del sistema de archivos integrado en el sistema de archivos "raíz" (/)

Todas las API del lenguaje C que se listan en el apartado "Realizar operaciones utilizando API" en la página 49 pueden operar en el sistema de archivos "raíz" (/) de un modo seguro en ejecución multihebra.

Cambios de objeto de diario en el sistema de archivos "raíz" (/)

Los objetos del sistema de archivos "raíz" (/) pueden registrarse por diario. La principal finalidad de la gestión por diario es permitir la recuperación de los cambios realizados en un objeto desde que se guardó

por última vez. Para obtener más información sobre el registro en diario de los cambios de los objetos en el sistema de archivos "raíz" (/), consulte Capítulo 7, "Soporte del registro por diario para los objetos del sistema de archivos integrado" en la página 95.

Sistema de archivos de sistemas abiertos (QOpenSys)

El sistema de archivos QOpenSys es compatible con los estándares de sistemas abiertos basados en UNIX como, por ejemplo, POSIX y XPG. Al igual que el sistema de archivos "raíz" (/), este sistema de archivos aprovecha el soporte de directorios y archivos continuos que facilita el sistema de archivos integrado.

Además:

- Se accede a él a través de una estructura jerárquica de directorios semejante a la de los sistemas UNIX.
- Está optimizado para la entrada/salida de archivos continuos.
- Soporta múltiples enlaces fijos y enlaces simbólicos.
- Soporta nombres sensibles a las mayúsculas y minúsculas.
- Soporta zócalos locales.
- Soporta objetos *OOPOOL.
- Soporta las API que permiten ejecución multihebra.
- Soporta objetos *FIFO.
- Ofrece soporte para el registro en diario de los cambios de los objetos.

El sistema de archivos QOpenSys tiene las mismas características que el sistema de archivos "raíz" (/), excepto que es sensible a las mayúsculas y minúsculas para habilitar el soporte para los estándares de sistemas abiertos basados en UNIX.

Para obtener más información sobre QOpenSys, consulte Utilización de QOpenSys.

Para obtener información sobre las conversiones de directorios de *TYPE1 a *TYPE2 y sobre las restricciones del sistema de archivos QOpenSys, consulte Directorios *TYPE2.

Utilización de QOpenSys

Al QOpenSys puede accederse a través de la interfaz del sistema de archivos integrado utilizando los mandatos del servidor de archivos OS/400 o sistema de archivos integrado, las pantallas de usuario y las API del lenguaje C.

Mayúsculas y minúsculas en el sistema de archivos QOpenSys

A diferencia del sistema de archivos "raíz" (/), el sistema de archivos QOpenSys distingue entre minúsculas y mayúsculas en las búsquedas de nombres de objeto. Por ejemplo, una serie de caracteres suministrada toda en mayúsculas no coincidirá con la misma serie de caracteres donde alguno de los caracteres esté en minúsculas.

Esta distinción de mayúsculas y minúsculas permite utilizar nombres duplicados, dado que existen diferencias en las mayúsculas y minúsculas de los caracteres que componen el nombre. Por ejemplo, puede tener un objeto denominado Sue1do, otro objeto denominado SuE1do y otro denominado SUELD0 en el mismo directorio de QOpenSys.

Nombres de vía de acceso en el sistema de archivos QOpenSys

- Los nombres de vía de acceso tienen el formato siguiente:

Directorio/Directorio/ . . . /Objeto

- Cada componente del nombre de vía de acceso puede tener una longitud máxima de 255 caracteres. El nombre completo de la vía de acceso puede tener una longitud máxima de 16 megabytes.

- No hay límite en la profundidad de la jerarquía de directorios más que los límites de espacio del programa y del servidor.
- Los caracteres de los nombres se convierten al formato UCS2 Nivel 1 (para directorios *TYPE1) y UTF-16 (para directorios *TYPE2) cuando se guardan los nombres (consulte “Continuidad de nombres” en la página 23). Consulte el apartado Directorios *TYPE2 para obtener más información sobre los formatos de directorios.

Enlaces en el sistema de archivos QOpenSys

Se permiten múltiples enlaces fijos con el mismo objeto del sistema de archivos QOpenSys. Los enlaces simbólicos se soportan totalmente. Se puede utilizar un enlace simbólico para tener un enlace desde el sistema de archivos QOpenSys con un objeto de otro sistema de archivos.

Consulte el apartado “Enlace” en la página 18 para ver la descripción de los enlaces.

Utilización de los mandatos y pantallas del sistema de archivos integrado en el sistema de archivos QOpenSys

Todos los mandatos listados en el apartado “Realizar operaciones mediante mandatos CL” en la página 26 y las pantallas descritas en el apartado “Realizar operaciones utilizando los menús y pantallas del iSeries” en la página 25 pueden realizar operaciones en el sistema de archivos QOpenSys. Sin embargo, puede que la ejecución de estos mandatos en un proceso que admite la ejecución multihebra no sea segura.

Utilización de las API del sistema de archivos integrado en el sistema de archivos QOpenSys

Todas las funciones del lenguaje C que se listan en el apartado “Realizar operaciones utilizando API” en la página 49 pueden realizar operaciones en el sistema de archivos QOpenSys de forma que permita la ejecución multihebra.

Registro por diario de los cambios de los objetos en el sistema de archivos QOpenSys

Los objetos del sistema de archivos QOpenSys pueden registrarse por diario. La principal finalidad de la gestión por diario es permitir la recuperación de los cambios realizados en un objeto desde que se guardó por última vez. Para obtener más información sobre el registro en diario de los cambios de los objetos en el sistema de archivos QOpenSys, consulte Capítulo 7, “Soporte del registro por diario para los objetos del sistema de archivos integrado” en la página 95.

Sistema de archivos definido por el usuario (UDFS)

Los sistemas de archivos UDFS residen en las agrupaciones de almacenamiento auxiliar (ASP) o las agrupaciones de almacenamiento auxiliar independientes que se elija. Estos sistemas de archivos los crea y los gestiona el usuario.

Además:

- Proporcionan una estructura jerárquica de directorios semejante a la de los sistemas operativos de PC como, por ejemplo, DOS y OS/2.
- Están optimizados para la entrada/salida de archivos continuos.
- Soportan múltiples enlaces fijos y enlaces simbólicos.
- Soportan zócalos locales.
- Soporta las API que permiten ejecución multihebra.
- Soporta objetos *FIFO.
- Soporta el registro en diario de los cambios de los objetos.

Pueden crearse varios UDFS dándole a cada uno un nombre exclusivo. Durante su creación, pueden especificarse diversos atributos del UDFS, entre otros:

- Un número de ASP o un nombre de ASP independiente en los que se almacenan los objetos situados en el UDFS.

- Las características de distinción entre mayúsculas y minúsculas de los nombres de objeto situados en el UDFS.

La distinción entre mayúsculas y minúsculas de un UDFS determina si los caracteres en mayúsculas y en minúsculas coincidirán al realizar búsquedas de nombres de objeto dentro del UDFS.

Consulte los temas siguientes para obtener más información sobre los sistemas de archivos definidos por el usuario:

- Conceptos de los UDFS
- Utilización de los UDFS mediante la la interfaz del sistema de archivos integrado

Conceptos de los UDFS

En un UDFS, al igual que ocurre en los sistemas de archivos “raíz” (/) y QOpenSys, pueden crearse directorios, archivos continuos, enlaces simbólicos, zócalos locales y objetos SOM.

Un objeto archivo especial de bloqueo (*BLKSF) representa un UDFS. A medida que se van creando UDFS, también se van creando automáticamente archivos especiales de bloqueo. Sólo puede accederse al archivo especial de bloqueo mediante los mandatos genéricos del sistema de archivos integrado, las API y la interfaz QFileSvr.400.

Un UDFS existe únicamente en dos estados: **montado** y **desmontado**. Cuando se monta un UDFS, puede accederse a los objetos que éste contiene. Cuando se desmonta un UDFS, los objetos que éste contiene dejan de ser accesibles.

Para poder acceder a los objetos de un UDFS, debe montarse este último en un directorio (por ejemplo, /home/JUAN). Cuando monte un UDFS en un directorio, el contenido original de dicho directorio, incluyendo los objetos y los subdirectorios, quedan inaccesibles. El contenido del UDFS se vuelve accesible a través de la vía de acceso del directorio en el que se monta el UDFS. Por ejemplo, el directorio /home/JUAN contiene un archivo /home/JUAN/sueldo. Un UDFS contiene tres directorios correo, acciones y salida. Después de montar el UDFS en /home/JUAN, el archivo /home/JUAN/sueldo es inaccesible y tres directorios de UDFS pasan a estar accesibles como /home/JUAN/correo, /home/JUAN/acciones, y /home/JUAN/salida. Después de desmontar el UDFS, se puede acceder de nuevo al archivo /home/JUAN/sueldo y no a los tres directorios del UDFS.

Nota: No es posible montar un UDFS en una ASP independiente.

Para obtener más información acerca del montaje de sistemas de archivos, consulte la publicación

OS/400 Network File System Support  .

Utilización de los UDFS mediante la la interfaz del sistema de archivos integrado

Es posible acceder a un UDFS a través de la interfaz del sistema de archivos integrado utilizando el servidor de archivos del OS/400 o los mandatos, las pantallas de usuario y las API del sistema de archivos integrado. En lo referente al uso de la interfaz del sistema de archivos integrado, deberá tener en cuenta las consideraciones y limitaciones siguientes.

Mayúsculas y minúsculas en un sistema de archivos integrado UDFS

Se puede especificar si los nombres de objeto del UDFS que se van a crear serán sensibles a las mayúsculas y minúsculas o no.

Si selecciona que lo sean, al buscar nombres de objetos se distinguirá entre caracteres en mayúsculas y caracteres en minúsculas. Por ejemplo, si se facilita un nombre todo en mayúsculas, no coincidirá con el mismo nombre con alguno de los caracteres en minúsculas. Por lo tanto, /home/MURPH/ y /home/murph/ se

reconocen como directorios distintos. Para crear un UDFS sensible a las mayúsculas y minúsculas, puede especificar *MIXED para el parámetro CASE al utilizar el mandato CRTUDFS.

Si se selecciona que no sea sensible a mayúsculas y minúsculas, el servidor no hará distinción entre unas y otras al realizar búsquedas de nombres. Por lo tanto, el servidor reconocerá /home/MAYTE y /HOME/mayte como el mismo directorio, no como dos directorios distintos. Para crear un UDFS no sensible a las mayúsculas y minúsculas, puede especificar *MONO para el parámetro CASE al utilizar el mandato CRTUDFS.

En cualquier caso, el sistema de archivos guarda el formato de mayúsculas y minúsculas con el que el usuario ha entrado los nombres de objeto. La opción de distinguir entre mayúsculas y minúsculas sólo se aplica al modo en que el usuario busca los nombres en el servidor.

Nombres de vía de acceso en un sistema de archivos integrado UDFS

Un objeto archivo especial de bloqueo (*BLKSF) representa un UDFS cuando la totalidad del UDFS y todos los objetos que contiene han de manipularse. Si el UDFS reside en una ASP de usuario básica, los nombres de archivo especial de bloqueo deben tener el formato

```
/dev/QASPXX/nombre_udfs.udfs
```

donde XX es el número de la ASP donde se almacena el UDFS y nombre_udfs es el nombre exclusivo del UDFS dentro de dicha ASP. Observe que el nombre del UDFS debe terminar con la extensión .udfs.

Si el UDFS reside en una ASP independiente, los nombres de archivo especial de bloqueo deben tener el formato

```
/dev/asp_name/udfs_name.udfs
```

siendo asp_name el nombre de la ASP independiente en que se almacena el UDFS y udfs_name el nombre exclusivo del UDFS dentro de dicha ASP independiente. Observe que el nombre del UDFS debe terminar con la extensión .udfs.

Los nombres de vía de acceso para objetos en un UDFS son relativos al directorio sobre el que se monta un UDFS. Por ejemplo, si se monta el UDFS /dev/qasp01/wysocki.udfs en /home/daniel, los nombres de vía de acceso de todos los objetos del UDFS empezarán por /home/daniel.

Reglas sobre nombres de vía de acceso adicionales:

- Cada componente del nombre de vía de acceso puede tener una longitud máxima de 255 caracteres. El nombre completo de la vía de acceso puede tener una longitud máxima de 16 megabytes.
- No hay límite en la profundidad de la jerarquía de directorios más que los límites de espacio del programa y del servidor.
- Los caracteres de los nombres se convierten al formato UCS2 Nivel 1 (para directorios *TYPE1) y UTF-16 (para directorios *TYPE2) cuando se guardan los nombres (consulte “Continuidad de nombres” en la página 23). Consulte el apartado Directorios *TYPE2 para obtener más información sobre los formatos de directorios.

Enlaces en un sistema de archivos integrado UDFS

Los objetos de un UDFS permiten múltiples enlaces fijos con el mismo objeto y soportan plenamente los enlaces simbólicos. Un enlace simbólico puede crear un enlace de un UDFS a un objeto de otro sistema de archivos.

Consulte el apartado “Enlace” en la página 18 para ver la descripción de los enlaces.

Utilización de los mandatos de sistema de archivos integrado en un UDFS

Todos los mandatos listados en el apartado “Realizar operaciones mediante mandatos CL” en la página 26 y las pantallas descritas en el apartado “Realizar operaciones utilizando los menús y pantallas del iSeries” en la página 25 pueden realizar operaciones en un sistema de archivos definido por el usuario. Existen otros mandatos CL que son específicos para el sistema de archivos definido por el usuario y otros

sistemas de archivos montados en general. Están descritos en la tabla siguiente.

Tabla 6. Mandatos CL del sistema de archivos definido por el usuario

Mandato	Descripción
ADDMFS	Añadir sistema de archivos montado. Coloca los sistemas de archivos de servidor remoto exportados en directorios de cliente locales.
CRTUDFS	Crear UDFS. Crea un sistema de archivos definido por el usuario.
DLTUDFS	Suprimir UDFS. Suprime un sistema de archivos definido por el usuario.
DSPMFSINF	Visualizar información de sistema de archivos montado. Visualiza información acerca de un sistema de archivos montado.
DSPUDFS	Visualizar UDFS. Visualiza información acerca de un sistema de archivos definido por el usuario.
MOUNT	Montar sistema de archivos. Coloca los sistemas de archivos de servidor remoto exportados en directorios de cliente locales. Es un alias del mandato ADDMFS.
RMVMFS	Eliminar sistema de archivos montado. Elimina los sistemas de archivos de servidor remoto exportados del espacio de nombres del cliente local.
UNMOUNT	Desmontar sistema de archivos. Elimina los sistemas de archivos de servidor remoto exportados del espacio de nombres del cliente local. Es un alias del mandato RMVMFS.

Nota: para que los mandatos del sistema de archivos integrado puedan realizar operaciones sobre los objetos almacenados en un UDFS, primero debe montar éste.

Utilización de las API del sistema de archivos integrado en un UDFS

Todas las funciones del lenguaje C que se listan en el apartado “Realizar operaciones utilizando API” en la página 49 pueden realizar operaciones en un sistema de archivos definido por el usuario.

Nota: para que los mandatos del sistema de archivos integrado puedan realizar operaciones sobre los objetos almacenados en un UDFS, primero debe montar éste.

Interfaz gráfica de usuario para un UDFS

El Navigator del iSeries, una interfaz gráfica de usuario en el PC, proporciona un acceso sencillo y cómodo a UDFS. Dicha interfaz permite crear, suprimir, visualizar, montar y desmontar un UDFS desde un cliente Windows.

Pueden realizarse operaciones en un UDFS mediante el Navigator del iSeries. Entre las tareas básicas se incluyen:

- “Crear un sistema de archivos definido por el usuario” en la página 41.
- “Montar un sistema de archivos definido por el usuario” en la página 41.
- “Desmontar un sistema de archivos definido por el usuario” en la página 42.

Crear un sistema de archivos integrado UDFS

El mandato Crear sistema de archivos definido por el usuario (CRTUDFS) crea un sistema de archivos que puede verse por medio de los mandatos CL, las API y el espacio de nombres del sistema de archivos integrado. Los mandatos ADDMFS o MOUNT colocan el UDFS “encima” del directorio local ya existente. Puede crearse un UDFS en la ASP o ASP independiente que se prefiera. También puede especificarse la distinción entre mayúsculas y minúsculas.

Suprimir un sistema de archivos integrado UDFS

El mandato Suprimir sistema de archivos definido por el usuario (DLTUDFS) suprime un UDFS no montado existente y todos los objetos incluidos en él. El mandato no funcionará si el UDFS está montado.

La supresión de un UDFS provocará la supresión de todos los objetos del mismo. Si no tiene la autorización adecuada para suprimir todos los objetos de un UDFS, no se suprime ningún objeto.

Visualizar un sistema de archivos integrado UDFS

El mandato Visualizar sistema de archivos definido por el usuario (DSPUDFS) presenta los atributos de un UDFS existente, tanto si está montado como si no. El mandato Visualizar información de sistema de archivos montado (DSPMFSINF) también presentará información acerca de un UDFS montado y de cualquier otro sistema de archivos montado.

Montar un sistema de archivos integrado UDFS

Los mandatos Añadir sistema de archivos montado (ADDMFS) y MOUNT hacen accesibles los objetos de un sistema de archivos al espacio de nombres del sistema de archivos integrado. Para montar un UDFS, es necesario especificar *UDFS para el parámetro TYPE del mandato ADDMFS.

Nota: No es posible montar un UDFS en una ASP independiente.

Desmontar un sistema de archivos integrado UDFS

El mandato UNMOUNT hace que el contenido de un UDFS resulte inaccesible para las interfaces del sistema de archivos integrado. Una vez desmontado un UDFS, no se podrá acceder a los objetos del mismo de forma individual. Los mandatos Eliminar sistema de archivos montado (RMVMFS) o UNMOUNT hacen inaccesible un sistema de archivos montado al espacio de nombres del sistema de archivos integrado. Si se está utilizando alguno de los objetos del sistema de archivos (por ejemplo, hay un archivo abierto) en el momento de la utilización del mandato, recibirá un mensaje de error. El UDFS permanecerá montado. Si se monta algo encima de una parte del UDFS, éste no podrá desmontarse hasta que quede descubierto.

Por ejemplo, se monta el UDFS /dev/qasp02/jenn.udfs sobre /home/judy en el espacio de nombres del sistema de archivos integrado. Si, a continuación, se monta otro sistema de archivos, /pubs encima de /home/judy, el contenido de jenn.udfs dejará de ser accesible. Además, no podrá desmontar jenn.udfs hasta que desmonte el segundo sistema de archivos desde /home/judy.

Nota: No es posible montar un UDFS en una ASP independiente.

Guardar y restaurar el UDFS de un sistema de archivos integrado

Se tiene la posibilidad de salvar y restaurar todos los objetos del UDFS, así como las autorizaciones asociadas al mismo. El mandato Salvar (SAV) le permite salvar objetos de un UDFS, mientras que el mandato Restaurar (RST) le permite restaurar objetos en un UDFS. Ambos mandatos funcionarán tanto si el UDFS está montado como si no lo está. Sin embargo, para guardar correctamente los atributos del UDFS, y no sólo los objetos dentro del UDFS, es preciso desmontar el UDFS.

Registro por diario de los cambios de los objetos en un sistema de archivos UDFS

Es posible registrar por diario los objetos de los sistemas de archivos definidos por el usuario. La principal finalidad de la gestión por diario es permitir la recuperación de los cambios realizados en un objeto desde que se guardó por última vez. Para obtener más información sobre el registro en diario de los cambios de los objetos en un sistema de archivos UDFS, consulte Capítulo 7, "Soporte del registro por diario para los objetos del sistema de archivos integrado" en la página 95.

Sistema de archivos de biblioteca (QSYS.LIB)

El sistema de archivos QSYS.LIB da soporte a la estructura de bibliotecas iSeries server. Este sistema de archivos permite acceder a los archivos de base de datos y al resto de tipos de objeto del iSeries server que gestiona el soporte de bibliotecas en las ASP de usuario y del sistema.

Además:

- Soporta todas las interfaces de usuario e interfaces de programación que realizan operaciones sobre las bibliotecas del iSeries server y sobre los objetos de dichas bibliotecas.
- Soporta todos los lenguajes y recursos de programación que operan sobre archivos de base de datos.

- Ofrece un amplio soporte de administración para gestionar los objetos del servidor iSeries.
- Soporta operaciones de E/S continuas en miembros de archivos físicos, espacios de usuario y archivos de salvar.

Antes de la Versión 3 del OS/400, al sistema de archivos QSYS.LIB probablemente se le habría llamado *el* sistema de archivos del servidor iSeries. Los programadores que utilizaban lenguajes tales como RPG o COBOL y recursos como las DDS para desarrollar aplicaciones estaban utilizando el sistema de archivos QSYS.LIB. Los operadores de sistemas que utilizaban mandatos, menús y pantallas para manipular colas de salida estaban utilizando el sistema de archivos QSYS.LIB, al igual que los administradores de sistemas que creaban y cambiaban perfiles de usuario.

Todos estos recursos y las aplicaciones basadas en ellos funcionan igual que antes de incorporar el sistema de archivos integrado. Sin embargo, dichos recursos no pueden acceder a QSYS.LIB a través de la la interfaz del sistema de archivos integrado.

Para obtener más información sobre QSYS.LIB, consulte Utilización de QSYS.LIB mediante la interfaz del sistema de archivos integrado.

Utilización de QSYS.LIB mediante la interfaz del sistema de archivos integrado

Al sistema de archivos QSYS.LIB puede accederse mediante la la interfaz del sistema de archivos integrado utilizando los mandatos del servidor de archivos OS/400 o sistema de archivos integrado, las pantallas de usuario y las API del lenguaje C. En lo que respecta al uso de dichas interfaces del sistema de archivos integrado, debe tener en cuenta las consideraciones y limitaciones siguientes.

Lista de autorizaciones QPWFSEVER en el sistema de archivos QSYS.LIB

QPWFSEVER es una lista de autorizaciones (el tipo de objeto es *AUTL) que proporciona requisitos de acceso adicionales a todos los objetos que se encuentran en el sistema de archivos QSYS.LIB a los que se acceda mediante clientes remotos. Las autorizaciones de la lista de autorizaciones se aplican a todos los objetos del sistema de archivos QSYS.LIB.

La autorización por omisión de este objeto es la autorización PUBLIC *USE. El administrador puede utilizar los mandatos EDTAUTL (Editar lista de autorizaciones) o WRKAUTL (Trabajar con lista de autorizaciones) para modificar el valor de esta autorización. El administrador puede asignar la autorización PUBLIC *EXCLUDE a la lista de autorizaciones de tal forma que el público general no pueda acceder a los objetos de QSYS.LIB desde clientes remotos.

Restricciones en el manejo de archivos en el sistema de archivos QSYS.LIB

- No se da soporte a archivos lógicos.
- Los archivos físicos soportados para el acceso en modalidad de texto son los archivos físicos descritos por programa que contienen un solo campo y los archivos físicos fuente que contienen un solo campo de texto. Los archivos físicos soportados para el acceso en modalidad binaria incluyen los archivos físicos descritos externamente además de los archivos soportados para acceso en modalidad de texto.
- No se da soporte al bloqueo de rango de bytes. (Si desea obtener más información sobre el bloqueo de rango de bytes, consulte el tem fcntl() de iSeries Information Center.)
- Si algún trabajo tiene abierto un miembro de archivo de base de datos, sólo un trabajo tiene acceso de grabación a ese miembro de archivo en todo momento. Las demás peticiones disponen de acceso de sólo lectura.

Soporte de espacios de usuario en el sistema de archivos QSYS.LIB

QSYS.LIB soporta operaciones de E/S continuas en objetos espacio de usuario. Por ejemplo, un programa puede grabar datos continuos en un espacio de usuario y leer datos de un espacio de usuario. El tamaño máximo de un espacio de usuario es 16 776 704 bytes.

Tenga en cuenta que los espacios de usuario no tienen CCSID (identificador de juego de caracteres). Por consiguiente, el CCSID devuelto es el CCSID por omisión del trabajo.

Soporte de archivos de salvar en el sistema de archivos de QSYS.LIB

El sistema de archivos QSYS.LIB admite las operaciones de E/S continuas para guardar objetos de archivo. Por ejemplo, un archivo de salvar existente tiene datos que pueden leerse o copiarse a otro archivo hasta que sea necesario poner los datos en un tercer objeto archivo de salvar ya existente y vacío. Cuando un archivo de salvar se abre para grabación, no se permiten otras instancias de abrir del archivo. Un archivo de salvar **permite** que existan varias instancias de abrir para lectura, a condición de que ningún trabajo tenga más de una instancia de abrir del archivo para lectura. Un archivo de salvar no puede abrirse para acceso de lectura-grabación. Las operaciones de E/S continuas para guardar los datos de un archivo no están permitidas si un trabajo se ejecuta en varias hebras.

Las operaciones de E/S continuas en un archivo de salvar no están permitidas si el archivo de salvar o su directorio van a exportarse a través del servidor del sistema de archivos de red. No obstante, sí que pueden accederse desde clientes PC y a través del sistema de archivos QFileSvr.400.

Mayúsculas y minúsculas en el sistema de archivos QSYS.LIB

En general, el sistema de archivos QSYS.LIB no distingue las mayúsculas y minúsculas de los nombres de objetos. Una búsqueda de nombres de objetos obtiene el mismo resultado independientemente de que los caracteres de los nombres estén en mayúsculas o en minúsculas.

Sin embargo, si un nombre está delimitado por comillas, se conservan las mayúsculas y minúsculas de los caracteres. Una búsqueda de nombres entre comillas, por lo tanto, es sensible a las mayúsculas y minúsculas de los caracteres del nombre entre comillas.

Nombres de vía de acceso en el sistema de archivos QSYS.LIB

- Cada componente del nombre de vía de acceso debe contener el nombre del objeto seguido de su tipo de objeto. Por ejemplo:

`/QSYS.LIB/QGPL.LIB/PRT1.OUTQ`

`/QSYS.LIB/EMP.LIB/PAY.FILE/TAX.MBR`

El nombre del objeto y el tipo del objeto se separan con un punto (.). Los objetos de una biblioteca pueden tener el mismo nombre si tienen tipos de objeto distintos, por lo que el tipo de objeto debe especificarse para identificar exclusivamente al objeto.

- El nombre de objeto de cada componente puede tener una longitud máxima de 10 caracteres y el tipo de objeto, 6 caracteres.
- La jerarquía de directorios dentro de QSYS.LIB puede tener una extensión de dos o tres niveles (dos o tres componentes en el nombre de vía de acceso) en función del tipo de objeto al que se accede. Si el objeto es un archivo de base de datos, la jerarquía puede contener tres niveles (biblioteca, archivo, miembro); de lo contrario, sólo puede haber dos niveles (biblioteca, objeto). La combinación de la longitud de cada nombre de componente y el número de niveles de directorio determina la longitud máxima del nombre de la vía de acceso.

Si se incluyen "raíz" (/) y QSYS.LIB como los dos primeros niveles, la jerarquía de directorios del QSYS.LIB puede tener una profundidad de cinco niveles.

- Los caracteres de los nombres se convierten a CCSID 37 al almacenar los nombres. Los nombres entrecomillados, en cambio, se almacenan utilizando el CCSID del trabajo.

Para obtener más información sobre CCSID, consulte la sección dedicada a la Globalización del iSeries Information Center.

Enlaces en el sistema de archivos QSYS.LIB

Los enlaces simbólicos no se pueden crear ni almacenar en el sistema de archivos QSYS.LIB.

La relación entre una biblioteca y los objetos de una biblioteca es equivalente a un enlace fijo entre la biblioteca y cada objeto de la misma. El sistema de archivos integrado maneja la relación biblioteca-objeto

como un enlace. Así pues, es posible tener un enlace desde un sistema de archivos con soporte para enlaces simbólicos con un objeto del sistema de archivos QSYS.LIB.

Consulte el apartado “Enlace” en la página 18 para ver la descripción de los enlaces.

Utilización de los mandatos y pantallas del sistema de archivos integrado en el sistema de archivos QSYS.LIB

Los mandatos listados en el apartado “Realizar operaciones mediante mandatos CL” en la página 26 pueden realizar operaciones en el sistema de archivos QSYS.LIB, con las excepciones siguientes:

- El mandato ADDLNK sólo puede utilizarse para crear un enlace simbólico *con* un objeto de QSYS.LIB.
- Las operaciones de archivos sólo se pueden efectuar con archivos físicos descritos por programa y con archivos físicos fuente.
- Los mandatos STRJRN y ENDJRN no pueden utilizarse en los archivos físicos de bases de datos.

Las mismas restricciones son aplicables a las pantallas de usuario descritas en el apartado “Realizar operaciones utilizando los menús y pantallas del iSeries” en la página 25.

Utilización de las API del sistema de archivos integrado en el sistema de archivos QSYS.LIB

Las funciones del lenguaje C listadas en el apartado “Realizar operaciones utilizando API” en la página 49 pueden realizar operaciones en el sistema de archivos QSYS.LIB, con las siguientes excepciones:

- Las operaciones de archivos sólo se pueden efectuar con archivos físicos descritos por programa y con archivos físicos fuente.
- La función symlink() sólo puede utilizarse para enlazar con un objeto de QSYS.LIB desde otro sistema de archivos que ofrece soporte a enlaces simbólicos.
- Las API QjoStartJournal() y QjoEndJournal() no pueden utilizarse en archivos físicos de base de datos.

QSYS.LIB de ASP independiente

El sistema de archivos QSYS.LIB de ASP independiente ofrece soporte para la estructura de bibliotecas del iSeries server en cualquier agrupación de almacenamiento auxiliar (ASP) independiente que cree y defina. Este sistema de archivos proporciona acceso a los archivos de base de datos y al resto de tipos de objeto del iSeries server que gestiona el soporte de bibliotecas en las ASP independientes.

Además:

- Ofrece soporte para todas las interfaces de usuario e interfaces de programación que realizan operaciones sobre las bibliotecas del iSeries server y sobre los objetos de dichas bibliotecas en las ASP independientes.
- Soporta todos los lenguajes y recursos de programación que operan sobre archivos de base de datos.
- Ofrece un amplio soporte de administración para gestionar los objetos del iSeries server
- Soporta operaciones de E/S continuas en miembros de archivos físicos, espacios de usuario y archivos de salvar.

Para obtener más información sobre el sistema de archivos QSYS.LIB de ASP independiente, consulte Utilización de QSYS.LIB de ASP independientes mediante la interfaz del sistema de archivos integrado.

Utilización de QSYS.LIB de ASP independiente mediante la interfaz del sistema de archivos integrado

Es posible acceder al sistema de archivos QSYS.LIB de ASP independiente mediante la interfaz del sistema de archivos integrado utilizando los mandatos del servidor de archivos OS/400 o sistema de archivos integrado, las pantallas de usuario, y las API del lenguaje C. En lo que respecta al uso de dichas interfaces del sistema de archivos integrado, debe tener en cuenta las consideraciones y limitaciones siguientes.

Lista de autorizaciones QPWFSERVER en el sistema de archivos QSYS.LIB de ASP independiente

QPWFSERVER es una lista de autorizaciones (el tipo de objeto es *AUTL) que proporciona requisitos de acceso adicionales a todos los objetos que se encuentran en el sistema de archivos QSYS.LIB de ASP independiente a los que se acceda mediante clientes remotos. Las autorizaciones especificadas en la lista de autorizaciones se aplican a todos los objetos del sistema de archivos QSYS.LIB de ASP independiente.

La autorización por omisión de este objeto es la autorización PUBLIC *USE. El administrador puede utilizar los mandatos EDTAUTL (Editar lista de autorizaciones) o WRKAUTL (Trabajar con lista de autorizaciones) para modificar el valor de esta autorización. El administrador puede asignar la autorización PUBLIC *EXCLUDE a la lista de autorizaciones de forma que el público general no pueda acceder a los objetos de QSYS.LIB de ASP independiente desde clientes remotos.

Restricciones en el manejo de archivos QSYS.LIB de ASP independiente

- No se da soporte a archivos lógicos.
- Los archivos físicos soportados para el acceso en modalidad de texto son los archivos físicos descritos por programa que contienen un solo campo y los archivos físicos fuente que contienen un solo campo de texto. Los archivos físicos soportados para el acceso en modalidad binaria incluyen los archivos físicos descritos externamente además de los archivos soportados para acceso en modalidad de texto.
- No se da soporte al bloqueo de rango de bytes. (Si desea obtener más información sobre el bloqueo de rango de bytes, consulte el tema fcntl() de iSeries Information Center.)
- Si algún trabajo tiene abierto un miembro de archivo de base de datos, sólo un trabajo tiene acceso de grabación a ese miembro de archivo en todo momento. Las demás peticiones disponen de acceso de sólo lectura.

Soporte de espacios de usuario en el sistema de archivos QSYS.LIB de ASP independiente

El QSYS.LIB de ASP independiente da soporte a operaciones de E/S continuas en objetos espacio de usuario. Por ejemplo, un programa puede grabar datos continuos en un espacio de usuario y leer datos de un espacio de usuario. El tamaño máximo de un espacio de usuario es 16 776 704 bytes.

Tenga en cuenta que los espacios de usuario no tienen CCSID (identificador de juego de caracteres). Por consiguiente, el CCSID devuelto es el CCSID por omisión del trabajo.

Soporte de archivos de salvar en el sistema de archivos de QSYS.LIB de ASO independiente

EL QSYS.LIB de ASP independiente admite las operaciones de E/S continuas para guardar objetos de archivo. Por ejemplo, un archivo de salvar existente tiene datos que pueden leerse o copiarse a otro archivo hasta que sea necesario poner los datos en un tercer objeto archivo de salvar ya existente y vacío. Cuando un archivo de salvar se abre para grabación, no se permiten otras instancias de abrir del archivo. Un archivo de salvar **permite** que existan varias instancias de abrir para lectura, a condición de que ningún trabajo tenga más de una instancia de abrir del archivo para lectura. Un archivo de salvar no puede abrirse para acceso de lectura-grabación. Las operaciones de E/S continuas para guardar los datos de un archivo no están permitidas si un trabajo se ejecuta en varias hebras.

Las operaciones de E/S continuas en un archivo de salvar no están permitidas si el archivo de salvar o su directorio van a exportarse a través del servidor del sistema de archivos de red. No obstante, sí que pueden accederse desde clientes PC y a través del sistema de archivos QFileSvr.400.

Mayúsculas y minúsculas en el sistema de archivos QSYS.LIB de ASP independiente

En general, el sistema de archivos QSYS.LIB de ASP independiente no distingue entre mayúsculas y minúsculas en los nombres de objetos. Una búsqueda de nombres de objetos obtiene el mismo resultado independientemente de que los caracteres de los nombres estén en mayúsculas o en minúsculas.

Sin embargo, si un nombre está delimitado por comillas, se conservan las mayúsculas y minúsculas de los caracteres. Una búsqueda de nombres entre comillas, por lo tanto, es sensible a las mayúsculas y minúsculas de los caracteres del nombre entre comillas.

Nombres de vía de acceso en el sistema de archivos QSYS.LIB de ASP independiente

- Cada componente del nombre de vía de acceso debe contener el nombre del objeto seguido de su tipo de objeto. Por ejemplo:

```
/asp_name/QSYS.LIB/QGPL.LIB/PRT1.OUTQ
```

```
/asp_name/QSYS.LIB/EMP.LIB/PAY.FILE/TAX.MBR
```

donde `asp_name` es el nombre de la ASP independiente. El nombre del objeto y el tipo del objeto se separan con un punto (.). Los objetos de una biblioteca pueden tener el mismo nombre si tienen tipos de objeto distintos, por lo que el tipo de objeto debe especificarse para identificar exclusivamente al objeto.

- El nombre de objeto de cada componente puede tener una longitud máxima de 10 caracteres y el tipo de objeto, 6 caracteres.
- La jerarquía de directorios dentro de QSYS.LIB de ASP independiente puede tener una extensión de dos o tres niveles (dos o tres componentes en el nombre de vía de acceso) en función del tipo de objeto al que se accede. Si el objeto es un archivo de base de datos, la jerarquía puede contener tres niveles (biblioteca, archivo, miembro); de lo contrario, sólo puede haber dos niveles (biblioteca, objeto). La combinación de la longitud de cada nombre de componente y el número de niveles de directorio determina la longitud máxima del nombre de la vía de acceso.

Si /, `asp_name` y QSYS.LIB se incluyen como los tres primeros niveles, la jerarquía de directorios del sistema de archivos QSYS.LIB de ASP independiente puede tener una profundidad de seis niveles.

- Los caracteres de los nombres se convierten a CCSID 37 al almacenar los nombres. Los nombres entrecomillados, en cambio, se almacenan utilizando el CCSID del trabajo.

Para obtener más información sobre CCSID, consulte la sección dedicada a la Globalización del iSeries Information Center.

Enlaces en el sistema de archivos QSYS.LIB de ASP independiente

No pueden crearse o almacenarse enlaces simbólicos en el sistema de archivos QSYS.LIB de ASP independiente.

La relación entre una biblioteca y los objetos de una biblioteca es equivalente a un enlace fijo entre la biblioteca y cada objeto de la misma. El sistema de archivos integrado maneja la relación biblioteca-objeto como un enlace. Así pues, es posible tener un enlace desde un sistema de archivos con soporte para enlaces simbólicos con un objeto del sistema de archivos QSYS.LIB de ASP independiente.

Consulte el apartado “Enlace” en la página 18 para ver la descripción de los enlaces.

Utilización de los mandatos y pantallas del sistema de archivos integrado en el sistema de archivos QSYS.LIB de ASP independiente

Los mandatos que aparecen en “Realizar operaciones mediante mandatos CL” en la página 26 pueden realizar operaciones en el sistema de archivos QSYS.LIB de ASP independiente, con las excepciones siguientes:

- El mandato ADDLNK sólo puede utilizarse para crear un enlace simbólico *con* un objeto de QSYS.LIB de ASP independiente.
- Las operaciones de archivos sólo se pueden efectuar con archivos físicos descritos por programa y con archivos físicos fuente.
- Los mandatos STRJRN y ENDJRN no pueden utilizarse en los archivos físicos de bases de datos.

- No es posible trasladar bibliotecas en el sistema de archivos QSYS.LIB de ASP independiente a agrupaciones de almacenamiento auxiliar (ASP) básicas utilizando el mandato MOV. Sin embargo, puede trasladar bibliotecas de QSYS.LIB de ASP independiente a la ASP del sistema u otras ASP independientes.
- Si utiliza SAV o RST para guardar o restaurar objetos de biblioteca en una ASP independiente, esa ASP independiente debe asociarse con el trabajo que ejecuta SAV o RST, o es preciso especificar la ASP independiente en el parámetro ASPDEV. La convención para la asignación de nombres a las vías de acceso de /asp_name/QSYS.LIB/object.type no se admite para SAV y RST.

Las mismas restricciones son aplicables a las pantallas de usuario descritas en el apartado “Realizar operaciones utilizando los menús y pantallas del iSeries” en la página 25.

Utilización de las API del sistema de archivos integrado en el sistema de archivos QSYS.LIB de ASP independiente

Las funciones del lenguaje C listadas en el apartado “Realizar operaciones utilizando API” en la página 49 pueden realizar operaciones en el sistema de archivos QSYS.LIB de ASP independiente, excepto las siguientes:

- Las operaciones de archivos sólo se pueden efectuar con archivos físicos descritos por programa y con archivos físicos fuente.
- La función symlink() puede utilizarse sólo para enlazar con un objeto de QSYS.LIB de ASP independiente desde otro sistema de archivos que admita enlaces simbólicos.
- Las API QjoStartJournal() y QjoEndJournal() no pueden utilizarse en archivos físicos de base de datos.

Sistema de archivos de servicios de biblioteca de documentos (QDLS)

El sistema de archivos QDLS soporta la estructura de carpetas. Proporciona acceso a documentos y carpetas.

Además:

- Soporta carpetas y objetos de biblioteca de documentos (DLO) del servidor iSeries.
- Soporta datos almacenados en archivos continuos.

Para obtener más información sobre QDLS, consulte Utilización de QDLS mediante la interfaz del sistema de archivos integrado.

Utilización de QDLS mediante la interfaz del sistema de archivos integrado

Al sistema de archivos QDLS puede accederse mediante la la interfaz del sistema de archivos integrado utilizando los mandatos del servidor de archivos OS/400 o sistema de archivos integrado, las pantallas de usuario y las API del lenguaje C. En lo que respecta al uso de dichas interfaces del sistema de archivos integrado, debe tener en cuenta las consideraciones y limitaciones siguientes.

Sistema de archivos integrado y HFS en el sistema de archivos QDLS

Las operaciones se pueden efectuar sobre los objetos del sistema de archivos QDLS no sólo a través de los mandatos CL de los objetos de biblioteca de documentos (DLO), sino también a través de la la interfaz del sistema de archivos integrado o las API que proporciona un sistema de archivos jerarquizado denominado HFS. Mientras que el sistema de archivos integrado se basa en el modelo de programa del entorno de lenguajes integrados (ILE), HFS se basa en el modelo de programa del servidor iSeries original.

Las API del HFS permiten realizar operaciones adicionales que no están soportadas en el sistema de archivos integrado. En particular, las API del HFS se pueden utilizar para acceder y modificar los atributos ampliados de directorio (también conocidos como *atributos de entrada de directorio*). Tenga en cuenta que

las reglas de denominación para utilizar las API del HFS son distintas de las reglas de denominación para las API que utilizan la interfaz del sistema de archivos integrado.

Para obtener más información sobre HFS, consulte el apartado API del sistema de archivos jerarquizado del iSeries Information Center.

Incorporación de usuarios en el sistema de archivos QDLS

Es necesario haberse incorporado en el directorio de distribución del sistema para trabajar con objetos de QDLS.

Mayúsculas y minúsculas en el sistema de archivos QDLS

QDLS convierte los caracteres alfabéticos ingleses de la **a** a la **z** que están en minúsculas a mayúsculas cuando se utilizan en nombres de objetos. Por lo tanto, la búsqueda de nombres de objetos utilizando sólo esos caracteres no es sensible a las mayúsculas y minúsculas.

En todos los demás casos, la búsqueda es sensible a las mayúsculas y minúsculas en QDLS.

Para obtener más información, consulte el apartado Nombre de carpeta y documento del iSeries Information Center.

Nombres de vía de acceso en el sistema de archivos QDLS

- Cada componente del nombre de la vía de acceso puede componerse de un solo nombre, como el ejemplo siguiente:

`/QDLS/FLR1/DOC1`

o de un nombre más una extensión (semejante a la extensión de los archivos de DOS), como por ejemplo:

`/QDLS/FLR1/DOC1.TXT`

- El nombre de cada componente puede tener una longitud máxima de 8 caracteres, y la extensión (si la hay) puede tener una longitud máxima de 3 caracteres. La longitud máxima del nombre de la vía de acceso es de 82 caracteres, asumiendo un nombre de vía de acceso absoluto que empieza con /QDLS.
- La jerarquía de directorios dentro de QDLS puede tener una extensión de 32 niveles. Si se incluyen / y QDLS como los dos primeros niveles, la jerarquía de directorios puede tener una profundidad de 34 niveles.
- Los caracteres de los nombres se convierten a la página de códigos del trabajo al almacenarse los nombres, a menos que se haya creado el área de datos Q0DEC500 en la biblioteca QUSRSYS. Si existe esta área de datos, los caracteres de los nombres se convierten a la página de códigos 500 al almacenarse los nombres. Esta función aporta compatibilidad con el comportamiento del sistema de archivos QDLS en releases anteriores. Puede rechazarse un nombre si no puede convertirse a la página de códigos apropiada.

Para obtener más información sobre las páginas de código, consulte la sección dedicada a la globalización, en iSeries Information Center.

Enlaces en el sistema de archivos QDLS

Los enlaces simbólicos no se pueden crear ni almacenar en el sistema de archivos QDLS.

El sistema de archivos integrado maneja la relación entre una carpeta y los objetos de biblioteca de documentos situados en una carpeta como equivalente a un enlace entre la carpeta y cada objeto de la carpeta. Así pues, es posible enlazar con un objeto del sistema de archivos QDLS desde un sistema de archivos con soporte para enlaces simbólicos.

Consulte el apartado "Enlace" en la página 18 para ver la descripción de los enlaces.

Utilización de los mandatos y pantallas del sistema de archivos integrado en el sistema de archivos QDLS

Los mandatos listados en “Realizar operaciones mediante mandatos CL” en la página 26 pueden realizar operaciones en el sistema de archivos QDLS, con las excepciones siguientes:

- El mandato ADDLNK sólo puede utilizarse para enlazar *con* un objeto de QDLS desde otro sistema de archivos que soporte enlaces simbólicos.
- Los mandatos CHKIN y CHKOUT están soportados en los archivos, pero no en los directorios.
- No se soportan los mandatos APYJRNCHG, ENDJRN, SNDJRNE y STRJRN.

Las mismas restricciones son aplicables a las pantallas de usuario descritas en el apartado “Realizar operaciones utilizando los menús y pantallas del iSeries” en la página 25.

Utilización de las API del sistema de archivos integrado en el sistema de archivos QDLS

Las funciones del lenguaje C listadas en el apartado “Realizar operaciones utilizando API” en la página 49 pueden realizar operaciones en el sistema de archivos QDLS, con las siguientes excepciones:

- La función `symlink()` puede utilizarse solamente para enlazar con un objeto de QDLS desde otro sistema de archivos que soporte enlaces simbólicos.
- Las funciones siguientes no están soportadas:

`givedescriptor()`

`ioctl()`

`link()`

`QjoEndJournal()`

`QjoRetrieveJournalInformation()`

`QJORJIDI()`

`QJOSJRNE()`

`QjoStartJournal()`

`Qp0IGetPathFromFileID()`

`readlink()`

`takedescriptor()`

Sistema de archivos óptico (QOPT)

El sistema de archivos QOPT da acceso a los datos continuos almacenados en soportes ópticos.

Además:

- Proporciona una estructura jerárquica de directorios semejante a la de los sistemas operativos de PC como DOS y OS/2.
- Está optimizado para la entrada/salida de archivos continuos.
- Soporta datos almacenados en archivos continuos.

Para obtener más información sobre QOPT, consulte Utilización de QOPT mediante el la interfaz del sistema de archivos integrado.

Utilización de QOPT mediante la interfaz del sistema de archivos integrado

Al sistema de archivos QOPT se puede acceder a través de la la interfaz del sistema de archivos integrado utilizando el servidor de archivos OS/400 o los mandatos del sistema de archivos integrado, las pantallas de usuario y las API. En lo que respecta al uso de la interfaz del sistema de archivos integrado, deberá tener en cuenta las consideraciones y limitaciones siguientes.

Para obtener más detalles, consulte la publicación Optical Support  .

Sistema de archivos integrado y HFS en el sistema de archivos QOPT

Las operaciones se pueden efectuar sobre los objetos del sistema de archivos QOPT mediante la interfaz del sistema de archivos integrado o las API proporcionadas por el sistema de archivos jerarquizado denominado HFS. Mientras que el sistema de archivos integrado se basa en el modelo de programa del entorno de lenguajes integrados (ILE), HFS se basa en el modelo de programa del servidor iSeries original.

Las API del HFS permiten realizar operaciones adicionales que no están soportadas en el sistema de archivos integrado. En concreto, puede utilizar las API del HFS para acceder y cambiar atributos ampliados de directorio (también denominados *atributos de entrada de directorio*) o para trabajar con archivos ópticos retenidos. Tenga en cuenta que las reglas de denominación para utilizar las API del HFS son distintas de las reglas de denominación para las API que utilizan la interfaz del sistema de archivos integrado.

Para obtener más información sobre las API de HFS, consulte el apartado API del sistema de archivos jerarquizado del iSeries Information Center, o de la publicación Optical Support  .

Mayúsculas y minúsculas en el sistema de archivos QOPT

Dependiendo del formato del soporte de almacenamiento óptico, se conservarán o no las mayúsculas y minúsculas al crear archivos o directorios en QOPT. Sin embargo, las búsquedas de archivos y directorios no distinguen entre mayúsculas y minúsculas independientemente del formato del soporte de almacenamiento óptico.

Nombres de vía de acceso en el sistema de archivos QOPT

- El nombre de vía de acceso debe empezar por una barra inclinada (/). La vía de acceso consta del nombre de sistema de archivos, el nombre de volumen, los nombres de directorio y subdirectorio y el nombre de archivo. Por ejemplo:

/QOPT/NOMBREVOLUMEN/NOMBREDIRECTORIO/NOMBRESUBDIRECTORIO/NOMBREARCHIVO

- El nombre de sistema de archivos, QOPT, es obligatorio.
- La longitud del nombre de volumen y de vía de acceso varían dependiendo del formato del soporte de almacenamiento óptico.
- Puede especificarse simplemente /QOPT en el nombre de la vía de acceso o incluir uno o más directorios o subdirectorios en el nombre de la vía de acceso. Los nombres de directorios y archivos permiten cualquier carácter excepto los caracteres hexadecimales que van del X'00' a X'3F' y X'FF'. Pueden aplicarse otras restricciones dependiendo del formato del soporte de almacenamiento óptico.
- El nombre de archivo es el último elemento del nombre de vía de acceso. La longitud del nombre de archivo está limitada por la longitud del nombre de directorio en la vía de acceso.

Si quiere obtener más detalles sobre las reglas de denominación de las vías de acceso en el sistema de archivos QOPT, consulte la explicación acerca de las “reglas de denominación de las vías de acceso” en la publicación Optical Support  .

Enlaces en el sistema de archivos QOPT

El sistema de archivos QOPT sólo soporta un enlace con un objeto. Los enlaces simbólicos no se pueden crear ni almacenar en QOPT. Sin embargo, se puede acceder a los archivos de QOPT utilizando un enlace simbólico desde el sistema de archivos “raíz” (/) o QOpenSys.

Consulte el apartado “Enlace” en la página 18 para ver la descripción de los enlaces.

Utilización de los mandatos y pantallas del sistema de archivos integrado en el sistema de archivos QOPT

La mayoría de los mandatos listados en el apartado “Realizar operaciones mediante mandatos CL” en la página 26 pueden realizar operaciones en el sistema de archivos QOPT. Sin embargo, hay unas pocas excepciones. Tenga presente que puede que la ejecución de esos mandatos CL en un proceso que puede ejecutarse en varias hebras no sea segura; puede que se apliquen ciertas restricciones, dependiendo del formato del soporte de almacenamiento óptico. Las mismas restricciones son aplicables a las pantallas de usuario descritas en el apartado “Realizar operaciones utilizando los menús y pantallas del iSeries” en la página 25.

Los siguientes mandatos del sistema de archivos integrado no se admiten en el sistema de archivos QOPT:

- ADDLNK
- APYJRNCHG
- CHKIN
- CHKOUT
- ENDJRN
- SNDJRNE
- STRJRN
- WRKOBJOWN
- WRKOBJPGP

Utilización de las API del sistema de archivos integrado en el sistema de archivos QOPT

Todas las API del lenguaje C que se listan en el apartado “Realizar operaciones utilizando API” en la página 49 pueden operar en el sistema de archivos “raíz” (/) de un modo seguro en ejecución multihebra, con las siguientes excepciones:

- QjoEndJournal()
- QjoRetrieveJournalInformation()
- QJORJIDI()
- QJOSJRNE()
- QjoStartJournal()

Sistema de archivos NetWare (QNetWare)

Los sistemas de archivos QNetWare proporcionan acceso a los datos ubicados en un Integrated xSeries Server for iSeries local o remoto en el que se ejecuta Novell NetWare 4.10 o 4.11, o en Servidores PC autónomos en los que se ejecuta Novell NetWare 3.12, 4.10, 4.11 o 5.0.

Además:

- Proporciona acceso a los objetos de los Servicios de Directorio NetWare (NDS).
- Soporta datos almacenados en archivos continuos.
- Proporciona montaje dinámico de sistemas de archivos NetWare en el espacio de nombres local.

Nota: el sistema de archivos QNetWare está disponible sólo si en el sistema está instalada la integración mejorada NetWare para iSeries 400, opción 25 del BOSS. Después de realizar la siguiente IPL una vez efectuada la instalación, el directorio /QNetWare y sus subdirectorios aparecen como parte de la estructura de directorios del sistema de archivos integrado.

Consulte los siguientes temas para obtener más información sobre el sistema de archivos QNetWare:

- Montar sistemas de archivos NetWare

- Estructura de directorios QNetWare
- Utilización de QNetWare mediante la la interfaz del sistema de archivos integrado

Montar sistemas de archivos NetWare

Los sistemas de archivos NetWare ubicados en servidores Novell NetWare pueden montarse en el sistema de archivos “raíz” (/), en QOpenSys y en otros sistemas de archivos con el fin de simplificar el acceso y de que el rendimiento sea mejor que con el directorio /QNetWare. Montar los sistemas de archivos NetWare puede servir también para sacar partido de las opciones del mandato Añadir sistema de archivo montado (ADDMFS) como, por ejemplo, montar un sistema de archivos de lectura-grabación como de sólo lectura.

Los sistemas de archivos NetWare pueden montarse utilizando una vía de acceso NDS o especificando una vía de acceso NetWare en la forma SERVIDOR/VOLUMEN:directorio/directorio. Por ejemplo, para montar el directorio portal ubicado en el volumen Nido del servidor Ruiseñor, se utilizaría la sintaxis siguiente:

```
RUISEÑOR/NIDO:portal
```

Esta sintaxis de vía de acceso es semejante a la sintaxis del mandato MAP de NetWare. Las vías de acceso de NDS pueden utilizarse para especificar una vía de acceso a un volumen de NetWare, pero, en sí, no pueden montarse

Estructura de directorios de QNetWare

La estructura de directorios de /QNetWare representa varios sistemas de archivos distintos:

- La estructura representa los servidores Novell NetWare y los volúmenes en la red de la forma siguiente:

```
/QNetWare/SERVIDOR.SVR/VOLUMEN
```

La extensión .SVR se utiliza para representar un servidor Novell NetWare.

- Cuando se accede a un volumen de un servidor por medio de los menús, los mandatos o las API del sistema de archivos integrado, el directorio raíz del volumen NetWare queda montado de forma automática en el directorio VOLUMEN de /QNetWare.
- QNetWare representa los árboles de NDS de la red de la manera siguiente:

```
/QNetWare/ARB_EMPR.TRE/ESP.C/ORG.O/UNID_ORG.OU/VOL_SVR1.CN
```

Las extensiones .TRE, .C, .O, .OU y .CN se utilizan para representar árboles de NDS, países, organizaciones, unidades organizativas y nombres comunes, respectivamente. Si a un volumen de Novell NetWare se accede a través de la vía de acceso de NDS por medio de un objeto volumen o del alias de un objeto volumen, el directorio raíz del mismo se monta también de forma automática en el objeto de NDS.

Utilización de QNetWare mediante la la interfaz del sistema de archivos integrado

Al sistema de archivos QNetWare puede accederse mediante la la interfaz del sistema de archivos integrado utilizando el servidor de archivos OS/400 o los mandatos, las pantallas de usuario y las API del sistema de archivos integrado. Debe tener presentes las consideraciones, limitaciones y dependencias siguientes.

Autorizaciones y propiedad en el sistema de archivos QNetWare

Los archivos y directorios de QNetWare los almacenan y gestionan servidores Novell NetWare. Al utilizar mandatos y API para recuperar o establecer las autorizaciones de los propietarios o usuarios, QNetWare correlaciona los usuarios de NetWare con los usuarios del servidor iSeries tomando como base el nombre de los usuarios. Si el nombre de NetWare tiene una longitud de más de diez caracteres o no existe ningún usuario del servidor iSeries correspondiente, no se correlaciona la autorización. Los propietarios

que no pueden correlacionarse se correlacionan de forma automática con el perfil de usuario QDFTOWN. Las autorizaciones de los usuarios pueden visualizarse y cambiarse con los mandatos WRKAUT y CHGAUT. Al transferirlas al servidor, o desde el servidor, se correlacionan con autorizaciones del servidor iSeries.

Auditorías en el sistema de archivos QNetWare

Aunque Novell NetWare soporta la auditoría de archivos y directorios, el sistema de archivos QNetWare no puede cambiar los valores de auditoría de dichos objetos. Por lo tanto, el mandato CHGAUD no está soportado.

Archivos y directorios en el sistema de archivos QNetWare

El sistema de archivos QNetWare no contempla el caso en que los archivos y directorios se entran en un mandato o API. Todos los nombres se pasan a mayúsculas al transmitirlos al servidor NetWare. Novell NetWare también soporta los espacios de nombres de múltiples plataformas, como por ejemplo DOS, OS/2, Apple Macintosh y NFS. El sistema de archivos QNetWare sólo da soporte al espacio de nombres de DOS. Puesto que el espacio de nombres DOS es obligatorio en todos los volúmenes de Novell NetWare, todos los archivos y directorios aparecerán en el sistema de archivos QNetWare.

Objetos NDS en el sistema de archivos QNetWare

El sistema de archivos QNetWare da soporte a la visualización de nombres de NDS en mayúsculas y en minúsculas.

Enlaces en el sistema de archivos QNetWare

El sistema de archivos QNetWare da soporte sólo a un enlace con un objeto. En QNetWare no pueden crearse ni almacenarse enlaces simbólicos. Sin embargo, pueden crearse en los directorios "raíz" (/) o de QOpenSys que señalen a un archivo o directorio de QNetWare.

Utilización de los mandatos y las pantallas del sistema de archivos integrado en el sistema de archivos QNetWare

Los mandatos listados en el apartado "Realizar operaciones mediante mandatos CL" en la página 26 pueden realizar operaciones en el sistema de archivos QNetWare, con las siguientes excepciones:

- ADDLINK
- APYJRNCHG
- CHGAUD
- CHGPGP
- CHKIN
- CHKOUT
- ENDJRN
- SNDJRN
- STRJRN
- WRKOBJOWN
- WRKOBJPGP

Además de los mandatos anteriores, los siguientes no pueden utilizarse en volúmenes, servidores ni objetos de NDS:

- CHGOWN
- CPYFRMSTMF
- CPYTOSTMF
- CRTDIR

Utilización de las API del sistema de archivos integrado en el sistema de archivos QNetWare

Las funciones del lenguaje C listadas en el apartado "Realizar operaciones utilizando API" en la página 49 pueden realizar operaciones en el sistema de archivos, QNetWare, excepto las API siguientes:

- givedescriptor()
- link()
- | QjoEndJournal()
- | QjoRetrieveJournalInformation()
- | QJORJIDI()
- | QJOSJRNE()
- | QjoStartJournal()
- readlink()
- symlink()
- takedescriptor()

Además de las API anteriores, no pueden utilizarse las siguientes API sobre objetos, servidores o volúmenes NDS:

- chmod()
- chown()
- create()
- fchmod()
- fchown()
- fcntl()
- ftruncate()
- lseek()
- mkdir()
- read()
- readv()
- unmask()
- write()
- writenv()

Sistema de archivos de Windows NT Server (QNTC)

El sistema de archivos QNTC proporciona acceso a los datos y objetos que se encuentran almacenados en un Integrated xSeries Server for iSeries local o remoto en el que se ejecuta Windows NT 4.0 Server o superior, o en un servidor autónomo. Posibilita que las aplicaciones del servidor iSeries utilicen los mismos datos que los clientes de Windows NT. Almacena datos en archivos continuos.

El sistema de archivos QNTC forma parte del sistema operativo OS/400 base. Para utilizar el sistema de archivos QNTC, los Programas de Utilidad de Conectividad TCP/IP para iSeries 400 han de estar instalados (número de pieza 5769-TC1). Para acceder a /QNTC no es necesario que esté instalada la Integración de iSeries 400 con Windows NT Server, opción 29 del sistema operativo.

Para obtener más información sobre QNTC, consulte Utilización de QNTC mediante la la interfaz del sistema de archivos integrado.

Utilización de QNTC mediante la la interfaz del sistema de archivos integrado

Al utilizar los mandatos del servidor de archivos OS/400 o los mandatos del sistema de archivos integrado, las pantallas de usuario y las API, se puede acceder al sistema de archivos QNTC mediante la la interfaz del sistema de archivos integrado. Debe tener en cuenta las consideraciones y limitaciones siguientes.

Autorizaciones y propiedad en el sistema de archivos QNTC

El sistema de archivos QNTC no soporta el concepto de propiedad de un archivo o de un directorio. Los intentos de utilizar un mandato o una API para cambiar la propiedad de los archivos almacenados en QNTC serán infructuosos. Un perfil de usuario de sistema, que recibe el nombre de QDFTOWN, tiene en propiedad todos los archivos y directorios de QNTC.

La autorización sobre los archivos y directorios del servidor NT se administra desde el servidor Windows NT. QNTC no soporta los mandatos WRKAUT y CHGAUT.

Mayúsculas y minúsculas en el sistema de archivos QNTC

El sistema de archivos QNTC conserva el mismo formato de mayúsculas y minúsculas en el que se han entrado los nombres de objetos, pero no distingue mayúsculas de minúsculas en los nombres. Una búsqueda de nombres de objetos obtiene el mismo resultado independientemente de que los caracteres de los nombres estén en mayúsculas o en minúsculas.

Nombres de vía de acceso en el sistema de archivos QNTC

- El nombre de vía de acceso debe empezar por una barra inclinada y puede contener 255 caracteres como máximo.
- Los nombres de vía de acceso son sensibles a las mayúsculas y minúsculas.
- La vía de acceso está formada por el nombre de sistema de archivos, el nombre del servidor Windows NT, el nombre compartido, los nombre del directorio y los subdirectorios y el nombre del objeto. Los nombres de vía de acceso tienen el formato siguiente:

```
/QNTC/NombreServidor/NombreCompartido/Directorio/ . . . /Objeto  
(QNTC es una parte necesaria del nombre de vía de acceso.)
```

- El nombre de servidor puede tener una longitud de 15 caracteres como máximo. Debe formar parte de la vía de acceso.
- El nombre compartido puede tener una longitud de 12 caracteres como máximo.
- Cada componente del nombre de vía de acceso tras el nombre compartido puede tener una longitud máxima de 255 caracteres.
- Dentro de QNTC, suele haber disponibles 130 niveles de jerarquía. Si todos los componentes del nombre de la vía de acceso se incluyen como niveles de jerarquía, la jerarquía de directorios puede llegar a tener una extensión de 132 niveles.
- Los nombres se almacenan en el CCSID de Unicode.
- Cada servidor Windows NT en funcionamiento en la subred local aparecerá automáticamente como un directorio bajo /QNTC. Utilice el mandato Crear directorio (MKDIR) (consulte Tabla 2 en la página 27) o la API mkdir() (consulte "Realizar operaciones utilizando API" en la página 49) para añadir servidores Windows NT externos a la subred local.

Enlaces en el sistema de archivos QNTC

El sistema de archivos QNTC da soporte sólo a un enlace con un objeto. No puede crear o almacenar enlaces simbólicos en QNTC. Puede utilizar un enlace simbólico desde los sistemas de archivos "raíz" (/) o QOpenSys para acceder a los datos de QNTC.

Consulte el apartado "Enlace" en la página 18 para ver la descripción de los enlaces.

Utilización de los mandatos y pantallas del sistema de archivos integrado en el sistema de archivos QNTC

Los mandatos listados en el apartado "Realizar operaciones mediante mandatos CL" en la página 26 pueden realizar operaciones en el sistema de archivos QNTC, con las siguientes excepciones:

```
ADDLNK  
APYJRNCHG  
CHGOWN  
CHGAUT
```

CHGPGP
CHKIN
CHKOUT
DSPAUT
| ENDJRN
RST
SAV
| SNDJRNE
| STRJRN
WRKAUT
WRKOBJOWN
WRKOBJPGP

Las mismas restricciones son aplicables a las pantallas de usuario descritas en el apartado “Realizar operaciones utilizando los menús y pantallas del iSeries” en la página 25.

Utilización del mandato MKDIR en el sistema de archivos QNTC

Utilice el mandato Crear directorio (MKDIR) para añadir un directorio de servidor al directorio /QNTC.

Todos los servidores Windows NT en funcionamiento de la subred local se crean automáticamente. Estos servidores Windows NT externos a la subred local deben añadirse utilizando el mandato MKDIR o la API mkdir(). Por ejemplo:

```
MKDIR '/QNTC/NTSRV1'
```

añadirá el servidor NTSRV1 en la estructura de directorios del sistema de archivos QNTC para permitir el acceso a los archivos y directorios de este servidor.

También puede añadir un nuevo servidor a la estructura de directorios utilizando la dirección TCP/IP. Por ejemplo:

```
MKDIR '/QNTC/9.130.67.24'
```

añadirá el servidor en la estructura de directorios del sistema de archivos QNTC.

Nota: si utiliza la API mkdir() o el mandato CL MKDIR para añadir directorios a la estructura de directorios, éstos no continuarán siendo visibles después de una IPL. El mandato MKDIR o la API mkdir() deben volverse a emitir después de cada IPL del sistema.

Utilización de las API del sistema de archivos integrado en el sistema de archivos QNTC

Las funciones del lenguaje C que se enumeran en el apartado “Realizar operaciones utilizando API” en la página 49 pueden operar en el sistema de archivos QNTC, con las siguientes excepciones:

- Las funciones chmod(), fchmod(), utime() y umask() no tendrán ningún efecto sobre los objetos de QNTC, pero si se intentan utilizar no se producirá ningún error.
- El sistema de archivos QNTC no da soporte a las funciones siguientes.

chown()

fchown()

givedescriptor()

link()

| QjoEndJournal()

| QjoRetrieveJournalInformation()

| QJORJIDI()

| QJOSJRNE()

QjoStartJournal()
QpOIGetPathFromFileID()
readlink()
symlink()
takedescriptor()

Sistema de archivos del servidor de archivos OS/400 (QFileSvr.400)

El sistema de archivos del servidor de archivos OS/400 proporciona acceso transparente a otros sistemas de archivos que residen en servidores iSeries remotos. Se accede a el mismo mediante una estructura de directorios jerárquica.

El sistema de archivos QFileSvr.400 debe considerarse como un cliente que actúa en nombre de los usuarios para ejecutar peticiones de archivo. QFileSvr.400 interactúa con el servidor de archivos OS/400 del sistema de destino para efectuar la operación de archivo real.

Para obtener más información sobre QFileSvr.400, consulte Utilización de QFileSvr.400 mediante el la interfaz del sistema de archivos integrado.

Utilización de QFileSvr.400 mediante la interfaz del sistema de archivos integrado

Al sistema de archivos QFileSvr.400 se puede acceder a través de la la interfaz del sistema de archivos integrado utilizando el servidor de archivos OS/400 o los mandatos del sistema de archivos integrado, las pantallas de usuario y las API. En lo que respecta al uso de dichas interfaces del sistema de archivos integrado, debe tener en cuenta las consideraciones y limitaciones siguientes.

Nota: las características del sistema de archivos QFileSvr.400 vienen determinadas por las características del sistema de archivos al que se accede en el servidor de destino.

Mayúsculas y minúsculas en el sistema de archivos del servidor de archivos OS/400

Para un directorio de primer nivel, que en realidad representa el directorio "raíz" (/) del sistema de destino, el sistema de archivos QFileSvr.400 conserva el mismo formato de mayúsculas y minúsculas en el que se han entrado los nombres de objeto. Sin embargo, no se efectúa ninguna distinción entre mayúsculas y minúsculas cuando QFileSvr.400 busca nombres.

Para todos los demás directorios, el hecho de distinguir mayúsculas y minúsculas depende del sistema de archivos específico al que se acceda. QFileSvr.400 conserva el mismo formato de mayúsculas y minúsculas en el que se entran los nombres de objeto cuando las peticiones de archivo se envían al servidor de archivos OS/400.

Nombres de vía de acceso en el sistema de archivos del servidor de archivos OS/400

- Los nombres de vía de acceso tienen el formato siguiente:

```
/QFileSvr.400/NombreUbicaciónRemota/Directorio/Directorio . . . /Objeto
```

El directorio de primer nivel (es decir, NombreUbicaciónRemota en el ejemplo anterior) representa lo siguiente:

- El nombre del servidor de destino que se utilizará para establecer una conexión de comunicaciones. El nombre de servidor de destino puede ser uno de los siguientes:
 - Un nombre de sistema principal TCP/IP (por ejemplo, beowulf.newyork.corp.com)
 - Un nombre de LU 6.2 SNA (por ejemplo, appn.newyork).
- El directorio "raíz" (/) del servidor de destino

Por consiguiente, cuando se crea un directorio de primer nivel utilizando una interfaz de sistema de archivos integrado, los atributos especificados se pasan por alto.

Nota: los directorios de primer nivel no perduran tras una IPL. Es decir, los directorios de primer nivel deben crearse de nuevo después de cada IPL.

- Cada componente del nombre de vía de acceso puede tener una longitud máxima de 255 caracteres. El nombre completo de la vía de acceso puede tener una longitud máxima de 16 megabytes.

Nota: el sistema de archivos en el que reside el objeto puede restringir la longitud de componente y la longitud del nombre de vía de acceso a menos que el máximo permitido por QFileSvr.400.

- No existe límite para la extensión de la jerarquía de directorios, a excepción de los límites del programa y del sistema, así como los límites impuestos por el sistema de archivos al que se accede.
- Los caracteres de los nombres se convierten al formato UCS2 Nivel 1 al almacenarse los nombres (consulte el apartado “Continuidad de nombres” en la página 23).

Comunicaciones en el sistema de archivos del servidor de archivos OS/400

- Las conexiones TCP con el servidor de archivos de un servidor de destino pueden establecerse sólo si el subsistema QSERVER del servidor de destino está activo.
- Las conexiones LU 6.2 SNA sólo se intentan si hay una sesión controlada localmente que no esté en uso (por ejemplo, una sesión establecida específicamente para que la utilice la conexión LU 6.2). Al establecer conexiones LU 6.2, el sistema de archivos QFileSvr.400 utiliza la modalidad BLANK. En el sistema de destino, el trabajo QPWFSERV se somete en el subsistema QSERVER. El perfil de usuario de este trabajo se define mediante la entrada de comunicaciones para la modalidad BLANK. Para obtener más información sobre las comunicaciones LU 6.2, consulte la publicación APPC Programming



- Las peticiones de servidor de archivos que utilizan TCP como protocolo de comunicaciones se ejecutan dentro del contexto del trabajo que emite la petición. Las peticiones de servidor de archivos que utilizan SNA como protocolo de comunicaciones se ejecutan en el trabajo del sistema OS/400 Q400FILSVR.
- Si todavía no se ha establecido conexión con el servidor de destino, el sistema de archivos QFileSvr.400 asume que el directorio de primer nivel representa un nombre de sistema principal TCP/IP. El sistema de archivos QFileSvr.400 sigue los siguientes pasos para establecer una conexión con el servidor de destino:
 1. Resolver el nombre de ubicación remota en una dirección IP.
 2. Conectar con el correlacionador de servidores del servidor de sistema principal en el puerto conocido públicamente 449 utilizando la dirección IP resuelta. A continuación, enviar una consulta al correlacionador de servidores para el nombre de servicio “as-file.” Como resultado de la consulta se produce una de las siguientes situaciones:
 - Si “as-file” está en la tabla de servicio del servidor de destino, el correlacionador de servidores devuelve el puerto en el que el daemon del servidor de archivos OS/400 está a la escucha.
 - Si el correlacionador de servidores no está activo en el servidor de destino, para “as-file” se utiliza el número de puerto por omisión (8473).

A continuación, el sistema de archivos QFileSvr.400 intenta establecer una conexión TCP con el daemon del servidor de archivos OS/400 en el servidor de destino. Cuando se establece la conexión, QFileSvr.400 intercambia peticiones y responde con el servidor de archivos. Dentro del subsistema QSERVER, las peticiones de prearranque QPWFSERVSO toman el control de la conexión. Cada trabajo de prearranque se ejecuta en su propio perfil de usuario.

3. Si el nombre de ubicación remota no se resuelve en una dirección IP, se presupone que el directorio de primer nivel es un nombre de LU 6.2 SNA. Por consiguiente, se intenta establecer una conexión APPC con el servidor de archivos OS/400
- El sistema de archivos QFileSvr.400 efectúa una comprobación periódicamente (cada 2 horas) para determinar si hay conexiones que no se utilicen (por ejemplo, que no haya ningún archivo abierto)

asociado con la conexión) y que dichas conexiones no han tenido actividad durante un periodo de 2 horas. Si se encuentra una conexión que cumpla estas características, la conexión se finaliza.

- El sistema de archivos QFileSvr.400 no puede detectar bucles. El siguiente nombre de vía de acceso es un ejemplo de un bucle:

```
/QFileSvr.400/Remoto2/QFileSvr.400/Remoto1/QFileSvr.400/Remoto2/...
```

donde Remoto1 es el sistema local. Cuando se especifica un nombre de vía de acceso que contiene un bucle, el sistema de archivos QFileSvr.400 devuelve un error tras un breve periodo de tiempo. El error indica que se ha producido un tiempo de espera excedido.

El sistema de archivos QFileSvr.400 utilizará una sesión libre existente al comunicarse a través de SNA. Es necesario arrancar el módulo y establecer una sesión para QFileSvr.400 para conectarse satisfactoriamente al sistema de comunicaciones remoto.

Seguridad y autorizaciones sobre objetos en el sistema de archivos del servidor de archivos OS/400

Si ambos sistemas tienen Kerberos configurados, y el usuario se ha autenticado en Kerberos, puede utilizarse Kerberos para autenticar un sistema de archivos que resida en un servidor iSeries de destino. Si la autenticación de Kerberos no es satisfactoria, pueden utilizarse el ID de usuario y la contraseña para verificar el acceso.

Nota: Si el ticket de otorgación de tickets o el ticket del servidor caduca cuando el servidor de destino ya ha verificado el acceso, la caducidad no entrará en vigor hasta que haya finalizado la conexión con el servidor de destino. Para obtener más información sobre Kerberos, consulte el apartado Servicio de autenticación de red en el Information Center del iSeries.

- Para acceder a un sistema de archivos que resida en un iSeries server de destino, el usuario debe tener un ID y una contraseña de usuario en el servidor local si no se utiliza Kerberos para la autenticación.

Nota: si se cambia la contraseña del servidor local o de destino después de haber verificado el acceso al servidor de destino, el cambio no se reflejará hasta que finalice la conexión con el servidor de destino. Sin embargo, no hay retardo si se suprime el perfil de usuario del servidor local y se crea otro perfil de usuario con el mismo ID de usuario. En este caso, el sistema de archivos QFileSvr.400 verifica que se tenga acceso al servidor de destino.

- La autorización sobre objetos se basa en el perfil de usuario que reside en el servidor de destino. Es decir, sólo puede accederse a un objeto del sistema de archivos en el servidor de destino si el perfil de usuario del servidor de destino tiene la autorización adecuada sobre el objeto.

Enlaces en el sistema de archivos del servidor de archivos OS/400

El sistema de archivos QFileSvr.400 soporta sólo un enlace con un objeto. Los enlaces simbólicos no se pueden crear ni almacenar en QFileSvr.400. No obstante, puede accederse a los archivos de QFileSvr.400 utilizando un enlace simbólico desde los sistemas de archivos "raíz" (/), QOpenSys o definidos por el usuario.

Consulte el apartado "Enlace" en la página 18 para ver la descripción de los enlaces.

Utilización de los mandatos y pantallas del sistema de archivos integrado en el sistema de archivos servidor de archivos OS/400

Los mandatos listados en "Realizar operaciones mediante mandatos CL" en la página 26 pueden realizar operaciones en el sistema de archivos QFileSvr.400, con las excepciones siguientes:

```
ADDLNK
APYJRNCHG
CHGAUT
CHGOWN
DSPAUT
```

| ENDJRN
RST
SAV
| SNDJRNE
| STRJRN
WRKOBJOWN
WRKOBJPGP

Las mismas restricciones son aplicables a las pantallas de usuario descritas en el apartado “Realizar operaciones utilizando los menús y pantallas del iSeries” en la página 25.

Utilización de las API del sistema de archivos integrado en el sistema de archivos servidor de archivos OS/400

Las funciones del lenguaje C listadas en el apartado “Realizar operaciones utilizando API” en la página 49 pueden realizar operaciones en el sistema de archivos QFileSvr.400, con las siguientes excepciones:

chown()
fchown()
givedescriptor()
link()
| QjoEndJournal()
| QjoRetrieveJournalInformation()
| QJORJIDI()
| QJOSJRNE
| QjoStartJournal
Qp0IGetPathFromFileID()
symlink()
takedescriptor()

Sistema de archivos de red (NFS)

El sistema de archivos NFS proporciona al usuario acceso a los datos y objetos almacenados en un servidor NFS remoto. Un servidor NFS puede exportar un sistema de archivos de red que los clientes NFS montarán a continuación dinámicamente.

Además, todo sistema de archivos montado de forma local por medio del sistema de archivos de red tendrá las funciones, características, limitaciones y dependencias del directorio o sistema de archivos que se ha montado en el servidor remoto. Las operaciones no se realizan de forma local en los sistemas de archivos montados. El flujo de peticiones a través de la conexión con el servidor debe obedecer a los requisitos y restricciones del tipo de sistema de archivos en el servidor.

Para obtener más información sobre NFS, consulte Utilización de los sistemas de archivos NFS mediante la la interfaz del sistema de archivos integrado.

Utilización de los sistemas de archivos NFS mediante la la interfaz del sistema de archivos integrado

El sistema de archivos de red es accesible a través de la la interfaz del sistema de archivos integrado y tiene las siguientes consideraciones y limitaciones.

Características del sistema de archivos de red

Las características de cualquier sistema de archivos montado por medio de NFS dependen del tipo de sistema de archivos a partir del que se haya montado en el servidor. Es importante tener presente que las peticiones realizadas en lo que parece ser un directorio o un sistema de archivos local actúan, en realidad, en el servidor a través de la conexión NFS.

Esta relación cliente/servidor puede resultar confusa. Supongamos, por ejemplo, que se monta el sistema de archivos QDLS del servidor encima de una rama del directorio “raíz” (/) del cliente. Aunque el sistema de archivos montado parece ser una ampliación del directorio local, en realidad funcionará y actuará como el sistema de archivos QDLS.

Tener presente esta relación de los sistemas de archivos montados a través de NFS es importante para procesar las peticiones de forma local y mediante la conexión del servidor. Que un mandato se procese correctamente a nivel local no significa que vaya a funcionar en el directorio montado desde el servidor. Cada directorio montado en el cliente tendrá las propiedades y las características del sistema de archivos del servidor.

Combinaciones de servidores y clientes en el sistema de archivos de red

Existen tres posibilidades principales de conexiones cliente/servidor que pueden afectar a la forma en la que funcionará el sistema de archivos de red y sus características.

1. El usuario monta un sistema de archivos de un servidor iSeries en un cliente.
2. El usuario monta un sistema de archivos de un servidor UNIX en un cliente.
3. El usuario monta un sistema de archivos de un servidor que no es iSeries ni UNIX en un cliente.

En el primer caso, el sistema de archivos montado se comportará en el cliente exactamente igual que en el iSeries server. Sin embargo, deben tenerse en cuenta las dos características del Sistema de archivos de red y el sistema de archivos que se sirve. Por ejemplo, si un usuario monta el sistema de archivos QDLS del servidor en el cliente, tendrá las características y limitaciones del sistema de archivos QDLS. Por ejemplo, en el sistema de archivos QDLS, los componentes del nombre de la vía de acceso están limitados a 8 caracteres más una ampliación de 3 caracteres. Sin embargo, el sistema de archivos montado también tendrá las características y limitaciones de NFS. Por ejemplo, no podrá utilizar el mandato CHGAUD para modificar el valor de auditoría de un objeto NFS.

En el segundo caso, es importante darse cuenta de que cualquier sistema de archivos montado desde un servidor UNIX se comportará de modo similar al sistema de archivos QOpenSys del servidor iSeries. Si desea saber más acerca del sistema de archivos QOpenSys, consulte el apartado “Sistema de archivos de sistemas abiertos (QOpenSys)” en la página 67.

En el tercer caso, será necesario leer la documentación del sistema de archivos asociado con el sistema operativo del servidor.

Enlaces en el sistema de archivos de red

En general, en el sistema de archivos de red se permiten varios enlaces fijos con el mismo objeto. Los enlaces simbólicos se soportan totalmente. Se puede utilizar un enlace simbólico para tener un enlace desde el sistema de archivos de red con un objeto de otro sistema de archivos. La posibilidad de tener varios enlaces fijos y simbólicos depende por completo del sistema de archivos que se esté montando con NFS.

Consulte el apartado “Enlace” en la página 18 para ver la descripción de los enlaces.

Utilización de los mandatos del sistema de archivos integrado en el sistema de archivos de red

Todos los mandatos listados en el apartado “Realizar operaciones mediante mandatos CL” en la página 26 y las pantallas descritas en el apartado “Realizar operaciones utilizando los menús y pantallas del iSeries” en la página 25 pueden realizar operaciones en el sistema de archivos de red, con las siguientes excepciones:

- | • APYJRNCHG
- | • CHGAUD
- | • CHGATR
- | • CHGAUT
- | • CHGOWN
- | • CHGPGP
- | • CHKIN
- | • CHKOUT
- | • ENDJRN
- | • SNDJRNE
- | • STRJRN

Hay algunos mandatos CL que son específicos del sistema de archivos de red y otros sistemas de archivos montados en general. Sin embargo, puede que la ejecución de estos mandatos en un proceso que admite la ejecución multihebra no sea segura. Están descritos en la tabla siguiente. Encontrará la descripción completa de los mandatos y pantallas relacionados específicamente con el sistema de

archivos de red, en OS/400 Network File System Support .

Tabla 7. Mandatos CL del sistema de archivos de red

Mandato	Descripción
ADDMFS	Añadir sistema de archivos montado. Coloca los sistemas de archivos de servidor remoto exportados en directorios de cliente locales.
CHGNFSEXP	Cambiar exportación de sistema de archivos de red. Añade o elimina árboles de directorios de la tabla de exportación de los sistemas de archivos exportados a los clientes del sistema de archivos de red.
DSPMFSINF	Visualizar información de sistema de archivos montado. Visualiza información acerca de un sistema de archivos montado.
ENDNFSSVR	Finalizar servidor de sistema de archivos de red. Finaliza uno o todos los daemons de sistema de archivos de red del servidor.
EXPORTFS	Exportar sistema de archivos. Añade o elimina árboles de directorios de la tabla de exportación de los sistemas de archivos exportados a los clientes del sistema de archivos de red.
MOUNT	Montar sistema de archivos. Coloca los sistemas de archivos de servidor remoto exportados en directorios de cliente locales. Es un alias del mandato ADDMFS.
RLSIFSLCK	Liberar bloqueos del sistema de archivos integrado. Libera todos los bloqueos de rango de bytes del sistema de archivos de red mantenidos por un cliente o sobre un objeto.
RMVMFS	Eliminar sistema de archivos montado. Elimina los sistemas de archivos de servidor remoto exportados del espacio de nombres del cliente local.
STRNFSSVR	Iniciar servidor de sistema de archivos de red. Inicia uno o todos los daemons de sistema de archivos de red del servidor.
UNMOUNT	Desmontar sistema de archivos. Elimina los sistemas de archivos de servidor remoto exportados del espacio de nombres del cliente local. Es un alias del mandato RMVMFS.

Nota: para poder utilizar los mandatos en un sistema de archivos de red, primero debe montarse éste.

Utilización de las API del sistema de archivos integrado en el sistema de archivos de red

Todas las funciones del lenguaje C que se listan en el apartado “Realizar operaciones utilizando API” en la página 49 pueden realizar operaciones en el sistema de archivos de red, con las siguientes excepciones:

- QjoEndJournal()
- QjoRetrieveJournalInformation()
- QJORJIDI()
- QJOSJRNE()
- QjoStartJournal()

Hallará la descripción completa de las funciones del lenguaje C relacionadas específicamente con el sistema de archivos de red en OS/400 Network File System Support .

Nota: para poder utilizar las API en un sistema de archivos de red, primero debe montarse éste.

Capítulo 7. Soporte del registro por diario para los objetos del sistema de archivos integrado

La función primaria del registro por diario es permitir la recuperación de los cambios realizados en un objeto desde que se guardó por última vez.

En este capítulo se le ofrecerá una breve introducción a la gestión por diario, además de las consideraciones que hay que tener en cuenta para registrar por diario los objetos del sistema de archivos integrado y una descripción del soporte del registro por diario para los objetos del sistema de archivos integrado.

En los siguientes temas se presenta el soporte del registro por diario para los objetos del sistema de archivos integrado:

- “gestión por diario”
- “Objetos que debería registrar por diario”
- “Objetos del sistema de archivos integrado registrados por diario” en la página 96
- “Operaciones registradas en diario” en la página 97
- “Consideraciones especiales sobre las entradas de diario” en la página 98

| Para obtener más información sobre el registro por diario de objetos del sistema de archivos integrado,
| consulte el apartado gestión por diarios del iSeries Information Center.

gestión por diario

La principal finalidad de la gestión por diario es permitir la recuperación de los cambios realizados en un objeto desde que se guardó por última vez. La gestión por diario también puede utilizarse para:

- Un seguimiento de auditoría de la actividad que se da en los objetos del sistema
- Registrar la actividad que se da en los objetos distintos de los que se puede registrar por diario
- Realizar una recuperación más rápida cuando se restaura a partir de un soporte del tipo salvar mientras está activo
- Ayudar en la comprobación de programas de aplicación

Un diario puede utilizarse para definir los objetos que se quieren proteger con la gestión por diario. En el apartado “Objetos que debería registrar por diario” se dan más consideraciones sobre el registro por diario de objetos. En el sistema de archivos integrado se pueden registrar por diario archivos continuos, directorios y enlaces simbólicos. Sólo se admiten los objetos de los sistemas de archivos “raíz” (/), QOpenSys y UDFS.

Objetos que debería registrar por diario

Cuando vaya a decidir si debe registrar por diario un objeto del sistema de archivos integrado, hágase las preguntas siguientes:

- ¿Con qué frecuencia cambia el objeto? Un objeto con un gran volumen de cambio entre operaciones de salvar es un buen candidato a ser registrado por diario.
- ¿Cuál es la dificultad de reconstruir los cambios hechos a un objeto? ¿Se realizan muchos cambios en el objeto sin grabar registros? Por ejemplo, un objeto utilizado para anotar entradas de pedidos telefónicos será más difícil de reconstruir que un objeto utilizado para anotar los pedidos enviados por correo en formularios de pedidos.
- ¿Hasta qué punto es crítica la información del objeto? Si el objeto tuviera que ser restaurado al estado de la última operación de salvar, ¿qué efecto tendría en la empresa el retraso en la reconstrucción de los cambios?

- ¿Cómo se relaciona el objeto con los otros objetos del servidor? Aunque a lo mejor los datos de un objeto dado no cambien con mucha frecuencia, los datos de ese objeto puede que sean críticos para otros objetos más dinámicos del servidor. Suponga, por ejemplo, que muchos objetos dependan de un archivo maestro de clientes. Si hay que reconstruir los pedidos, el archivo maestro de clientes deberá incluir los clientes nuevos que se hayan añadido desde la operación de salvar anterior, además de los cambios realizados en los límites de crédito.

Objetos del sistema de archivos integrado registrados por diario

Algunos tipos de objetos del sistema de archivos integrado pueden registrarse por diario utilizando el soporte del registro por diario de OS/400. Los tipos de objeto admitidos son los archivos continuos, los directorios y los enlaces simbólicos. Los sistemas de archivos "raíz" (/), QOpenSys y UDFS son los únicos que admiten el registro por diario de esos tipos de objeto. Los objetos del sistema de archivos integrado pueden registrarse por diario utilizando la interfaz del sistema tradicional (mandatos CI o API) o utilizando el Navigator de iSeries. Se puede Iniciar el registro por diario y Finalizar el registro por diario mediante el Navigator de iSeries, así como mostrar información de registro por diario.

Nota: No es posible registrar por diario los archivos continuos correlacionados en memoria y los archivos continuos que utiliza el Integrated xSeries Server for iSeries (IXS) para el espacio de almacenamiento en unidades virtuales.

La lista siguiente resume el soporte del registro por diario del sistema de archivos integrado:

- Para realizar operaciones de registro por diario en los tipos de objeto admitidos pueden utilizarse tanto los mandatos genéricos como las API. Estas interfaces generalmente aceptan la identificación del objeto mediante el nombre de vía de acceso, el ID de archivo o ambos.
- Algunos mandatos de operaciones de registro por diario, incluyendo Iniciar registro por diario, Finalizar registro por diario y Aplicar cambios registrados por diario, pueden ejecutarse en subárboles completos de objetos del sistema de archivos integrado. Opcionalmente pueden utilizarse listas de incorporación y exclusión que pueden utilizar comodines para los nombres de los objetos. Por ejemplo, se puede utilizar el mandato Iniciar registro por diario para especificar que se inicie para todos los objetos del árbol "/MiEmpresa" que coincidan con el patrón "*.data" pero excluyendo a todos los objetos que coincidan con los patrones "A*.data" y "B*.data".
- El soporte del registro por diario en directorios incluye operaciones de directorios como por ejemplo añadir enlaces, eliminar enlaces, crear objetos, renombrar objetos y trasladar objetos dentro del directorio.

Los directorios registrados por diario admiten un atributo que puede establecerse para que haga que los objetos nuevos del subárbol hereden el estado actual del registro por diario del directorio. Si este atributo se activa para un directorio registrado por diario, todos los archivos continuos, directorios y enlaces simbólicos creados o enlazados en el directorio (al añadir un enlace fijo o al renombrar o trasladar el objeto) harán que el sistema inicie automáticamente el registro por diario.

Nota: Si se finaliza el registro por diario para un objeto, y a continuación se renombra ese objeto en el mismo directorio en el que actualmente reside, no se inicia el registro por diario para el objeto, incluso si el directorio tiene activado el atributo de heredar estado de registro por diario actual.

- Los nombres de objeto y los de las vías de acceso completas se incluyen en varias entradas de diario de objetos del sistema de archivos integrado. Los nombres de objeto y los de las vías de acceso permiten el soporte de idioma nacional (NLS).
- Si el sistema finaliza de forma anómala, se proporciona una recuperación mediante la carga del programa inicial (IPL) del sistema para los objetos del sistema de archivos integrado registrados por diario.
- El límite máximo de grabación admitido por las API write() y writev() es de 2 gigabytes-1. El tamaño máximo de entrada de diario si se especifica RCVSIZOPT (*MAXOPT2) es de 4.000.000.000. De lo contrario, el tamaño máximo de entrada de diario es 15.761.440 bytes. Si registra por diario el archivo

continuo y tiene operaciones de grabación que sobrepasan el límite de los 15.761.440 bytes, le interesará utilizar el soporte de *MAXOPT2 para evitar que ocurran errores.

Para obtener más información sobre el registro por diario de objetos del sistema de archivos integrado, consulte el apartado de gestión por diario del iSeries Information Center.

Para obtener más información acerca del diseño de las diferentes entradas de diario, el archivo incluye de C, qp0ljrn1.h, que se encuentra en el miembro QSYSINC/H (QP0LJRN1), contiene detalles sobre el contenido de los datos de las entradas de diario del sistema de archivos integrado y de sus formatos.

Para obtener un listado completo de todas las entradas de diario depositadas para los objetos del sistema de archivos integrado, consulte el Buscador de código de diario del iSeries Information Center.

Operaciones registradas en diario

Las operaciones siguientes sólo se registran por diario cuando el tipo del objeto o enlace que está utilizando la operación es un tipo que también puede registrarse por diario.

- Crear un objeto.
- Añadir un enlace a un objeto existente.
- Desenlazar un enlace.
- Redenominar un enlace.
- Redenominar un identificador de archivos.
- Trasladar un enlace al directorio o fuera de él.

Las siguientes operaciones registradas por diario son específicas de un archivo continuo:

- Grabación de datos
- Truncar y ampliar un archivo
- Datos de archivo forzados
- Guardar con almacenamiento liberado

Las siguientes operaciones registradas por diario son aplicables a todos los tipos de objetos registrados por diario:

- Cambios de atributos (incluyendo cambios de seguridad como las autorizaciones y la propiedad)
- Abrir
- Cerrar
- Iniciar registro por diario
- Finalizar registro por diario
- Iniciar el mandato Aplicar cambios registrados por diario (APYJRNCHG)
- Finalizar el mandato Aplicar cambios registrados por diario (APYJRNCHG)
- Guardar
- Restaurar

Para obtener más información sobre el registro por diario de objetos del sistema de archivos integrado, consulte el tema gestión por diarios del iSeries Information Center. Para obtener un listado completo de todas las entradas de diario depositadas para los objetos del sistema de archivos integrado, consulte el tema buscador de códigos de diario de la **administración de sistemas**.

Consideraciones especiales sobre las entradas de diario

Muchas operaciones del sistema de archivos integrado registradas por diario utilizan internamente el control de compromiso para formar una sola transacción a partir de las distintas funciones realizadas durante las operaciones. No se puede considerar que estas operaciones registradas por diario estén completadas a menos que el ciclo del control de compromiso tenga una entrada de diario de compromiso (código de diario C, tipo CM). Las operaciones registradas por diario que contienen una entrada de diario de retroacción (código de diario C, tipo RB) en el ciclo del control de compromiso son operaciones que han finalizado de forma anómala y las entradas de diario de esas operaciones no deben reproducirse ni duplicarse.

Entre las entradas del sistema de archivos integrado registradas por diario (código de diario B) que utilizan el control de compromiso de esta forma, se incluyen:

- AA — Cambiar valor de auditoría
- B0 — Iniciar crear
- B1 — Crear resumen
- B2 — Añadir enlace
- B3 — Renombrar/trasladar
- B4 — Desenlazar (directorio padre)
- B5 — Desenlazar (enlace)
- FA — Cambiar atributo
- JT — Iniciar diario (sólo cuando el registro por diario se ha iniciado por una operación en un directorio con el valor Sí de heredar atributo de registro por diario)
- OA — Cambiar autorización
- OG — Cambiar grupo principal de objeto
- OO — Cambiar propietario de objeto

Varias entradas de diario del sistema de archivos integrado tienen un campo de datos específico que indica si la entrada es una entrada de resumen. Las operaciones que envían tipos de entradas resumen enviarán al diario dos tipos de entrada idénticos. La primera entrada contendrá un subconjunto de los datos específicos de la entrada. La segunda contendrá los datos específicos de la entrada completos e indicará que es una entrada resumen. Los programas que duplican el objeto y/o reproducen la operación, generalmente sólo se interesan por las entradas resumen.

Para una operación de crear de un directorio registrado por diario, la entrada de diario B1 (Crear resumen) tiene la consideración de entrada resumen.

Algunas operaciones registradas por diario necesitan enviar una entrada de diario que es la inversa de la operación. Por ejemplo, un ciclo de control de compromiso que contenga una entrada de diario B4 (Desenlazar) también puede contener una entrada de diario B2 (Añadir enlace). Este caso sólo se dará en operaciones que den como resultado una entrada de diario de retroacción (C — RB).

Este caso sólo puede darse por dos razones:

1. La operación estaba a punto de fallar y la entrada se necesitaba internamente para limpiar la vía de acceso del error.
2. La operación fue interrumpida por un corte eléctrico y, durante la IPL posterior, la recuperación que necesitaba enviar la entrada se ejecutó para retrotraer la operación interrumpida.

Apéndice A. Llamada de procedimiento remoto independiente del transporte

Desarrollada por Sun Microsystems, la Llamada de Procedimiento Remoto (RPC), separa y distribuye aplicaciones clientes desde un mecanismo servidor. Incluye un estándar para la representación de datos, llamado eXternal Data Representation, o XDR, que posibilita que más de un tipo de máquina tenga acceso a los datos transmitidos. La RPC independiente de transporte (TI-RPC) es la versión más reciente de RPC. Proporciona un método para separar el protocolo subyacente que se utiliza en la capa de red, proporcionando una transición más compacta de un protocolo a otro. Los únicos protocolos actualmente disponibles en el servidor iSeries son TCP y UDP.

El desarrollo de aplicaciones distribuidas a través de una red es una tarea más compacta cuando se utiliza RPC. Los destinos primarios son las aplicaciones que tienden a distribuir las interfaces de usuario y la recuperación de datos.

Selecciones de red

Las API siguientes proporcionan los medios para elegir el transporte en el que una aplicación debe ejecutarse.

Estas API requieren que el archivo *STMF /etc/netconfig exista en el sistema. Si el archivo netconfig no existiese en el directorio /etc, el usuario debe copiarlo del directorio /QIBM/ProdData/OS400/RPC. El archivo netconfig se encuentra siempre en el directorio QIBM/ProdData/OS400/RPC.

API	Descripción
endnetconfig()	Libera el puntero a los registros almacenados en el archivo netconfig
freenetconfig()	Libera la estructura netconfig devuelta a partir de la llamada a la función getnetconfig()
getnetconfig()	Devuelve el puntero al registro actual del archivo netconfig e incrementa su puntero hasta el registro siguiente
getnetconfig()	Devuelve el puntero a la estructura netconfig que corresponde al identificador de red de entrada
setnetconfig()	Inicializa el puntero del registro a la primera entrada del archivo netconfig. La función setnetconfig() debe utilizarse antes de que se utilice por primera vez la función getnetconfig(). La función setnetconfig() devuelve un handle exclusivo (un puntero a los registros almacenados en el archivo netconfig) para que la función getnetconfig() lo utilice.

Conversión de nombre a dirección

Las API siguientes posibilitan que una aplicación obtenga la dirección de un servicio o de un sistema principal especificado independientemente del transporte.

API	Descripción
netdir_free()	Libera las estructuras que las API de conversión de nombre a dirección han asignado
netdir_getbyaddr()	Correlaciona las direcciones con los nombres de sistema principal y los nombres de servicio
netdir_getbyname()	Correlaciona el nombre de sistema principal y el nombre de servicio especificados en el parámetro de servicio con un conjunto de direcciones coherentes con el transporte identificado en la estructura netconfig

API	Descripción
netdir_options()	Proporciona interfaces a las posibilidades específicas del transporte como, por ejemplo, la dirección de difusión y los recursos de puerto reservado de TCP y UDP
netdir_spperror()	Emite un mensaje informativo en el que se indica la causa del fallo de una API de conversión de nombre a dirección
taddr2uaddr()	Convierte una dirección (local) específica del transporte a una dirección (universal) independiente del transporte
uaddr2taddr()	Convierte una dirección (universal) independiente del transporte a una dirección (local) específica del transporte (estructura netbuf)

eXternal Data Representation (XDR)

Las API siguientes posibilitan que las aplicaciones con Llamada de Procedimiento Remoto (RPC) manejen estructuras de datos arbitrarias, independientemente de los órdenes de los bytes de los sistemas principales o de los convenios de diseño de las estructuras.

API	Descripción
xdr_array()	Una primitiva de tipo filtro que convierte entre matrices de longitud variable y sus representaciones externas correspondientes. Se llama a esta función para codificar o decodificar cada elemento de la matriz
xdr_bool()	Una primitiva de tipo filtro que convierte entre booleanos (que equivalen a enteros en C) y sus representaciones externas. Al codificar los datos, los valores que este filtro produce pueden ser 1 o 0.
xdr_bytes()	Una primitiva de tipo filtro que convierte entre matrices con un número definido de bytes y sus representaciones externas. Esta función se ocupa de un subconjunto de matrices genéricas en las que el tamaño de los elementos de la matriz se sabe que es 1 y la descripción externa de cada elemento está incorporada. La longitud de la secuencia de bytes se encuentra de forma explícita en un entero sin signo. La secuencia de bytes no termina con un carácter nulo. La representación externa de los bytes es idéntica a la representación interna.
xdr_char()	Una primitiva de tipo filtro que convierte entre los caracteres propios del lenguaje C y sus representaciones externas
xdr_double()	Una primitiva de tipo filtro que convierte entre los números con precisión doble propios del lenguaje C y su representación externa
xdr_double_char()	Una primitiva de tipo filtro que convierte entre los caracteres de dos bytes propios del lenguaje C y sus representaciones externas
xdr_enum()	Una primitiva de tipo filtro que convierte entre las enumeraciones (enum) propias del lenguaje C y sus representaciones externas
xdr_free()	Libera recursivamente el objeto al que apunta el puntero que se ha pasado
xdr_float()	Una primitiva de tipo filtro que convierte entre los números de coma flotante (números normalizados con una única coma flotante) propios del lenguaje C y sus representaciones externas.
xdr_int()	Una primitiva de tipo filtro que convierte entre enteros propios del lenguaje C y sus representaciones externas.
xdr_long()	Una primitiva de tipo filtro que convierte entre enteros de tipo long propios del lenguaje C y sus representaciones externas.
xdr_netobj()	Una primitiva de tipo filtro que convierte entre datos opacos de longitud variable y sus representaciones externas.

API	Descripción
xdr_opaque()	Una primitiva de tipo filtro que convierte entre datos opacos de tamaño fijo y sus representaciones externas.
xdr_pointer()	Proporciona seguimiento de punteros dentro de estructuras y serializa punteros de tipo nulo. Puede representar estructuras de datos recursivas como, por ejemplo, árboles binarios o listas enlazadas.
xdr_reference()	Una primitiva de tipo filtro que proporciona seguimiento a punteros dentro de estructuras. Esta primitiva posibilita serializar, deserializar y liberar cualquier puntero dentro de una estructura que esté referenciado por otra estructura. La función xdr_reference() no concede un significado especial a un puntero nulo durante el proceso de serialización, por lo que el hecho de pasar la dirección de un puntero nulo puede provocar un error de memoria. En consecuencia, el programador debe describir los datos con una unión discriminada de dos lados. Un lado se utiliza cuando el puntero es válido, mientras que el otro se utiliza cuando el puntero es nulo.
xdr_short()	Una primitiva de tipo filtro que convierte entre enteros de tipo short propios del lenguaje C y sus representación externa
xdr_string()	Una primitiva de tipo filtro que convierte entre cadenas de caracteres propias del lenguaje C y sus representaciones externas correspondientes
xdr_u_char()	Una primitiva de tipo filtro que convierte entre los caracteres sin signo propios del lenguaje C y sus representaciones externas
xdr_u_int()	Una primitiva de tipo filtro que convierte entre los enteros sin signo propios del lenguaje C y sus representaciones externas
xdr_u_long()	Una primitiva de tipo filtro que convierte entre los enteros sin signo de tipo long propios del lenguaje C y sus representaciones externas
xdr_u_short()	Una primitiva de tipo filtro que convierte entre los enteros sin signo de tipo short propios del lenguaje C y sus representaciones externas
xdr_union()	Una primitiva de tipo filtro que convierte entre uniones C discriminadas y sus representaciones externas correspondientes
xdr_vector()	Una primitiva de tipo filtro que convierte entre matrices de longitud fija y sus representaciones externas correspondientes
xdr_void()	No tiene parámetros. Se pasa a otras funciones RPC que necesitan un parámetro, pero no transmite datos
xdr_wrapstring()	Una primitiva que llama a la API xdr_string(xdr, sp, maxuint), donde maxuint es el valor máximo de un entero sin signo. xdr_wrapstring() es útil porque el paquete RPC pasa un máximo de dos funciones XDR como parámetros y la función xdr_string() necesita tres.

Autenticación

Las API siguientes proporcionan autenticación a las aplicaciones con Llamada de procedimiento remoto independiente del transporte (TI-RPC).

API	Descripción
auth_destroy()	Destruye la estructura de la información de autenticación a la que el parámetro auth apunta
authnone_create()	Crea y devuelve un handle de autenticación RPC por omisión que pasa información de autenticación nula con cada llamada de procedimiento remoto.
authsys_create()	Crea y devuelve un handle de autenticación RPC que contiene información de autenticación

RPC independiente del transporte (TI-RPC)

Las API siguientes proporcionan un entorno para el desarrollo de aplicaciones distribuidas al aislar la aplicación de cualquier dispositivo de transporte específico. De esta forma se incrementa la facilidad de uso de los transportes.

Interfaces API TI-RPC simplificadas

Las API simplificadas siguientes especifican el tipo de transporte que ha de utilizarse. Las aplicaciones que utilicen este nivel no tienen que crear handles explícitamente.

API	Descripción
rpc_call()	Llama a un procedimiento remoto que se encuentra en el sistema especificado.
rpc_reg()	Registra un procedimiento con paquete de servicio RPC

Interfaces API TI-RPC de nivel superior

Las API siguientes posibilitan que la aplicación especifique el tipo de transporte.

API	Descripción
clnt_call()	Llama a un procedimiento remoto asociado con el cliente
clnt_control()	Cambia la información relativa a un objeto de cliente
clnt_create()	Crea un handle genérico de cliente
clnt_destroy()	Destruye el handle de RPC de cliente
svc_create()	Crea un handle de servidor
svc_destroy()	Destruye un handle de transporte del servicio RPC

Interfaces API TI-RPC de nivel intermedio

Las API siguientes son parecidas a las API de nivel superior, pero las aplicaciones de usuario seleccionan la información específica de transporte mediante las API de selección de red:

API	Descripción
clnt_tp_create()	Crea un handle de cliente
svc_tp_create()	Crea un handle de servidor

Interfaces API TI-RPC de nivel experto

Las API siguientes posibilitan que la aplicación seleccione el transporte que se desea utilizar. También ofrecen un nivel mayor de control sobre los detalles de los handles CLIENT y SVCXPRT. Estas API son parecidas a las API de nivel intermedio con un control adicional que se proporciona al utilizar las API de conversión de nombre a dirección.

Control adicional que se obtiene al utilizar las API de conversión de nombre a dirección.

API	Descripción
clnt_tli_create()	Crea un handle de cliente
rpcb_getaddr()	Busca la dirección universal de un servicio
rpcb_set()	Registra la dirección del servidor con el RPCbind
rpcb_unset()	Los servidores la utilizan para eliminar el registro de sus direcciones

API	Descripción
svc_reg()	Asocia un programa y la versión con despachar
svc_tli_create()	Crea un handle de servidor
svc_unreg()	Elimina una asociación definida mediante svc_reg()

Otras interfaces API TI-RPC

Estas API posibilitan que diversas aplicaciones funcionen de forma coordinada con las API simplificadas, de nivel superior, de nivel intermedio y de nivel experto.

API	Descripción
clnt_freeres()	Libera los datos asignados mediante el sistema RPC o el XDR
clnt_geterr()	Obtiene la estructura de error a partir del handle de cliente
svc_freeargs()	Libera los datos asignados mediante el sistema RPC o el XDR
svc_getargs()	Decodifica los argumentos de una petición RPC
svc_getrpccaller()	Obtiene la dirección de red del llamador
svc_run()	Espera a que lleguen las peticiones RPC
svc_sendreply()	Envía los resultados de una llamada de procedimiento a un cliente remoto.
svcerr_decode()	Envía información al cliente relativa a un error en la decodificación
svcerr_noproc()	Envía información al cliente relativa a un error en el número del procedimiento
svcerr_systemerr()	Envía información al cliente relativa a un error en el sistema

Apéndice B. Programa de ejemplo con funciones C del sistema de archivos integrado

Este sencillo programa de lenguaje C ilustra la utilización de varias funciones del sistema de archivos integrado. El programa ejecuta las operaciones siguientes:

- 1** Utiliza la función **getuid()** para determinar el ID de usuario (uid) real.
- 2** Utiliza la función **getcwd()** para determinar el directorio actual.
- 3** Utiliza la función **open()** para crear un archivo. Define que el propietario (la persona que ha creado el archivo) tenga autorización de lectura, grabación y ejecución sobre el archivo.
- 4** Utiliza la función **write()** para grabar una serie de bytes en el archivo. El descriptor de archivo proporcionado en la operación de apertura (**3**) identifica al archivo.
- 5** Utiliza la función **close()** para cerrar el archivo.
- 6** Utiliza la función **mkdir()** para crear un nuevo subdirectorio en el directorio actual. El propietario obtiene acceso de lectura, grabación y ejecución al subdirectorio.
- 7** Utiliza la función **chdir()** para pasar del subdirectorio nuevo al directorio actual.
- 8** Utiliza la función **link()** para crear un enlace con el archivo creado anteriormente (**3**).
- 9** Utiliza la función **open()** para abrir el archivo sólo para lectura. El enlace creado en (**8**) posibilita el acceso al archivo.
- 10** Utiliza la función **read()** para leer una serie de bytes del archivo. El descriptor de archivo proporcionado en la operación de apertura (**9**) identifica al archivo.
- 11** Utiliza la función **close()** para cerrar el archivo.
- 12** Utiliza la función **unlink()** para eliminar el enlace con el archivo.
- 13** Utiliza la función **chdir()** para pasar del directorio actual al directorio padre en el que se ha creado el nuevo subdirectorio.
- 14** Utiliza la función **rmdir()** para eliminar el subdirectorio creado anteriormente (**6**).
- 15** Utiliza la función **unlink()** para eliminar el archivo creado anteriormente (**3**).

Nota: este programa de ejemplo funcionará correctamente en aquellos sistemas en los que el CCSID del trabajo en el que se ejecuta sea 37. Los nombres de vía de acceso y de objeto de las API del sistema de archivos integrado deben codificarse en el CCSID del trabajo; sin embargo, el compilador C almacena las constantes de tipo carácter en el CCSID 37. Para que la compatibilidad sea total, convierta las constantes de tipo carácter, como, por ejemplo, los nombres de vía de acceso y de objeto, antes de pasar las API al CCSID del trabajo.

Esta información de exención de responsabilidad corresponde a ejemplos de código.

```
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/types.h>
```

```
#define BUFFER_SIZE      2048
#define NEW_DIRECTORY    "dirtest"
#define TEST_FILE        "test.file"
#define TEST_DATA        "¡Hola, mundo!"
#define USER_ID         "user_id_"
#define PARENT_DIRECTORY ".."
```

```

char InitialFile[BUFFER_SIZE];
char LinkName[BUFFER_SIZE];
char InitialDirectory[BUFFER_SIZE] = ".";
char Buffer[32];
int FilDes = -1;
int BytesRead;
int BytesWritten;
uid_t UserID;

void CleanUpOnError(int level){
    printf("Encontrado error, borrando.\n");
    switch ( level )      {
        case 1:
            printf("No se pudo obtener directorio de trabajo actual.\n");
            break;
        case 2:
            printf("No se pudo crear el archivo %s.\n",TEST_FILE);
            break;
        case 3:
            printf("No se pudo grabar en archivo %s.\n",TEST_FILE);
            close(FilDes);
            unlink(TEST_FILE);
            break;
        case 4:
            printf("No se pudo cerrar el archivo %s.\n",TEST_FILE);
            close(FilDes);
            unlink(TEST_FILE);
            break;
        case 5:
            printf("No se pudo crear el directorio %s.\n",NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 6:
            printf("No se pudo ir al directorio %s.\n",NEW_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 7:
            printf("No se pudo crear enlace %s con %s.\n",LinkName,InitialFile);
            chdir(PARENT_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 8:
            printf("No se pudo abrir el enlace %s.\n",LinkName);
            unlink(LinkName);
            chdir(PARENT_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 9:
            printf("No se pudo leer el enlace %s.\n",LinkName);
            close(FilDes);
            unlink(LinkName);
            chdir(PARENT_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 10:
            printf("No se pudo cerrar el enlace %s.\n",LinkName);
            close(FilDes);
            unlink(LinkName);
            chdir(PARENT_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
    }
}

```

```

        case 11:
            printf("No se pudo deshacer el enlace %s.\n",LinkName);
            unlink(LinkName);
            chdir(PARENT_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 12:
            printf("No se pudo ir al directorio %s.\n",PARENT_DIRECTORY);
            chdir(PARENT_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 13:
            printf("No se pudo eliminar el directorio %s.\n",NEW_DIRECTORY);
            rmdir(NEW_DIRECTORY);
            unlink(TEST_FILE);
            break;
        case 14:
            printf("No se pudo desenlazar el archivo %s.\n",TEST_FILE);
            unlink(TEST_FILE);
            break;
        default:
            break;
    }
    printf("El programa finalizó con Error.\n"
        "Pueden no estar eliminados todos los archivos y directorios de prueba.\n");}

int main (){
    1 /* Obtener e imprimir el ID de usuario real con la función getuid(). */
    UserID = getuid();
    printf("El ID de usuario real es %u. \n",UserID);
    2 /* Obtener el directorio de trabajo actual y almacenarlo en InitialDirectory. */
    if ( NULL == getcwd(InitialDirectory,BUFFER_SIZE) ) {
        perror("Error getcwd");
        CleanupOnError(1);    return 0;
    }
    printf("El directorio de trabajo actual es %s. \n",InitialDirectory);
    3 /* Crear el archivo TEST_FILE para grabar, si no existe.
    Otorgar al propietario autorización de lectura, grabación y ejecución. */
    FilDes = open(TEST_FILE, O_WRONLY | O_CREAT | O_EXCL, S_IRWXU);
    if ( -1 == FilDes ) {
        perror("Error open");
        CleanupOnError(2);    return 0;
    }
    printf("Creado %s en directorio %s.\n",TEST_FILE,InitialDirectory);
    4 /* Grabar TEST_DATA en TEST_FILE vía FilDes */
    BytesWritten = write(FilDes,TEST_DATA,strlen(TEST_DATA));
    if ( -1 == BytesWritten ) {
        perror("Error write");
        CleanupOnError(3);    return 0;
    }
    printf("Grabado %s en archivo %s.\n",TEST_DATA,TEST_FILE);
    5 /* Cerrar TEST_FILE vía FilDes */
    if ( -1 == close(FilDes) ) {
        perror("Error close");
        CleanupOnError(4);    return 0;
    }
    FilDes = -1;
    printf("Archivo %s cerrado.\n",TEST_FILE);
    6 /* Crear un nuevo directorio en el directorio de trabajo actual y
    otorgar al propietario autorización de lectura, grabación y ejecución */

```

```

    if ( -1 == mkdir(NEW_DIRECTORY, S_IRWXU) )    {
        perror("Error mkdir");
        CleanupOnError(5);    return 0;
    }
    printf("Directorio %s creado en directorio
%s.\n",NEW_DIRECTORY,InitialDirectory);
7
/* Cambiar el directorio de trabajo actual por el
directorío NEW_DIRECTORY recién creado. */
    if ( -1 == chdir(NEW_DIRECTORY) )    {
        perror("Error chdir");
        CleanupOnError(6);    return 0;
    }
    printf("Se ha pasado al directorio
%s/%s.\n",InitialDirectory,NEW_DIRECTORY);
/* Copiar PARENT_DIRECTORY en InitialFile y
añadir "/" y TEST_FILE a InitialFile. */
    strcpy(InitialFile,PARENT_DIRECTORY);
    strcat(InitialFile,"/");
    strcat(InitialFile,TEST_FILE);
/* Copiar USER_ID en LinkName después añadir el
ID usuario como una serie a LinkName. */
    strcpy(LinkName, USER_ID);
    sprintf(Buffer, "%d\0", (int)UserID);
    strcat(LinkName, Buffer);
8
/* Crear un enlace con el nombre de InitialFile con LinkName. */
    if ( -1 == link(InitialFile,LinkName) )    {
        perror("Error link");
        CleanupOnError(7);    return 0;
    }
    printf("Creado un enlace %s con %s.\n",LinkName,InitialFile);
9
/* Abrir el archivo LinkName sólo para lectura. */
    if ( -1 == (FilDes = open(LinkName,O_RDONLY)) )    {
        perror("Error open");
        CleanupOnError(8);    return 0;
    }
    printf("Abierto %s para lectura.\n",LinkName);
10
/* Leer en archivo LinkName, vía FilDes, en Buffer. */
    BytesRead = read(FilDes,Buffer,sizeof(Buffer));
    if ( -1 == BytesRead )    {
        perror("Error read");
        CleanupOnError(9);    return 0;
    }
    printf("%s leído en %s.\n",Buffer,LinkName);
    if ( BytesRead != BytesWritten )    {
        printf("AVISO: el número de bytes leídos no "\
"es igual al número de bytes grabados.\n");    }
11
/* Cerrar el archivo LinkName vía FilDes. */
    if ( -1 == close(FilDes) )    {
        perror("Error close");
        CleanupOnError(10);    return 0;
    }
    FilDes = -1;
    printf("Cerrado %s.\n",LinkName);
12
/* Desenlazar el enlace LinkName con InitialFile. */
    if ( -1 == unlink(LinkName) )    {
        perror("Error unlink");
        CleanupOnError(11);    return 0;
    }
    printf("%s está desenlazado.\n",LinkName);
13

```

```

/* Volver a cambiar el directorio de trabajo actual
por el directorio inicial. */
if ( -1 == chdir(PARENT_DIRECTORY) )    {
    perror("Error chdir");
    CleanupOnError(12);    return 0;
}
printf("pasando del directorio al %s.\n",InitialDirectory);
14
/* Eliminar el directorio NEW_DIRECTORY */
if ( -1 == rmdir(NEW_DIRECTORY) )    {
    perror("Error rmdir");
    CleanupOnError(13);    return 0;
}
printf("Eliminando directorio %s.\n",NEW_DIRECTORY);
15
/* Desenlazar el archivo TEST_FILE */
if ( -1 == unlink(TEST_FILE) )    {
    perror("Error unlink");
    CleanupOnError(14);    return 0;
}
printf("Desenlazando archivo %s.\n",TEST_FILE);

printf("Programa terminado satisfactoriamente.\n");    return 0;
}

```

Apéndice C. Ejemplo de código RPG para un sistema de archivos integrado

Fragmentos de código  contiene un ejemplo escrito en RPG de un sistema de archivos integrado. Para ver este ejemplo, siga estos pasos:

1. Seleccione **ILE RPG Source** en la lista desplegable de la categoría Search.
2. Pulse en **Search**.
3. Desplace la lista hasta que vea la opción **Using IFS from RPG**.
4. Pulse **Code for using IFS from RPG**.

Esta información de exención de responsabilidad corresponde a ejemplos de código.

Bibliografía

En esta bibliografía se lista la información del servidor iSeries que contiene información previa o más detallada de la que se expone en este manual.

- La información acerca del lenguaje de control en la categoría **Programación** del iSeries Information Center proporciona una descripción del lenguaje de control (CL) del servidor iSeries y de sus mandatos. Cada descripción de mandato incluye un diagrama de sintaxis, parámetros, valores por omisión, palabras clave y un ejemplo.
- El tema Globalización de iSeries Information Center explica los conceptos del soporte de idioma nacional (NLS), como por ejemplo, el juego de caracteres y la página de códigos, y ofrece información necesaria para evaluar, planificar y utilizar el NLS y las posibilidades plurilingües del servidor iSeries.
- La información de las APIs de la categoría **Programación** del iSeries Information Center proporciona una descripción de cada OS/400 API, inclusive las API sistema de archivos integrado.
- La información sobre la Administración de diarios de la categoría **Administración de sistemas** del iSeries Information Center proporciona información sobre cómo configurar, gestionar y solucionar problemas relacionados con la protección de rutas de acceso gestionada por el sistema (SMAPP), diarios locales y diarios remotos en un servidor iSeries.
- La información sobre el control de compromiso de la categoría **Base de datos** del iSeries Information Center explica cómo definir y procesar un grupo de cambios para recursos, por ejemplo archivos de base de datos o archivos sistema de archivos integrado, como una unidad de trabajo lógica.
- Soporte del sistema de archivos de red de

OS/400  En esta publicación está descrito el sistema de archivos de red por medio de una serie de aplicaciones sacadas de la vida real. Se incluye información sobre exportación, montaje y bloqueo de archivos, así como conceptos de seguridad. Con esta publicación aprenderá a utilizar NFS para crear y desarrollar un espacio de nombres de red protegido.

- Optical Support  Este manual sirve de guía del usuario y manual de consulta para IBM Optical Support en OS/400. La información de esta publicación puede servir de ayuda al usuario para comprender los conceptos de servidor de datos de biblioteca óptica, planificar una biblioteca óptica, administrar y manejar un servidor de datos de biblioteca óptica y solucionar problemas relativos al servidor de datos de biblioteca óptica.
- WebSphere Development Studio: Guía para programadores ILE C/C++  Esta publicación proporciona la información necesaria para diseñar, editar, compilar, ejecutar y depurar programas ILE C/400 en el servidor iSeries.
- WebSphere Development Studio: Referencia de lenguaje C/C++  Esta publicación proporciona información acerca de la estructura de los programas ILE C/400 y contiene detalles sobre las funciones de biblioteca y los archivos de incluir (cabecera).
- Security — Reference  Esta publicación proporciona información técnica detallada acerca de la seguridad de OS/400.
- APPC Programming  Este manual describe el soporte APPC (comunicaciones avanzadas entre programas) para el servidor iSeries. Constituye una guía para realizar el desarrollo de programas de aplicación que utilizan APPC y para la definición del entorno de comunicaciones para APPC.
- Copia de seguridad y recuperación  Este libro proporciona información general sobre las opciones de recuperación y disponibilidad para el servidor iSeries de IBM.

Bibliografía

Índice

A

- Acceso de cliente 33
- API
 - ILE C/400 55
 - programa de ejemplo 105
 - reglas de nombres de vías de acceso 56
 - utilización en programas C 45, 49
- archivo 105
 - menús y pantallas 26
 - modalidades de abrir 59
 - transferir 32
- archivo continuo
 - ¿qué es? 3
 - comparación con archivo orientado a registro 3
 - copiar en/de archivo de base de datos 46
 - en programa de ejemplo 105
 - indicando utilización en ILE C/400 55
 - por qué utilizarlo 3
 - utilización en programas 45
 - ventajas 1
- archivo de base de datos
 - comparación con archivo continuo 3
 - copiar en/de archivo continuo 46
 - crear a partir de archivo continuo 46
- archivos de salvar
 - utilización en el sistema de archivos QSYS.LIB 74
 - utilización en el sistema de archivos QSYS.LIB de ASP independiente 76
- atributos ampliados
 - ¿qué son? 22
 - continuidad a través de idiomas nacionales 23, 59
 - directrices de denominación 23
- autorización
 - en programa de ejemplo 105
 - limitaciones del sistema de archivos QNTC 86
 - limitaciones en el sistema de archivos
 - QFileSvr.400 90
 - mandatos 26
 - manejo en programas 58

B

- bibliografía 113

C

- carpetas
 - sistema de archivos QDLS 5, 78
- cliente PC
 - trabajar con el sistema de archivos integrado 31
- clientes PC
 - cómo están representados los sistemas de archivos 34
- conversión
 - datos 59
 - nombres de objeto 23, 59
- conversión de caracteres 2, 23, 59

- conversión de datos 59

D

- definido por el usuario, sistema de archivos 4
 - descripción 4
- descripción de archivo abierto 57
- descriptor de archivo 57
- directorio
 - ¿qué es? 6
 - actual 8
 - en programa de ejemplo 105
 - inicial (home) 8
 - mandatos 26
 - menús y pantallas 26
 - sistema de archivos integrado 35
 - ventajas 1
- directorio actual 8
- directorio de trabajo 8
- directorio inicial (home) 8

E

- ejemplo
 - nombres de vías de acceso 18, 29, 56
 - programa utilizando las API del sistema de archivos integrado 105
 - utilización de enlace simbólico 20
- enlace
 - ¿qué es? 18
 - comparación 21
 - en programa de ejemplo 105
 - fijo 19
 - mandatos 26
 - menús y pantallas 26
 - por qué utilizarlo 18
 - simbólico 20
 - utilización en el sistema de archivos / (raíz) 66
 - utilización en el sistema de archivos QDLS 79
 - utilización en el sistema de archivos
 - QFileSvr.400 90
 - utilización en el sistema de archivos QNTC 86
 - utilización en el sistema de archivos QOpenSys 68
 - utilización en el sistema de archivos QOPT 81
 - utilización en el sistema de archivos QSYS.LIB 74
 - utilización en el sistema de archivos QSYS.LIB de ASP independiente 77
- enlace fijo
 - ¿qué es? 19
 - comparación con enlace simbólico 21
- enlace simbólico
 - ¿qué es? 20
 - comparación con enlace fijo 21
 - ejemplo de utilización 20
- espacio de usuario
 - utilización en el sistema de archivos QSYS.LIB 73

espacio de usuario (*continuación*)
 utilización en el sistema de archivos QSYS.LIB de
 ASP independiente 76
esquemas de codificación 23, 59

F

FTP 32
funciones
 en programa de ejemplo 105
 ILE C/400 55
 reglas de nombres de vías de acceso 56
 utilización en programas C 45, 49

I

ILE C/400
 alternativas API 49
 funciones ANSI 55
información impresa relacionada 113
interfaz de sistema de archivos integrado 1, 2, 5
interfaz de usuario
 mandatos 26
 menús y pantallas 25
 vista desde PC 34

L

LU 6.2 en el sistema de archivos QFileSvr.400 88

M

mandatos
 lista 26
 reglas de nombres de vías de acceso 29
 utilización 26
menús
 reglas de nombres de vías de acceso 29
 utilización 25
migración a través de sistemas de archivos 34
modalidad de acceso 58
modalidad de archivo abierto binario 59
modalidad de archivo abierto de texto 59
modalidades de abrir archivo 59

N

NetServer 33
NetWare (QNetWare), sistema de archivos
 características y limitaciones 82
nombre de vía de acceso
 ¿qué es? 17
 nombre de vía de acceso absoluto 18
 nombre de vía de acceso relativo 18
 reglas para las API 56
 reglas para mandatos y pantallas 29
 utilización en el sistema de archivos / (raíz) 66
 utilización en el sistema de archivos QDLS 79
 utilización en el sistema de archivos
 QFileSvr.400 88

nombre de vía de acceso (*continuación*)
 utilización en el sistema de archivos QNTC 86
 utilización en el sistema de archivos QOpenSys 67
 utilización en el sistema de archivos QOPT 81
 utilización en el sistema de archivos QSYS.LIB 74
 utilización en el sistema de archivos QSYS.LIB de
 ASP independiente 77
nombre de vía de acceso absoluto 18
nombre de vía de acceso relativo 18
nombres
 continuidad a través de esquemas de
 codificación 23
 continuidad a través de idiomas nacionales 59
 utilización en el sistema de archivos / (raíz) 66
 utilización en el sistema de archivos QDLS 79
 utilización en el sistema de archivos
 QFileSvr.400 88
 utilización en el sistema de archivos QNTC 86
 utilización en el sistema de archivos QOpenSys 67
 utilización en el sistema de archivos QOPT 81
 utilización en el sistema de archivos QSYS.LIB 74
 utilización en el sistema de archivos QSYS.LIB de
 ASP independiente 76

O

objeto
 migración entre sistemas de archivos 34
operaciones (programa de ejemplo) 105
óptico, sistema de archivos (QOPT)
 descripción 5

P

página de códigos 2, 23, 59
pantallas
 reglas de nombres de vías de acceso 29
 utilización 25
permiso 58
procesador de E/S servidor de archivos 6
programa de lenguaje C
 ejemplo 105
 funciones ILE C/400 55
protocolo de transferencia de archivos 32
punteros 57

Q

QFileSvr.400 5
QFileSvr.400 (QFileSvr.400), sistema de archivos
 descripción 5
QFileSvr.400, sistema de archivos
 descripción 5
QNetWare, sistema de archivos 5
 características y limitaciones 82
 descripción 5
QOPT 5
QSYS.LIBde ASP independiente
 características y limitaciones 75

R

- raíz (/), sistema de archivos
 - características y limitaciones 65
 - descripción 4
- red, sistema de archivos 5
 - descripción 5
- Registrar por diario
 - Finalizar 42
 - Inicio 42
- Registrar por diario los objetos del sistema de archivos integrado 95

S

- seguridad
 - limitaciones del sistema de archivos QNTC 86
 - limitaciones en el sistema de archivos QFileSvr.400 90
 - mandatos 26
 - manejo en programas 58
- servidor de archivos 6
- servidor de archivos de PC 6
- servidor de archivos OS/400 6
- servidor de archivos OS/400 (QFileSvr.400), sistema de archivos
 - características y limitaciones 88
- sistema de archivos
 - ¿qué es? 4
 - ASP independiente QSYS.LIB
 - descripción 5
 - biblioteca (QSYS.LIB)
 - características y limitaciones 72
 - descripción 4
 - comparación 61
 - de servidor de archivos OS/400 (QFileSvr.400)
 - características y limitaciones 88
 - definido por el usuario, sistema de archivos
 - descripción 4
 - interfaz 5
 - migrar objetos 34
 - NetWare (QNetWare), sistema de archivos
 - características y limitaciones 82
 - óptico (QOPT)
 - características y limitaciones 80
 - QNetWare, sistema de archivos
 - descripción 5
 - QSYS.LIB de ASP independiente
 - características y limitaciones 75
 - raíz (/)
 - características y limitaciones 65
 - descripción 4
 - servicios de biblioteca de documentos (QDLS)
 - características y limitaciones 78
 - descripción 5
 - servidor QNTC
 - características y limitaciones 85
 - sistema de archivos de red
 - características y limitaciones 91
 - descripción 5

- sistema de archivos (*continuación*)
 - sistema de archivos definido por el usuario
 - características y limitaciones 68
 - sistema de archivos óptico (QOPT)
 - descripción 5
 - sistema de archivos QFileSvr.400 (QFileSvr.400)
 - descripción 5
 - sistema de archivos QNTC
 - descripción 5
 - sistemas abiertos (QOpenSys)
 - características y limitaciones 67
 - descripción 4
 - transferir archivos 32
 - ventajas 2
- sistema de archivos de biblioteca (QSYS.LIB)
 - características y limitaciones 72
 - descripción 4
- sistema de archivos de red
 - características y limitaciones 91
- sistema de archivos de servicios de biblioteca de documentos (QDLS)
 - características y limitaciones 78
 - descripción 5
- sistema de archivos de sistemas abiertos (QOpenSys)
 - características y limitaciones 67
 - descripción 4
- sistema de archivos de Windows NT Server (QNTC)
 - características y limitaciones 85
- sistema de archivos definido por el usuario
 - características y limitaciones 68
- sistema de archivos integrado
 - ¿por qué utilizarlo? 1
 - ¿qué es? 1
 - interfaces de programación
 - programa de ejemplo 105
 - punteros y descriptores de archivo 57
 - reglas de nombres de vías de acceso 56
 - seguridad 58
 - soporte de idioma nacional 59
 - utilización en programas C 49
 - mandatos
 - lista 26
 - reglas de nombres de vías de acceso 29
 - utilización 26
 - menús y pantallas
 - reglas de nombres de vías de acceso 29
 - utilización 25
 - trabajar desde PC
 - cómo están representados los sistemas de archivos 34
 - trabajar desde un PC
 - interacción con directorios y objetos 31
- sistema de archivos jerarquizado (HFS)
 - utilización de API para el sistema de archivos QDLS 78
 - utilización de las API para el sistema de archivos QOPT 81
- sistema de archivos ópticos (QOPT)
 - características y limitaciones 80
- sistema de archivos QDLS
 - características y limitaciones 78

- sistema de archivos QDLS (*continuación*)
 - descripción 5
- sistema de archivos QFileSvr.400
 - características y limitaciones 88
- sistema de archivos QNTC 5
 - características y limitaciones 85
 - descripción 5
- sistema de archivos QOpenSys
 - características y limitaciones 67
 - descripción 4
- sistema de archivos QOPT
 - características y limitaciones 80
 - descripción 5
- sistema de archivos QSYS.LIB
 - características y limitaciones 72
 - descripción 4
- sistema de archivos QSYS.LIB de ASP independiente
 - descripción 5
- soporte de idioma nacional 2, 23, 59

T

TCP/IP en el sistema de archivos QFileSvr.400 88

U

unicode 23

Z

zócalo 59



Impreso en España